

基础

Bounds

```
new Bounds (southWest,northEast)
```

描述:

-
-

Bounds 范围框

-
-

示例

```
let p1 = new KMap.Point(112.455121,22.55764);
let p2 = new KMap.Point(113.98416401,22.67109872);
let bounds = new KMap.Bounds(p1,p2);
```

参数:

名称	类型	描述
<code>southWest</code>	<code>Point</code>	西南方点
<code>northEast</code>	<code>Point</code>	东北方点

函数

```
getSouthWest()
```

描述:

-
-

获取西南方点坐标(左下角)

-
-

示例

```
let southWestPoint = bounds.getSouthWest();
```

返回:

点实例

类型:

Point

getNorthEast()

描述:

-
-

获取东北方点坐标(右上角)

-
-

示例

```
let northEastPoint = bounds.getNorthEast();
```

返回:

点实例

类型:

Point

getNorthWest()

描述:

-

-

获取西北方点坐标(左上角)

-

-

示例

```
let northWestPoint = bounds.getNorthWest();
```

返回:

点实例

类型:

Point

getSouthEast()

描述:

-

-

获取东南方点坐标(右下角)

-

-

示例

```
let southEastPoint = bounds.getSouthEast();
```

返回:

点实例

类型:

Point

getCenter()

描述:

-
-

获取中心点坐标

-
-

示例

```
let center = bounds.getCenter();
```

返回:

中心点

类型:

Point

getPoints()

描述:

-
-

获取四个点数组[左上角, 右上角, 右下角, 左下角]

-
-

示例

```
let points = bounds.getPoints();
```

返回:

点数组

类型:

Array.<Point>

toArray()

描述:

-
-

获取 Box

-
-

示例

```
let points = bounds.toArray();
```

返回:

点数组

类型:

`Array.<Point>`

toJSON()

描述:

-
-

获取对象 {left: *, bottom: *, right: *, top: *}

-
-

示例

```
let json = bounds.toJSON();
```

返回:

对象

类型:

`Object`

toString()

描述:

-
-

获取字符串 lon, lat;lon, lat

-
-

示例

```
let pointStr = bounds.toString();
```

返回:

点坐标字符串

类型:
string

byPoints(points)

描述:

-
-

通过很多的点获取矩形框

-
-

示例

```
let newBounds = bounds.byPoints([p1,p2,p3,p4]);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
points	Array.<Point>	点数组

contains(*point*)

描述:

-
-

点是否在矩形框中

-
-

示例

```
let p = new KMap.Point(112.455121,22.55764);  
let isContains = bounds.contains(p);
```

参数:

名称	类型	描述
point	Point	点实例

返回:

是/否

类型:
boolean

clone()

描述:

-
-

克隆对象

-
-

示例

```
let newBounds = bounds.clone();
```

返回:

Bounds 对象

类型:
 Bounds

Pixel

```
new Pixel (x,y)
```

描述:

-
-

Pixel 像素点

-
-

示例

```
let pixel = new KMap.Pixel(100,50);
```

参数:

名称	类型	描述
<code>x</code>	<i>number</i>	x 坐标像素值
<code>y</code>	<i>number</i>	y 坐标像素值

函数

toArray()

描述:

-
-

获取数组格式像素[x, y]

-
-

示例

```
let arr = pixel.toArray();
```

返回:

[x,y]坐标像素值

类型:

`Array.<number>`

toString()

描述:

-
-

获取字符串格式像素"x, y"

-
-

示例

```
let str = pixel.toString();
```

返回:

"x,y"坐标像素值

类型:
string

distance(p)

描述:

-
-

计算坐标距离

-
-

示例

```
let len = pixel.distance(new KMap.Pixel(400,800));
```

参数:

名称	类型	描述
p	Pixel	像素点

返回:

像素距离

类型:
number

subtract(p)

描述:

-
-

减去该点的 x 和 y 坐标, 返回一个新坐标

-

-

示例

```
let p = pixel.subtract(new KMap.Pixel(400,800));
```

参数:

名称	类型	描述
p	Pixel	像素点

返回:

新像素点

类型:
Pixel

```
addition(p)
```

描述:

-
-

加上该点的 x 和 y 坐标, 返回一个新坐标

-
-

示例

```
let p = pixel.addition(new KMap.Pixel(400,800));
```

参数:

名称	类型	描述
p	Pixel	像素点

返回:

新像素点

类型:

Pixel

mag()

描述:

-
-

从 0, 0 坐标到该点的 x 和 y 的距离

-
-

示例

```
let len = pixel.mag();
```

返回:

像素距离

类型:

number

div(k)

描述:

-
-

像素缩小倍数

-
-

示例

```
let p = pixel.div(2);
```

参数:

名称	类型	描述
k	number	倍数

返回:

新像素点

类型:

Pixel

```
mult(k)
```

描述:

-
-

像素放大倍数

-
-

示例

```
let p = pixel.mult(2);
```

参数:

名称	类型	描述
k	number	倍数

返回:

新像素点

类型:

Pixel

angleWith(*p*)

描述:

-
-

计算和另一个像素点的角度

-
-

示例

```
let p = pixel.angleWith(new KMap.Pixel(100,100));
```

参数:

名称	类型	描述
p	Pixel	像素点

返回:

角度

类型:
Number

clone()

描述:

-
-

克隆对象

-
-

示例

```
let p = pixel.clone();
```

返回:

新像素点

类型:

Pixel

属性

```
x
```

描述:

-
-

x 属性

-
-

示例

```
pixel.x = 120;
```

```
x
```

描述:

-
-

x 属性

-
-

示例

```
let x = pixel.x;
```

```
y
```

描述:

-
-

y 属性

-
-

示例

```
pixel.y = 60;
```

y

描述:

-
-

y 属性

-
-

示例

```
let y = pixel.y;
```

Point

new Point (lon,lat,alt)

描述:

-
-

Point 点坐标

-
-

示例

```
let p = new KMap.Point(112.455121,22.55764,0);
let p = new KMap.Point([114.12312,22.1312312],0);
let p = new KMap.Point({lon:114.12312,lat:22.1312312,alt:100});
let p = new KMap.Point('8w2a84b2g',100);
```

参数:

名称	类型	描述
lon	number, object, string	经度[-180-180],object{lon:number,lat:number},K码
lat	number	纬度[-90-90],高程(lon 传入非 number 时生效)
alt	number	高程, 单位米, 默认 0

函数

```
setAlt(alt)
```

描述:

-
-

设置高程

-
-

示例

```
p.setAlt(500);
```

参数:

名称	类型	描述
alt	number	高程

set(lon,lat,alt)

描述:

-
-

设置坐标

-
-

示例

p.set(114.123123,22.398981);

参数:

名称	类型	描述
lon	number	经度， -180-180
lat	number	纬度， -90-90
alt	number	高程， 单位米

toArray()

描述:

-
-

数组格式化坐标

-
-

示例

```
let array = p.toArray();
```

返回:

类型:

`Array.<number>`

toJSON()

描述:

-
-

JSON 格式化坐标

-
-

示例

```
let json = p.toJSON();
```

返回:

JSON 对象

类型:

`Object`

toString(*split*)

描述:

-
-

返回 lon, lat 格式字符串

-

-

示例

```
let str = p.toString();
```

参数:

名称	类型	描述
<code>split</code>	<code>string</code>	分隔字符串

返回:

格式坐标字符串

类型:

`string`

`equals(point)`

描述:

-
-

判断 2 个点是否相等

-
-

示例

```
let flag = p.equals(new KMap.Point(114.123123,22.398981));
```

参数:

名称	类型	描述
<code>point</code>	<code>Point</code>	比较点

返回:

是否相等

类型:

boolean,boolean

clone()

描述:

-
-

克隆

-
-

示例

```
let p2 = p.clone();
```

返回:

点对象

类型:

Point

属性

lon

描述:

-
-

获取经度

-
-

示例

```
let lon = p.lon;
```

lat

描述:

-
-

获取纬度

-
-

示例

```
let lat = p.lat;
```

alt

描述:

-
-

获取高程

-
-

示例

```
let alt = p.alt;
```

Size

new Size (*width,height*)

描述:

-
-

Size 像素大小

-
-

示例

```
let size = new KMap.Size(80,100);
```

参数:

名称	类型	描述
width	number	宽度
height	number	高度

函数

```
toString()
```

描述:

-
-

转成"width,height"格式字符串

-
-

示例

```
let str = size.toString();
```

返回:

字符串

类型:

string

```
toJSON()
```

描述:

-
-

转成对象

-
-

示例

```
let obj = size.toJSON();
```

返回:

`{"width":, "height":}`格式对象

类型:

`object`

clone()

描述:

-
-

克隆对象

-
-

示例

```
let newSize = size.clone();
```

返回:

新对象

类型:

`Size`

属性

width

描述:

-
-

width 属性

-
-

示例

```
size.width = 120;
```

width

描述:

-
-

width 属性

-
-

示例

```
let w = size.width;
```

height

描述:

-
-

height 属性

-
-

示例

```
size.height = 120;
```

height

描述:

-
-

height 属性

-
-

示例

```
let h = size.height;
```

地图

Map

new Map (container,props)

描述:

-
-

地图对象类，封装了地图的属性设置、图层、覆盖物，事件交互等接口的类。

-
-

示例

```
let map = new KMap.Map('container',{
  lang: 'zh-cn',
  center: new KMap.Point(114.02308, 22.53474),
  zoom: 18
});
let map = new KMap.Map('container', {
  viewer: new KMap.Viewer({
```

```

        center: new KMap.Point(114.03057485095093, 22.610380312958355,
0),
        viewMode: '3D',
        zoom: 15
    })
});
let map = new KMap.Map('container1', {
    lang: 'zh-cn',
    center: new KMap.Point(114.02308, 22.53474),
    zoom: 18,
    layers:[new KMap.SatelliteLayer({zIndex: 2})]
});

```

参数:

名称	类型	描述
container	HTMLElement, String	地图容器
props	propsObject	

propsObject :

名称	类型	描述
style	String	设置地图的显示样式，默认 kmap://style/bright, 1: 自定义 地图样式，如 "kmap://style/20210824104859173 4 2: 系统样式 kmap://style/bright,kmap://styl e/dark
interaction	propsInteractionObject	地图交互设置
Viewer	Viewer	视图控制器
viewMode	String	地图模式 3D 或 2D，默认: '2D'。如果

名称	类型	描述
		自定义视图控制器，则此参数无效
<code>minPitch</code>	Number	最小俯仰角，取值范围[0-80]，默认0，如果自定义视图控制器，则此参数无效
<code>maxPitch</code>	Number	最大俯仰角，取值范围[0-80]，默认80，如果自定义视图控制器，则此参数无效
<code>pitch</code>	Number	当前俯仰角，取值范围[0-80]，默认0，如果自定义视图控制器，则此参数无效
<code>minRotate</code>	Number	最小旋转角，取值范围[0-360]，默认0，如果自定义视图控制器，则此参数无效
<code>maxRotate</code>	Number	最大旋转角，取值范围[0-360]，默认360，如果自定义视图控制器，则此参数无效
<code>rotate</code>	Number	当前旋转角，取值范围[0-360]，默认0，如果自定义视图控制器，则此参数无效

名称	类型	描述
<code>minZoom</code>	<code>Number</code>	最小缩放级别，取值范围[4-21]，默认4，如果自定义视图控制器，则此参数无效
<code>maxZoom</code>	<code>Number</code>	最大缩放级别，取值范围[4-21]，默认21，如果自定义视图控制器，则此参数无效
<code>zoom</code>	<code>Number</code>	当前缩放级别，取值范围[4-21]，默认5，如果自定义视图控制器，则此参数无效
<code>limitBounds</code>	<code>Bounds</code>	视野限制，默认 <code>null</code> 。如果自定义视图控制器，则此参数无效
<code>center</code>	<code>Point, Array.<Number></code>	地图中心点，如果自定义视图控制器，并设置中心点，则此参数无效
<code>environment</code>	<code>String</code>	环境文件地址，默认空
<code>WebGLParams</code>	<code>Object</code>	额外配置的 WebGL 参数 eq: <code>preserveDrawingBuffer</code>
<code>sky</code>	<code>propsSkyObject</code>	天空，默认空

propsInteractionObject :

名称	类型	描述
<code>interactive</code>	<code>Boolean</code>	是否开启交互，默认 <code>true</code>
<code>animate</code>	<code>Boolean</code>	是否启用交互动画，默认 <code>true</code>
<code>inertia</code>	<code>Boolean</code>	是否启用交互惯性，默认 <code>true</code>
<code>mouseDragPan</code>	<code>Boolean</code>	是否启用鼠标拖拽，默认 <code>true</code>
<code>mouseDragRotate</code>	<code>Boolean</code>	是否启用鼠标旋转、俯仰，默认 <code>true</code>
<code>mouseDbClickZoom</code>	<code>Boolean</code>	是否启用鼠标双击缩放，默认 <code>true</code>
<code>wheelZoom</code>	<code>Boolean</code>	是否启用滚轮缩放，默认 <code>true</code>
<code>keyboardPan</code>	<code>Boolean</code>	是否启用键盘方向键移动，默认 <code>true</code>
<code>keyboardRotate</code>	<code>Boolean</code>	是否启用 <code>Ctrl</code> +键盘方向键旋转、俯仰，默认 <code>true</code>
<code>keyboardZoom</code>	<code>Boolean</code>	是否启用键盘+-键缩放，默认 <code>true</code>
<code>touchPan</code>	<code>Boolean</code>	是否启用触屏移动，默认 <code>true</code>
<code>touchZoomRotate</code>	<code>Boolean</code>	是否启用触屏旋转、俯仰，默认 <code>true</code>

名称	类型	描述
<code>zoomRounding</code>	<code>Boolean</code>	是否启用缩放地图取整，默认 <code>true</code>

propsSkyObject :

名称	类型	描述
<code>luminance</code>	<code>number</code>	天空亮度系数，取值范围 <code>[0, 1]</code> ，默认 <code>0.5</code> ，当设置 <code>renderer</code> 的 <code>toneMapping</code> 和 <code>toneMappingExposure</code> 参数，该值无效
<code>turbidity</code>	<code>number</code>	混浊度 影响天空的清晰度和亮度，取值范围 <code>[0, 20]</code> ，默认 <code>0</code>
<code>rayleigh</code>	<code>number</code>	散射强度，取值范围 <code>[0, 4]</code> ，默认 <code>0.5</code>
<code>mieCoefficient</code>	<code>number</code>	米氏散射系数 高米氏系数会使天空看起来更浑浊和暗淡，低米氏系数则会使天空更清晰和明亮，取值范围 <code>[0, 1]</code> ，默认 <code>0</code>
<code>mieDirectionalG</code>	<code>number</code>	米氏方向因子，默认 <code>0.5</code> ， 其中 <code>G = -1</code> 时表示完全后向散射， <code>G</code> 时表等射， <code>1</code> 时完全前向射 <code>[-1, 1]</code> ， <code>1</code> 值越高，天空中的款射光线就越集中在太阳所在的方向，天空呈现出更亮更明亮的效果，反之，值越低，散射光线则更均匀地分布在天空中，呈现出柔和、膜胧的效果

函数

requestAnimationFrame(*callback*)

描述:

-
-

添加任务到地图的执行队列

-
-

参数:

名称	类型	描述
callback	function	

返回:

任务 ID

cancelAnimationFrame(*id*)

描述:

-
-

移除任务到地图的执行队列

-
-

参数:

名称	类型	描述
id		id 为 requestAnimationFrame 返回的任务 id

resize()

描述:

-
-

刷新地图适应容器尺寸，地图大小改变是调用此方法

-
-

getMapContainer()

描述:

-
-

获取地图主容器

-
-

返回:

地图容器 DOM

类型:

HTMLElement

update()

描述:

-
-

刷新地图，地图/覆盖物/图层等改变地图未作出正确显示调用刷新方法

-

-

返回:

地图对象

类型:

Map

defaultLayers()

描述:

-
-

获取默认系统图层

-
-

示例

```
map.defaultLayers();
```

返回:

图层数组 eq:[BackgroundLayer,RoadNetLayer,BuildingLayer,LabelLayer]

类型:

Array.<Layer>

getLayers()

描述:

-
-

获取所有图层

-
-

示例

```
map.getLayers();
```

返回:

eq:[BackgroundLayer,RoadNetLayer,BuildingLayer,LabelLayer]

类型:

Array.<Layer>

getId()

描述:

-
-

获取地图 ID

-
-

示例

```
map.getId();
```

返回:

唯一 ID

类型:

String

setLang(str)

描述:

-
-

设置地图语言

-

-

示例

```
map.setLang('zh-cn');
```

参数:

名称	类型	描述
str	String	地图语言，当前支持 zh-cn、en

getLang()

描述:

-
-

获取地图当前语言

-
-

示例

```
map.getLang();
```

返回:

地图语言 eq: zh-cn , en

类型:

String

setStyle(value,theme,baseStyle)

描述:

-
-

设置地图表达样式

-
-

示例

```
map.setStyle('202108241048591734');
map.setStyle('202108241048591734',KMap.Theme.dark);
map.setStyle('kmap://style/dark');
map.setStyle('kmap://style/silvery',KMap.Theme.bright);
```

参数:

名称	类型	描述
value	object,String	设置地图的显示样式， 1：自定义地图样式，如 "202108241048591734 2：系统地图样式 如： kmap://style/dark
theme	String	地图风格， KMap.Theme.bright KMap.Theme.dark,主要设置 比例尺字样的颜色

```
setCustomStyle(value)
```

描述:

-
-

设置用户自定义地图表达样式

-
-

示例

```
map.setStyle(json);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
value	object	设置地图的显示样式， json 对象

add(target)

描述:

-
-

将对象添加到地图

-
-

示例

```
let measureTool = new KMap.MeasureTool({});
map.add(measureTool);

let ctrlCompass = new KMap.CompassControl();
map.add(ctrlCompass);

let roadLayer = new KMap.RoadNetLayer({zIndex: 3});
map.add(roadLayer);

let marker = new KMap.Marker({position: new KMap.Point(114.02308,
22.53474,0 ),content: '<div>这是 Dom Marker</div>',});
map.add(roadLayer);

let labelMarker = new KMap.LabelMarker({position : position,cursor:
'pointer',style : style,text:'凯立德地图',zIndex:1000});
map.add(roadLayer);
```

参数:

名称	类型	描述
target	Object	要添加的对象：控件、图层、覆盖物、测量、绘制、自定义动画

remove(target)

描述:

-
-

从地图中移除对象

-
-

示例

```
let measureTool = new KMap.MeasureTool({});
map.remove(measureTool);
```

参数:

名称	类型	描述
----	----	----

target	*	要移除的对象：控件、图层、覆盖物、测量、绘制、自定义动画
--------	---	------------------------------

has(target)

描述:

-
-

对象是否在地图中

-
-

示例

```
let measureTool = new KMap.MeasureTool({});
map.add(measureTool);
map.has(measureTool);
```

参数:

名称	类型	描述
target	*	对象：控件、图层、覆盖物、测量、绘制、自定义动画

```
open(target,position)
```

描述:

- -
- 打开信息窗口或右键菜单

-
-

参数:

名称	类型	描述
obj	InfoWindow, Contextmenu	信息窗口或右键菜单对象
position	Point	纬度度坐标 map.open(contextmenu,new KMap.Point(113.96893972,22.677957 64)); map.open(infoWindow,new KMap.Point(113.96893972,22.677957 64));

```
closeInfoWindow()
```

描述:

-
-

关闭地图已打开的信息窗口

-
-

示例

```
map.closeInfowindow();
```

closeContextmenu()

描述:

-
-

关闭地图已打开的右键菜单

-
-

示例

```
map.closeContextmenu();
```

close(target)

描述:

-
-

关闭指定信息窗口或右键菜单

-
-

示例

```
map.close(contextmenu);  
map.close(infoWindow);
```

参数:

名称	类型	描述
target	InfoWindow, Contextmenu	信息窗口或右键菜单对象

setInteraction(options)

描述:

-
-

设置交互参数

-
-

示例

```
map.setInteraction({wheelZoom: true,keyboardPan: false});
```

参数:

名称	类型	描述
options	Object	交互参数对象

setAnimate(b)

描述:

-
-

是否启用交互动画

-
-

示例

```
map.setAnimate(true);
```

参数:

名称	类型	描述
b	Boolean	是否启用

setInertia(*b*)

描述:

-
-

是否启用交互惯性

-
-

示例

```
map.setInertia(true);
```

参数:

名称	类型	描述
b	Boolean	是否启用

setInteractive(*b*)

描述:

-
-

设置是否开启鼠标、键盘、触屏交互

-
-

示例

```
map.setInteractive(true);
```

参数:

名称	类型	描述
b	Boolean	是否启用

```
setMouseAble(b)
```

描述:

-
-

设置是否启用鼠标拖拽平移，旋转

-
-

示例

```
map.setMouseAble(true);
```

参数:

名称	类型	描述
b	Boolean	是否启用

```
setWheelAble(b)
```

描述:

-
-

设置是否启用滚轮缩放地图

-
-

示例

```
map.setWheelAble(true);
```

参数:

名称	类型	描述
b	Boolean	是否启用

```
setTouchable(b)
```

描述:

-
-

设置是否启用触屏移动、旋转、缩放地图

-
-

示例

```
map.setTouchable(true);
```

参数:

名称	类型	描述
b	Boolean	是否启用

```
setKeyboardAble(b)
```

描述:

-
-

设置是否启用键盘平移，旋转，缩放

-
-

示例

```
map.setKeyboardAble(true);
```

参数:

名称	类型	描述
b	Boolean	是否启用

getInteraction()

描述:

-
-

获取地图交互设置

-
-

示例

```
map.getInteraction();
```

返回:

类型:

Object

getViewMode()

描述:

-

-

获取地图模式

-

-

示例

```
map.getViewMode();
```

返回:

3D 或者 2D

类型:

String

```
setZoom(z,duration,immediate)
```

描述:

-

-

设置地图缩放级别

-

-

示例

```
map.setZoom(10,500);  
map.setZoom(18,0);
```

参数:

名称	类型	描述
z	Number	地图缩放级别，取值范围[4-21]
duration	Number	动画过度时间，单位毫秒

名称	类型	描述
<code>immediate</code>	<code>Boolean</code>	是否跟随刷新地图，默认： <code>false</code>

`getZoom()`

描述:

-
-

获取地图缩放级别

-
-

示例

```
map.getZoom();
```

返回:

类型:

`Number`

`setCenterAndZoom(c,z,duration)`

描述:

-
-

设置中心点与缩放级别

-
-

示例

```
map.setCenterAndZoom([116.478935,39.997761],15,500);  
map.setCenterAndZoom(new KMap.Point(116.478935,39.997761),18,0);  
map.setCenterAndZoom(new KMap.Point(116.478935,39.997761),18);
```

参数:

名称	类型	描述
c	Point, Array.<Number>	地图中心点
z	Number	缩放级别，取值范围[4-21]
duration	Number	动画过度时间，如果不传，则由系统计算

setZoomLimit(min,max)

描述:

-
-

设置地图缩放限制

-
-

示例

```
map.setZoomLimit(8,18);
```

参数:

名称	类型	描述
min	Number	最小级别，取值范围[4-21]
max	Number	最大级别，取值范围[4-21]

getZoomLimit()

描述:

-
-

获取地图缩放限制

-
-

示例

```
map.getZoomLimit()
```

返回:

类型:

`Array.<Number>,undefined`

clearZoomLimit()

描述:

-
-

取消地图缩放限制

-
-

示例

```
map.clearZoomLimit()
```

setCenter(*p,duration,immediate*)

描述:

-
-

设置地图中心点

-
-

示例

```
map.setCenter([116.478935,39.997761],500)
map.setCenter(new KMap.Point(116.478935,39.997761,0),500)
```

参数:

名称	类型	描述
p	Point, Array.<Number>	中心点
duration	Number	动画过度时间
immediate	Boolean	是否跟随刷新地图，默认值 false

```
setRegion(str,duration)
```

描述:

-
-

按照行政区名称或 adcode 来设置地图显示的中心点。

-
-

示例

```
map.setRegion("深圳市",500);
map.setRegion("440300",0);
```

参数:

名称	类型	描述
str	String	行政区名称或编码

名称	类型	描述
<code>duration</code>	<code>Number</code>	动画过度时间，如果不传，则由系统计算

`getCenter()`

描述:

-
-

获取地图中心点

-
-

示例

```
map.getCenter();
```

返回:

类型:

`Point`

`setBoundsLimit(bounds)`

描述:

-
-

设置视野限制

-
-

示例

```
map.setBoundsLimit(new KMap.Bounds(new  
KMap.Point(113.96893972,22.67795764),new  
KMap.Point(114.01196614,22.63031712)));
```

参数:

名称	类型	描述
<code>bounds</code>	<code>Bounds</code>	范围框

`clearBoundsLimit()`

描述:

-
-

清除视野限制

-
-

示例

```
map.clearBoundsLimit();
```

`getBoundsLimit()`

描述:

-
-

获取视野限制

-
-

示例

```
map.getBoundsLimit();
```

返回:

类型:
`Bounds`

setPitch(*p,duration,immediate*)

描述:

-
-

设置地图俯仰角

-
-

示例

```
map.setPitch(15, 500);
map.setPitch(15, 0);
```

参数:

名称	类型	描述
p	Number	俯仰角，取值范围[0-80]
duration	Number	动画过度时间，毫秒
immediate	Boolean	是否跟随刷新地图，默认值: false

getPitch()

描述:

-
-

获取地图俯仰角

-
-

示例

```
map.getPitch();
```

返回:

类型:
Number

```
setPitchLimit(min,max)
```

描述:

-
-

设置俯仰角度限制

-
-

示例

```
map.setPitchLimit(10, 45);
```

参数:

名称	类型	描述
----	----	----

min	Number	俯仰最小角度，取值范围[0-80]
-----	--------	-------------------

max	Number	俯仰最大角度，取值范围[0-80]
-----	--------	-------------------

```
getPitchLimit()
```

描述:

-
-

获取俯仰角度限制

-
-

示例

```
map.getPitchLimit();
```

返回:

类型:

`Array.<Number>`

clearPitchLimit()

描述:

-
-

清除俯仰角度限制

-
-

示例

```
map.clearPitchLimit();
```

setRotation(*p,duration,immediate*)

描述:

-
-

设置地图旋转角

-
-

示例

```
map.setRotation(45, 0);  
map.setRotation(45, 500);
```

参数:

名称	类型	描述
<code>p</code>	Number	旋转角度，取值范围 [0-360]
<code>duration</code>	Number	动画过度时间，毫秒
<code>immediate</code>	Boolean	是否跟随刷新地图，默认值: <code>false</code>

```
getRotation()
```

描述:

-
-

获取地图旋转角

-
-

示例

```
map.getRotation();
```

返回:

类型:
Number

```
setRotateLimit(min,max)
```

描述:

-
-

设置旋转角度限制

-

-

示例

```
map.setRotateLimit(4,21);
```

参数:

名称	类型	描述
min	Number	最小角度，取值范围[0-360]
max	Number	最大角度，取值范围[0-360]

```
clearRotateLimit()
```

描述:

-
-

清除旋转角度限制

-
-

示例

```
map.clearRotateLimit();
```

```
getRotateLimit()
```

描述:

-
-

获取旋转角度限制

-
-

示例

```
map.getRotateLimit();
```

返回:

类型:
Array.<Number>

```
getFitZoomAndCenterByOverlays(overlays,avoid,maxZoom)
```

描述:

-
-

根据 overlays 计算出合适的中心点和 zoom 级别

-
-

参数:

名称	类型	描述
points		
avoid		[6,6,6,6]

返回:

类型:
Object

```
setFitView(overlays,avoid,maxZoom,immediately)
```

描述:

-
-

根据地图上添加的覆盖物分布情况，自动缩放地图到合适的视野级别，参数均可缺省。 overlayList 为覆盖物数组，缺省时为当前地图上添加的所有覆盖物图层，

-
-

参数:

名称	类型	描述
p		

```
flyTo(center,zoom)
```

描述:

-
-

飞行到一个纬度度坐标点，并设置地图比例尺

-
-

示例

```
map.flyTo(new KMap.Point(112.455121,22.55764,0), 15)
```

参数:

名称	类型	描述
center	Point, Array.<Number>	纬度度坐标点
zoom	Number	比例尺

```
getBounds()
```

描述:

-
-

获取地图视野

-
-

示例

```
map.getBounds();
```

返回:

范围框

类型:

Bounds

```
setCursor(cursor)
```

描述:

-
-

设置鼠标样式

-
-

示例

```
map.setCursor('pointer');
```

参数:

名称	类型	描述
cursor	String, Url	鼠标样式

getCenterLocation(callback)

描述:

-
-

获取中心点位置信息

-
-

示例

```
map.getCenterLocation((e) => { console.log(e);})
```

参数:

名称	类型	描述
callback	function	回调方法

setBackgroundColor(color,alpha)

描述:

-
-

设置画布背景色

-
-

示例

```
map.setBackgroundColor("#000000",0.5);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
color	String	十六进制 HTML 色彩代码
alpha	Number	透明度

```
setBackgroundGrid(imgUrl,color)
```

描述:

-
-
- 设置凯立德数据图背景栅格
-
-

示例

```
map.setBackgroundGrid("../images/cellgrid.png","#fcf9f2");
```

参数:

名称	类型	描述
imgUrl	String	图片地址
color	String	十六进制 HTML 色彩代码

```
removeBackgroundGrid()
```

描述:

-
-
- 移除凯立德数据图背景栅格

-
-

示例

```
map.removeBackgroundGrid();
```

setEnvironment(url)

描述:

-
-

设置环境

-
-

示例

```
map.setEnvironment('/examples/textures/equirectangular/blouberg_sunrise_2_1k.hdr')
```

参数:

名称	类型	描述
url	String	环境地址

setSky(sky)

描述:

-
-

设置天空

-
-

示例

```
map.setSky({luminance:0.5})
```

参数:

名称	类型	描述
sky	Object	

```
getMapImage()
```

描述:

-
-

获取当前地图的图片数据，需要初始化地图设置

“WebGLParams: {preserveDrawingBuffer:true}” 参数

-
-

示例

```
map.getMapImage();
```

```
clearOverlays()
```

描述:

-
-

清除所有覆盖物

-
-

示例

```
map.clearOverlays();
```

事件

complete()

描述:

-
-

地图加载完成事件

-
-

movestart()

描述:

-
-

地图移动开始事件

-
-

参数:

名称	类型	描述
----	----	----

e	Object	结果
---	--------	----

moving()

描述:

-
-

地图移动中事件事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
moveend()
```

描述:

-
-

地图移动结束事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
pitchstart()
```

描述:

-
-

地图俯仰角改变开始事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
pitchchange()
```

描述:

-
-

地图俯仰角改变事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
pitchend()
```

描述:

-
-

地图俯仰角改变结束事件

-
-

参数:

名称	类型	描述
e	Object	结果

rotatestart()

描述:

-
-

地图旋转开始事件

-
-

参数:

名称	类型	描述
e	Object	结果

rotatechange()

描述:

-
-

地图旋转中事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

```
rotateend()
```

描述:

-
-

地图旋结束事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
zoomstart()
```

描述:

-
-

地图缩放开始事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

zoomchange()

描述:

-
-

地图缩放中事件

-
-

参数:

名称	类型	描述
e	Object	结果

zoomend()

描述:

-
-

地图缩放结束事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

dragstart()

描述:

-
-

拖拽地图开始事件

-
-

参数:

名称	类型	描述
e	Object	结果

dragging()

描述:

-
-

拖拽地图中事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

dragend()

描述:

-
-

拖拽地图结束事件

-
-

参数:

名称	类型	描述
e	Object	结果

mousemove()

描述:

-
-

鼠标在地图上移动事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

```
click()
```

描述:

-
-

单机地图事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
dblclick()
```

描述:

-
-

双击地图事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

```
mouseover()
```

描述:

-
-

鼠标移入地图事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
mouseout()
```

描述:

-
-

鼠标移出地图事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

```
resize()
```

描述:

-
-

地图大小改变事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
destroy()
```

描述:

-
-

销毁地图事件

-
-

Viewer

new Viewer (props)

描述:

-
-

视图控制器

-
-

示例

```
let viewer = new KMap.Viewer({
  center: new KMap.Point(112.455121,22.55764,0),
  viewMode: '3D',
  zoom: 15
})
let map = new KMap({
  viewer: viewer
})
```

参数:

名称	类型	描述
props	propsObject	

propsObject :

名称	类型	描述
viewMode	String	地图模式 3D 或 2D
minPitch	Number	最小俯仰角
maxPitch	Number	最大俯仰角
pitch	Number	当前俯仰角

名称	类型	描述
<code>minRotate</code>	Number	最小旋转角
<code>maxRotate</code>	Number	最大旋转角
<code>rotate</code>	Number	当前旋转角
<code>minZoom</code>	Number	最小缩放级别
<code>maxZoom</code>	Number	最大缩放级别
<code>zoom</code>	Number	当前缩放级别
<code>limitBounds</code>	Bounds	视野限制
<code>center</code>	Point, Array.<Number>	地图中心点

动画

CustomAnimate

```
new CustomAnimate (map,duration,method)
```

描述:

-
-

用户自定义动画

-
-

示例

```
let customAnimate = new KMap.CustomAnimate(map,1000,(t)=>{ //do something });
```

参数:

名称	类型	描述
map	Map	地图对象
duration	number	动画执行时长，单位毫秒，等于 0 一直执行
method	function	动画执行方法

函数

start()

描述:

-
-

开始执行动画

-
-

示例

```
customAnimate.start();
```

resume()

描述:

-

-

恢复动画

-
-

示例

```
customAnimate.resume();
```

pause()

描述:

-
-

暂停动画

-
-

示例

```
customAnimate.pause();
```

stop()

描述:

-
-

停止动画

-
-

示例

```
customAnimate.stop();
```

destroy()

描述:

-
-

销毁对象 customAnimate.destroy(); customAnimate = null;

-
-

MoveAnimate

new MoveAnimate (map,props)

描述:

-
-

覆盖物移动

-
-

示例

```
let moveAnimate = new KMap.MoveAnimate(map, {rate: 1});
```

参数:

名称	类型	描述
----	----	----

map	Map	地图对象
-----	-----	------

props	propsObject	
-------	-------------	--

propsObject :

名称	类型	描述
----	----	----

名称	类型	描述
rate	Number	初始播放速率
status	Number	初始状态，0：暂停，1：播放，默认 1

函数

```
setPlayRate(target,rate)
```

描述:

-
-
- 设置覆盖物播放速率
-
-

示例

```
moveAnimate.setPlayRate(marker,2);
```

参数:

名称	类型	描述
target	Marker, LabelMarker, Gltf	覆盖物
rate	Number	速度倍率，1x,2x

```
setAllPlayRate(rate)
```

描述:

-

-

设置所有覆盖物播放速率

-

-

示例

```
moveAnimate.setAllPlayRate(2);
```

参数:

名称	类型	描述
rate	Number	速度倍率，1x, 2x

```
setStart(target)
```

描述:

-

-

设置开始播放

-

-

示例

```
moveAnimate.setStart(marker);
```

参数:

名称	类型	描述
target	Marker, LabelMarker, Gltf	覆盖物对象

```
setAllStart()
```

描述:

-
-

设置所有开始

-
-

示例

```
moveAnimate.setAllStart();
```

setStop(target)

描述:

-
-

设置停止

-
-

示例

```
moveAnimate.setStop(marker);
```

参数:

名称	类型	描述
target	Marker, LabelMarker, Gltf	覆盖物对象

setAllStop()

描述:

-
-

设置所有停止

-
-

示例

```
moveAnimate.setAllStop();
```

```
setPause(target)
```

描述:

-
-

设置暂停

-
-

示例

```
moveAnimate.setPause(marker);
```

参数:

名称	类型	描述
target	Marker, LabelMarker, Gltf	覆盖物对象

```
setAllPause()
```

描述:

-
-

设置所有暂停

-
-

示例

```
moveAnimate.setAllPause();
```

setResume(target)

描述:

-
-

设置恢复

-
-

示例

```
moveAnimate.setResume(marker);
```

参数:

名称	类型	描述
target	Marker, LabelMarker, Gltf	覆盖物对象

setAllResume()

描述:

-
-

设置所有恢复

-
-

示例

```
moveAnimate.setAllResume();
```

moveTo(target,position,opts)

描述:

-
-

覆盖物移动至

-
-

示例

```
let marker = new KMap.Marker({position:new  
KMap.Point(113.92389,22.50227528,0),...})  
map.add(marker);  
marker.on("moving",(e)=>{});  
marker.on("moveend",(e)=>{});  
moveAnimate.moveTo(marker,new  
KMap.Point(113.92400417,22.502585,0),{duration: 10000, easing:  
KMap.Easing.easeOut});
```

参数:

名称	类型	描述
target	Marker, LabelMarker, Gltf	覆盖物对象
position	Point	经纬度坐标
opts	optsObject	

optsObject :

名称	类型	描述
duration	Number	时长, 单位毫秒
easing	function	缓动函数, 默认: true

moveAlong(target,path,opts)

描述:

-
-

覆盖物沿线移动

-
-

示例

```
let marker = new KMap.Marker({.....})
map.add(marker);
marker.on("moving", (e)=>{});
marker.on("moveend", (e)=>{});
marker.on("movealong", (e)=>{});
let path = [{"position": new
KMap.Point(113.92389,22.50227528,0),"duration": 0},{ "position": new
KMap.Point(113.92400417,22.502585,0),"duration": 728},{ "position": new
KMap.Point(113.92408528,22.50269083,0),"duration": 287},{ "position":
new KMap.Point(113.92475472,22.50452806,0),"duration": 4311}]
moveAnimate.moveAlong(marker,path,{loop: false,rate:8});
```

参数:

名称	类型	描述
target	Marker, LabelMarker, Gltf	覆盖物对象
path	Array. <pathObject>	
opts	optsObject	

pathObject :

名称	类型	描述
duration	Number	前一个位置到此位置时长，单位毫秒
position	Point	纬度度坐标

optsObject :

名称	类型	描述
autoRotation	Boolean	是否沿路径旋转，默认 true
rate	Number	播放速度倍率
loop	Boolean	是否循环播放，默认 false

setAlongPath(target,path)

描述:

-
-

设置沿线移动点

-
-

示例

```
let path = [{"position": new KMap.Point(113.92389,22.50227528,0),"duration": 0}, {"position": new KMap.Point(113.92400417,22.502585,0),"duration": 728}, {"position": new KMap.Point(113.92408528,22.50269083,0),"duration": 287}, {"position": new KMap.Point(113.92475472,22.50452806,0),"duration": 4311}]
moveAnimate.setAlongPath(marker,path);
```

参数:

名称	类型	描述
target	Marker, LabelMarker, Gltf	覆盖物对象
path	Array. <pathObject>	

pathObject :

名称	类型	描述
<code>duration</code>	<code>Number</code>	前一个位置到此位置时长，单位毫秒
<code>position</code>	<code>Point</code>	纬度度坐标

```
remove(target)
```

描述:

-
-

移除覆盖物

-
-

示例

```
moveAnimate.remove(marker);
```

参数:

名称	类型	描述
<code>target</code>	<code>Marker, LabelMarker, Gltf</code>	覆盖物对象

```
removeAll()
```

描述:

-
-

移除所有覆盖物

-

-

示例

```
moveAnimate.removeAll();
```

destroy()

描述:

-
-

销毁覆盖物移动对象

-
-

示例

```
moveAnimate.destroy();  
moveAnimate = null;
```

TrackAnimation

new TrackAnimation (map,props)

描述:

-
-

轨迹播放

-
-

示例

```
let tmBaseTime = KMap.Util.now();  
let trackAnimation = new KMap.TrackAnimate(map,{  
  rate:1,  
  playTime:(t) => {  
  },  
  timeRange:(t) => {  
    if (t>=tmBaseTime && t<=(tmBaseTime+60000*3)) {
```

```

        return true
    } else {
        return false;
    }
},
maintainSpan:1
});

```

参数:

名称	类型	描述
map	Map	地图对象
props	propsObject	

propsObject :

名称	类型	描述
rate	Number	初始播放速度倍率
timeRange	function	用于插入轨迹节点判断和维护轨迹节点，返回值： true 在时间段内，保留轨迹点，返回值： false 在时间段外，丢弃轨迹点。例：(t) => {if (t>Date.now()-60000 && t<Date.now()+60000) {return true; } else {return false} }
maintainSpan	Number	单位分钟，维护数据时间间隔,大于等于 1 则维护，否则，不维护。
playTime	function	轨迹回放时间轴回调方法,例： (t) => {console.log(t)}

函数

addTarget(target,points)

描述:

-
-

添加覆盖物

-
-

示例

```
let marker = new KMap.Marker({.....})
map.add(marker);
trackAnimation.addTarget(marker,[{"position":new
KMap.Point(113.92389,22.50227528,0),"time":1676282068769},{ "position":n
ew
KMap.Point(113.92400417,22.502585,0),"time":1676282069497},{ "position":
new KMap.Point(113.92408528,22.50269083,0),"time":1676282069784}]]);
```

参数:

名称	类型	描述
target	Marker, LabelMarker, Gltf	覆盖物对象
points	Array.<pointsObject>	

pointsObject :

名称	类型	描述
position	Point	纬度度坐标
time	number	坐标时间，单位毫秒
angle	number	角度，正北顺时针，单位度。可选项。如果传入，则采用传入角度对目标旋转，否则由轨迹点计算得出

setPoints(target,points)

描述:

-
-

设置轨迹点

-
-

示例

```
trackAnimation.setPoints(marker,[{"position":new KMap.Point(113.92389,22.50227528,0),"time":1676282068769}, {"position":new KMap.Point(113.92400417,22.502585,0),"time":1676282069497}, {"position":new KMap.Point(113.92408528,22.50269083,0),"time":1676282069784}]);
```

参数:

名称	类型	描述
target	Marker, LabelMarker, Gltf	覆盖物对象
points	Array.<pointsObject>	

pointsObject :

名称	类型	描述
position	Point	纬度度坐标
time	number	坐标时间，单位毫秒
angle	number	角度，正北顺时针，单位度。可选项。如果传入，则采用传入角度对目标旋转，否则由轨迹点计算得出

appendPoints(target,points)

描述:

-
-

增加轨迹点，常用于实时监控

-
-

示例

```
trackAnimation.appendPoints(marker,[{"position":new KMap.Point(113.92475472,22.50452806,0),"time":1676282074095}])
```

参数:

名称	类型	描述
target	Marker, LabelMarker, Gltf	覆盖物对象
points	Array.<pointsObject>	

pointsObject :

名称	类型	描述
position	Point	纬度度坐标
time	number	坐标时间，单位毫秒
angle	number	角度，正北顺时针，单位度。可选项。如果传入，则采用传入角度对目标旋转，否则由轨迹点计算得出

removeTarget(target)

描述:

-
-

移除覆盖物

-
-

示例

```
trackAnimation.removeTarget(marker);
```

参数:

名称	类型	描述
target	Marker, LabelMarker, Gltf	覆盖物对象

```
getTarget(target)
```

描述:

-
-

获取覆盖物

-
-

示例

```
trackAnimation.getTarget(marker);
```

参数:

名称	类型	描述
target	Marker, LabelMarker, Gltf	覆盖物对象

setRate(rate)

描述:

-
-

设置速率

-
-

示例

```
trackAnimation.setRate(4);
```

参数:

名称	类型	描述
rate	Number	速度倍率，1x 2x

setTime(t)

描述:

-
-

设置轨迹回放时间点

-
-

参数:

名称	类型	描述
t	Number	时间，单位毫秒

名称	类型	描述
----	----	----

```
trackAnimation.setTime(1676282069784);
```

start()

描述:

-
-

开始

-
-

示例

```
trackAnimation.start();
```

stop()

描述:

-
-

停止

-
-

示例

```
trackAnimation.stop();
```

destroy()

描述:

-

-

销毁对象

-

-

示例

```
trackAnimation.destroy();  
trackAnimation = null;
```

控件

CtrlCompass

```
new CtrlCompass (options)
```

描述:

-

-

罗盘控件

-

-

示例

```
let ctrlCompass = new KMap.CompassControl();  
let ctrlCompass = new KMap.CompassControl({position:{right:"100px",bottom:"100px"}});  
map.add(ctrlCompass);
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
<code>position</code>	<code>optionsPositionObject</code>	控件停靠位置，默认值： <code>{bottom:"130px",right:"6px"}</code>
<code>offset</code>	<code>Pixel</code>	相对于地图容器左上角的偏移量，正数代表向右下偏移，默认值： <code>Null</code>

optionsPositionObject :

名称	类型	描述
<code>top</code>	<code>String</code>	距离顶部，如像素值： <code>"120px"</code>
<code>left</code>	<code>String</code>	距离左边，如像素值： <code>"25px"</code>
<code>right</code>	<code>String</code>	距离右边，如像素值： <code>"25px"</code>
<code>bottom</code>	<code>String</code>	距离底部，如像素值： <code>"500px"</code>

函数

```
show()
```

描述:

-
-

显示

-
-

示例

```
ctrlCompass.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
ctrlCompass.hide();
```

destroy()

描述:

-
-

销毁

-
-

示例

```
ctrlCompass.destroy();  
ctrlCompass = null;
```

CtrlScale

new CtrlScale (options)

描述:

-

-

控制比例尺显示位置和显示/隐藏的空间

-

-

参数:

名称	类型	描述
<i>optionsObject :</i>		
名称	类型	描述
position	optionsPositionObject	控件停靠位置，默认值： {bottom:"28px",left:"5px"}
offset	Pixel	相对于地图容器左上角的偏移量，正数代表向右下偏移，默认值：Null let ctrScale = new KMap.ScaleControl(); let ctrScale = new KMap.ScaleControl({position:{left:"5px",bottom:"5px"}}) map.add(ctrScale);
<i>optionsPositionObject :</i>		
名称	类型	描述
top	String	距离顶部，如像素值："120px"
left	String	距离左边，如像素值："25px"
right	String	距离右边，如像素值："25px"

名称	类型	描述
bottom	String	距离底部，如像素值: "500px"

函数

```
show()
```

描述:

-
-

显示

-
-

示例

```
ctrScale.show();
```

```
hide()
```

描述:

-
-

隐藏

-
-

示例

```
ctrScale.hide();
```

```
destroy()
```

描述:

-
-

销毁

-
-

示例

```
ctrScale.destroy();  
ctrScale = null;
```

CtrlZoom

new CtrlZoom (*options*)

描述:

-
-

缩放控件

-
-

参数:

名称	类型	描述
----	----	----

options	optionsObject	
---------	---------------	--

optionsObject :

名称	类型	描述
----	----	----

position	optionsPositionObject	控件停靠位置，默认值: {bottom: "60px", right: "25px"}
----------	-----------------------	---

offset	Pixel	相对于地图容器左上角的偏移量，正数
--------	-------	-------------------

名称	类型	描述
		代表向右下偏移，默认值：Null let ctrlZoom = new KMap.ZoomControl(); let ctrlZoom = new KMap.ZoomControl({offset: new KMap.Pixel(100, 100)}); map.add(ctrlZoom);

optionsPositionObject :

名称	类型	描述
top	String	距离顶部，如像素值： "120px"
left	String	距离左边，如像素值： "25px"
right	String	距离右边，如像素值： "25px"
bottom	String	距离底部，如像素值： "500px"

函数

```
show()
```

描述:

-
-

显示

-
-

示例

```
ctrlZoom.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
ctrlZoom.hide();
```

destroy()

描述:

-
-

销毁

-
-

示例

```
ctrlZoom.destroy();  
ctrlZoom = null;
```

图层/用户图层

CanvasLayer

new CanvasLayer (*options*)

描述:

-
-

画布图层

-
-

示例

```
let canvasLayerCanvas = document.createElement('canvas');
canvasLayerCanvas.width = canvasLayerCanvas.height = 200;
let canvasLayer = new KMap.CanvasLayer({
  bounds : new KMap.Bounds(new
KMap.Point(113.98169289,22.51295288),new
KMap.Point(114.11733939,22.65318792)),
  style : {
    borderStyle: 'dashed',
    borderDasharray : [15,5],
    borderWidth:1,
    borderColor:"rgba(0,13,255,0.5)",
    fillColor: "rgba(255,0,0,0.5)",
    posStyle: 0,
    repeatWidth:0,
    repeatHeight:0,
  },
  canvas: canvasLayerCanvas
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
bounds	Bounds	范围信息

名称	类型	描述
<code>canvas</code>	<code>object</code>	实例化的画布对象
<code>style</code>	<code>PolygonStyle</code>	多边形样式表达

函数

```
setCanvas(canvas)
```

描述:

-
-

设置画布信息

-
-

示例

```
let canvasLayerCanvas = document.createElement('canvas');
canvasLayer.setCanvas(canvasLayerCanvas);
```

参数:

名称	类型	描述
<code>canvas</code>	<code>object</code>	画布对象

```
getCanvas()
```

描述:

-
-

获取画布信息

-
-

示例

```
let canvas = canvasLayer.getCanvas();
```

返回:

画布对象

类型:

Object

setStyle(style)

描述:

-
-

设置表达样式

-
-

示例

```
canvasLayer.setStyle({
  borderStyle: 'dashed',
  borderDasharray : [15,5],
  borderWidth:1,
  borderColor:"rgba(0,13,255,0.5)",
  fillColor: "rgba(255,0,0,0.5)",
  posStyle: 0,
  repeatWidth:0,
  repeatHeight:0,
});
```

参数:

名称	类型	描述
style	PolygonStyle	多边形样式表达

getStyle()

描述:

-
-

获取表达样式

-
-

示例

```
let style = canvasLayer.getStyle();
```

返回:

多边形样式表达

类型:
PolygonStyle

setBounds(bounds)

描述:

-
-

设置范围

-
-

示例

```
canvasLayer.setBounds(new KMap.Bounds(new KMap.Point(113.98169289,22.51295288),new KMap.Point(114.11733939,22.65318792)));
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
bounds	Bounds	范围对象

getBounds()

描述:

-
-

获取范围

-
-

示例

```
let bounds = canvasLayer.getBounds();
```

返回:

范围对象

类型:
Bounds

refresh()

描述:

-
-

刷新图层

-
-

示例

```
canvasLayer.refresh();
```

setzIndex(zIndex)

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setzIndex(100);
```

参数:

名称	类型	描述
zIndex	Number	层级，1-59

setOpacity(opacity)

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

show()

描述:

-
-

显示

-
-

示例

```
layer.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-

-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
zooms	Array.<Number>	显示范围, [min,max], 如[10,16]在 10-16 比例尺级别显示

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getzIndex();
```

返回:

层级

类型:

number

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

Array.<number>

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

boolean

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型:

number

update()

描述:

-
-

更新图层

-
-

示例

```
layer.update();
```

render()

描述:

-
-

渲染图层

-
-

示例

```
layer.render();
```

属性

id

描述:

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

DataLayer

new DataLayer (*options*)

描述:

-
-

数据图层

-
-

示例

```
let dataLayer = new KMap.DataLayer({
  source : dataSource,
  zooms : [16,21],
  polygonStyle : [{
    fillColor: "rgba(103,248,128,0.89)",
    borderColor: "rgba(16,71,28,0.86)",
    borderDasharray: [10, 5],
    borderWidth: 3,
    borderStyle: "solid", //dashed
  },{
    fillColor: "rgba(82,142,246,0.89)",
    borderColor: "rgba(23,48,108,0.86)",
    borderDasharray: [10, 5],
    borderWidth: 3,
    borderStyle: "solid", //dashed
  }],
  parseCallBack: function (feature) {
    if(feature.properties.id % 2 == 0) {
      return 1;
    }
  }
})
```

```
        return 0;
    }
});
```

参数:

名称	类型	描述
<code>options</code>	<code>optionsObject</code>	

optionsObject :

名称	类型	描述
<code>id</code>	<code>String</code>	图层 ID
<code>zIndex</code>	<code>Number</code>	图层层级，1-59
<code>zooms</code>	<code>Array.<Number></code>	图层显示比例尺级别，如 [10,16]在 10-16 比例尺级别显示
<code>opacity</code>	<code>Number</code>	图层透明度，0-1，0 为全透明，1 为不透明，默认 1
<code>visible</code>	<code>boolean</code>	图层是否显示，默认 true
<code>source</code>	<code>DataSource</code>	数据源
<code>pointStyle</code>	<code>Array.<PointStyle></code>	标注样式数组
<code>polylineStyle</code>	<code>Array.<PolylineStyle></code>	线样式数组

名称	类型	描述
<code>polygonStyle</code>	<code>Array.<PolygonStyle></code>	面样式数组
<code>parseCallBack</code>	<code>ParseCallback</code>	数据解析设置样式回调，返回样式索引

函数

```
setSource(source)
```

描述:

-
-

设置 Data 来源

-
-

示例

```
dataLayer.setSource(dataSource);
```

参数:

名称	类型	描述
<code>source</code>	<code>DataSource</code>	数据源

```
setParseCallBack(callback)
```

描述:

-
-

设置数据解析设置样式回调，返回样式索引

-
-

示例

```
dataLayer.setParseCallBack(function(feature) {  
    return 0;  
});
```

参数:

名称	类型	描述
callback	ParseCallback	回调,参数 feature,返回 styleIndex

```
setStyle(type,style,zoom)
```

描述:

-
-

设置样式

-
-

示例

```
dataLayer.setStyle(KMap.DataLayer.TYPE.polygon,[{  
    fillColor: "rgba(103,248,128,0.89)",  
    borderColor: "rgba(16,71,28,0.86)",  
    borderDasharray: [10, 5],  
    borderWidth: 3,  
    borderStyle: "solid", //dashed  
}]);
```

参数:

名称	类型	描述
type	string	数据类型,

名称	类型	描述
		KMap.DataLayer.TYPE
style	Array.<PointStyle>, Array.<PolylineStyle>, Array.<PolygonStyle>	样式对象数组
zoom	number, undefined	级别，设置 0 为所有级别，默认 0

getStyle(type, zoom)

描述:

-
-

获取样式

-
-

示例

```
dataLayer.getStyle(KMap.DataLayer.TYPE.polygon);
```

参数:

名称	类型	描述
type	string	数据类型，KMap.DataLayer.TYPE
zoom	number, undefined	级别，设置 0 为所有级别，默认 0

setzIndex(zIndex)

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setzIndex(100);
```

参数:

名称	类型	描述
zIndex	Number	层级，1-59

setOpacity(opacity)

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

show()

描述:

-
-

显示

-
-

示例

```
layer.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-

-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
zooms	Array.<Number>	显示范围, [min,max], 如[10,16]在 10-16 比例尺级别显示

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getzIndex();
```

返回:

层级

类型:

number

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

Array.<number>

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

boolean

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型:

number

update()

描述:

-
-

更新图层

-
-

示例

```
layer.update();
```

render()

描述:

-
-

渲染图层

-
-

示例

```
layer.render();
```

属性

id

描述:

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

事件

```
mouseover/mouseout/click/rightclick/bdclick()
```

描述:

-
-

数据事件

-
-

参数:

名称	类型	描述
originEvent	Object	原始事件
pixel	Pixel	像素坐标
position	Point	地图坐标
target	targetObject	目标

targetObject :

名称	类型	描述
type	String	类型
feature	Feature	GeoJson.Feature

ImageLayer

new ImageLayer (options)

描述:

-
-

图片图层

-
-

示例

```
let imageLayer = new KMap.ImageLayer({
  url : './images/yangtaishan.png',
  bounds : new KMap.Bounds(new KMap.Point(113.96893972,22.67795764),new KMap.Point(114.01196614,22.63031712)),
  zooms: [13, 20]
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
----	----	----

名称	类型	描述
<code>id</code>	<code>String</code>	图层 ID
<code>zIndex</code>	<code>Number</code>	图层层级，1-59
<code>zooms</code>	<code>Array.<Number></code>	图层显示比例尺级别，如 <code>[10,16]</code> 在 10-16 比例尺级别显示
<code>opacity</code>	<code>Number</code>	图层透明度，0-1，0 为全透明，1 为不透明，默认 1
<code>visible</code>	<code>boolean</code>	图层是否显示，默认 <code>true</code>
<code>url</code>	<code>String</code>	图片地址
<code>bounds</code>	<code>Bounds</code>	图片范围框

函数

`setUrl(url)`

描述:

-
-

设置图片地址

-
-

示例

```
imageLayer.setUrl('./images/yangtaishan.png');
```

参数:

名称	类型	描述
url	String	地址

```
getUrl()
```

描述:

-
-

获取图片地址

-
-

示例

```
imageLayer.getUrl();
```

返回:

地址

类型:

String

```
setBounds(bounds)
```

描述:

-
-

设置范围

-

-

示例

```
imageLayer.setBounds(new KMap.Bounds(new KMap.Point(113.96893972,22.67795764),new KMap.Point(114.01196614,22.63031712)));
```

参数:

名称	类型	描述
bounds	Bounds	图片范围框

```
getBounds()
```

描述:

-
-

获取范围

-
-

示例

```
imageLayer.getBounds();
```

返回:

图片范围框

类型:
Bounds

```
setzIndex(zIndex)
```

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setZIndex(100);
```

参数:

名称	类型	描述
zIndex	Number	层级，1-59

setOpacity(*opacity*)

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

show()

描述:

-
-

显示

-
-

示例

```
layer.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-
-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
zooms	Array.<Number>	显示范围，[min,max]，如[10,16]在 10-16 比例尺级别显示

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getZIndex();
```

返回:

层级

类型:

number

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

Array.<number>

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

boolean

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型:

number

update()

描述:

-
-

更新图层

-
-

示例

```
layer.update();
```

render()

描述:

-
-

渲染图层

-
-

示例

```
layer.render();
```

属性

```
id
```

描述:

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

```
type
```

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

ThreeLayer

```
new ThreeLayer (options)
```

描述:

-
-

ThreeJS 图层

-
-

示例

```
let threeLayer = new KMap.ThreeLayer({
  zooms : [16,21],
  center : new KMap.Point(114.04047074,22.53375938,200)
});
```

参数:

名称	类型	描述
options	optionsObject	
center	Point	场景中心点

optionsObject :

名称	类型	描述
id	String	图层 ID
zIndex	Number	图层层级，1-59
zooms	Array.<Number>	图层显示比例尺级别，如[10,16]在 10-16 比例

名称	类型	描述
		尺级别显示
<code>opacity</code>	<code>Number</code>	图层透明度， <code>0-1</code> ， <code>0</code> 为全透明， <code>1</code> 为不透明，默认 <code>1</code>
<code>visible</code>	<code>boolean</code>	图层是否显示，默认 <code>true</code>

函数

```
setCenter(point)
```

描述:

-
-

设置中心点坐标

-
-

参数:

名称	类型	描述
<code>point</code>	<code>Point</code>	中心点

```
add(obj)
```

描述:

-
-

添加 ThreeJS 对象

-
-

示例

```
const geometry = new KMap.THREE.SphereGeometry( 150, 32, 16 );
const material = new KMap.THREE.MeshBasicMaterial( { color: 0xffff00 ,
transparent : true} );
const mesh = new KMap.THREE.Mesh( geometry, material );
threeLayer.add(mesh);
```

参数:

名称	类型	描述
obj	Object	ThreeJS 对象

remove(obj)

描述:

-
-

移除 ThreeJS 对象

-
-

示例

```
const geometry = new KMap.THREE.SphereGeometry( 150, 32, 16 );
const material = new KMap.THREE.MeshBasicMaterial( { color: 0xffff00 ,
transparent : true} );
const mesh = new KMap.THREE.Mesh( geometry, material );
threeLayer.remove(mesh);
```

参数:

名称	类型	描述
obj	Object	ThreeJS 对象

getChildren()

描述:

-
-

获取 ThreeJS 对象集合

-
-

示例

```
threeLayer.getChildren();
```

返回:

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();
```

setzIndex(zIndex)

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setzIndex(100);
```

参数:

名称	类型	描述
zIndex	Number	层级，1-59

```
setOpacity(opacity)
```

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

```
show()
```

描述:

-
-

显示

-
-

示例

```
layer.show();
```

```
hide()
```

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

```
setZooms(zooms)
```

描述:

-
-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
zooms	Array.<Number>	显示范围, [min,max], 如[10,16]在 10-16 比例尺级别显示

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getzIndex();
```

返回:

层级

类型:
number

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

`Array.<number>`

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

`boolean`

getOpacity()

描述:

-
-

获取透明度

-

-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型:

number

update()

描述:

-
-

更新图层

-
-

示例

```
layer.update();
```

render()

描述:

-
-

渲染图层

-
-

示例

```
layer.render();
```

属性

id

描述:

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

事件

mouseover/mouseout/click/rightclick/bdclick()

描述:

-
-

数据事件

-
-

参数:

名称	类型	描述
<code>originEvent</code>	<code>Object</code>	原始事件
<code>pixel</code>	<code>Pixel</code>	像素坐标
<code>position</code>	<code>Point</code>	地图坐标
<code>target</code>	<code>targetObject</code>	目标

targetObject :

名称	类型	描述
<code>type</code>	<code>String</code>	类型
<code>feature</code>	<code>Feature</code>	<code>GeoJson.Feature</code>

TileLayer

```
new TileLayer (options)
```

描述:

-
-

谷歌瓦片标准图层，url 中带 {x}、{y}、{-y}、{z} 格式化字符串

-

•

示例

```
let tileLayer = new KMap.TileLayer({
  getTileUrl : function (x,y,z) {
    return 'https://my.maptile.com?x='+x+'&y='+y+'&z='+z;
  }
});
let tileLayer = new KMap.TileLayer({
  getTileUrl : 'https://my.maptile.com?x={x}&y={y}&z={z}'
});
let tileLayer = new KMap.TileLayer({
  getTileUrl : {
    url : 'https://{sub}.maptile.com?x={x}&y={y}&z={z}',
    subdomains : ['p1','p2','p3','p4']
  }
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
id	String	图层 ID
zIndex	Number	图层层级, 1-59
zooms	Array.<Number>	图层显示比例尺级别, 如[10,16] 在 10-16 比例尺级别显示
opacity	Number	图层透明度, 0-1, 0 为全透明, 1 为不透明, 默认 1
visible	boolean	图层是否显示, 默认 true

名称	类型	描述
<code>getTileUrl</code>	<code>String, Object, function</code>	瓦片地址
<code>coordinates</code>	<code>String</code>	瓦片的坐标系，默认 <code>gcj02</code>
<code>tileSize</code>	<code>Number</code>	瓦片的大小，默认 <code>256</code>

函数

```
setzIndex(zIndex)
```

描述:

-
-

设置 `zIndex`

-
-

示例

```
layer.setzIndex(100);
```

参数:

名称	类型	描述
<code>zIndex</code>	<code>Number</code>	层级，1-59

```
setOpacity(opacity)
```

描述:

-

-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

```
show()
```

描述:

-
-

显示

-
-

示例

```
layer.show();
```

```
hide()
```

描述:

-
-

隐藏

-

-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-
-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
----	----	----

zooms	Array.<Number>	显示范围, [min,max], 如[10,16]在 10-16 比例尺级别显示
-------	----------------	--

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getZIndex();
```

返回:

层级

类型:

number

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

`Array.<number>`

setVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.setVisible();
```

返回:

是否显示

类型:

`boolean`

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型：
number

update()

描述：

-
-

更新图层

-
-

示例

layer.update();

render()

描述：

-
-

渲染图层

-
-

示例

layer.render();

属性

id

描述：

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

WMSLayer

new WMSLayer (options)

描述:

-
-

WMS 图层

-
-

示例

```
let wmsLayer = new KMap.WMSLayer({
  source : ogcSource
});
```

参数:

名称	类型	描述
<code>options</code>	<code>optionsObject</code>	

optionsObject :

名称	类型	描述
<code>id</code>	<code>String</code>	图层 ID
<code>zIndex</code>	<code>Number</code>	图层层级, 1-59
<code>zooms</code>	<code>Array.<Number></code>	图层显示比例尺级别, 如[10,16]在 10-16 比例尺级别显示
<code>opacity</code>	<code>Number</code>	图层透明度, 0-1, 0 为全透明, 1 为不透明, 默认 1
<code>visible</code>	<code>boolean</code>	图层是否显示, 默认 true
<code>source</code>	<code>OGCSource</code>	OGC 数据源

函数

setSource(source)

描述:

-
-

设置 OGC 来源

-
-

示例

```
wmsLayer.setSource(source);
```

参数:

名称	类型	描述
source	OGCSource	OGC 数据源

setzIndex(zIndex)

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setzIndex(100);
```

参数:

名称	类型	描述
zIndex	Number	层级, 1-59

setOpacity(opacity)

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

```
show()
```

描述:

-
-

显示

-
-

示例

```
layer.show();
```

```
hide()
```

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-
-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
zooms	Array.<Number>	显示范围, [min,max], 如[10,16]在 10-16 比例尺级别显示

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getZIndex();
```

返回:

层级

类型:

number

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

Array.<number>

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

boolean

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型：
number

update()

描述：

-
-

更新图层

-
-

示例

layer.update();

render()

描述：

-
-

渲染图层

-
-

示例

layer.render();

属性

id

描述：

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

WMTSLayer

new WMTSLayer (options)

描述:

-
-

WMTS 图层

-
-

示例

```
let wmtsLayer = new KMap.WMTSLayer({
  source : ogcSource
});
```

参数:

名称	类型	描述
<code>options</code>	<code>optionsObject</code>	

optionsObject :

名称	类型	描述
<code>id</code>	<code>String</code>	图层 ID
<code>zIndex</code>	<code>Number</code>	图层层级, 1-59
<code>zooms</code>	<code>Array.<Number></code>	图层显示比例尺级别, 如[10,16]在 10-16 比例尺级别显示
<code>opacity</code>	<code>Number</code>	图层透明度, 0-1, 0 为全透明, 1 为不透明, 默认 1
<code>visible</code>	<code>boolean</code>	图层是否显示, 默认 true
<code>source</code>	<code>OGCSource</code>	OGC 数据源

函数

setSource(source)

描述:

-
-

设置 OGC 来源

-
-

示例

```
wmsLayer.setSource(source);
```

参数:

名称	类型	描述
source	OGCSource	OGC 数据源

setzIndex(zIndex)

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setzIndex(100);
```

参数:

名称	类型	描述
zIndex	Number	层级, 1-59

setOpacity(opacity)

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

```
show()
```

描述:

-
-

显示

-
-

示例

```
layer.show();
```

```
hide()
```

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-
-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
zooms	Array.<Number>	显示范围, [min,max], 如[10,16]在 10-16 比例尺级别显示

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getZIndex();
```

返回:

层级

类型:

number

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

Array.<number>

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

boolean

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型：
number

update()

描述：

-
-

更新图层

-
-

示例

layer.update();

render()

描述：

-
-

渲染图层

-
-

示例

layer.render();

属性

id

描述：

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

HeatMap

new HeatMap (map,options)

描述:

-
-

热力图

-
-

示例

```
let heatmap = new KMap.HeatMap(map);
```

参数:

名称	类型	描述
<code>map</code>	<code>Map</code>	地图对象
<code>options</code>	<code>optionsObject</code>	

optionsObject :

名称	类型	描述
<code>radius</code>	<code>number</code>	半径，单位像素
<code>gradient</code>	<code>object</code>	变化率对象，key 为间隔位置,value 为 0-1 的变化颜色
<code>opacity</code>	<code>Array.<number></code>	透明度变化率,0 为全透明，1 为不透明，默认 [0,1]

函数

setOptions(options)

描述:

-
-

设置参数

-
-

示例

```
heatmap.setOptions({
  radius:50,
  opacity:[0,0.8]
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
radius	number	半径，单位像素
gradient	object	变化率对象，key 为间隔位置,value 为 0-1 的变化颜色
opacity	Array.<number>	透明度变化率,0 为全透明，1 为不透明，默认 [0,1]

getOptions()

描述:

-
-

获取配置

-
-

示例

```
let opts = heatmap.getOptions();
```

返回:

配置

类型:

object

setDataSet(dataset)

描述:

-
-

设置数据集

-
-

示例

```
let data = [{ position:[116.191031,39.988585],count: 30},
{position:[116.389275,39.925818],count: 15}];
let opts = heatmap.setDataSet({
  data : data,
  max : 100
});
```

参数:

名称	类型	描述
dataset	datasetObject	数据集对象

datasetObject :

名称	类型	描述
max	number	最大值
data	Array.<object>	最大值

datasetData[]Object :

名称	类型	描述
----	----	----

名称	类型	描述
<code>position</code>	<code>Array.<number></code>	坐标数组, [lon,lat]
<code>count</code>	<code>number</code>	数量

`getDataSet()`

描述:

-
-

获取数据集

-
-

示例

```
let dataset = heatmap.getDataSet();
```

返回:

数据集对象

类型:

`object`

`setzIndex(zIndex)`

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setZIndex(100);
```

参数:

名称	类型	描述
zIndex	Number	层级，1-59

setOpacity(*opacity*)

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

show()

描述:

-
-

显示

-
-

示例

```
layer.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-
-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
----	----	----

zooms	<code>Array.<Number></code>	显示范围, [min,max], 如[10,16]在 10-16 比例
--------------	-----------------------------------	-------------------------------------

名称	类型	描述
		尺级别显示

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getZIndex();
```

返回:

层级

类型:

number

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

Array.<number>

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

boolean

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型:

number

update()

描述:

-
-

更新图层

-
-

示例

```
layer.update();
```

render()

描述:

-
-

渲染图层

-
-

示例

```
layer.render();
```

属性

```
id
```

描述:

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

```
type
```

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

MarkerCluster

new MarkerCluster (*map,options*)

描述:

-
-

点聚合

-
-

示例

```
let cluster = new KMap.MarkerCluster(map,{
  textFormat: 'abbr',
  pointIcon : new KMap.Icon({
    size: new KMap.Size(20, 20),
    imageOffset: new KMap.Pixel(0, 0),
    image: './images/node.png',
    imageSize: new KMap.Size(20, 20)
  }),
  clusterIcon : [
    new KMap.Icon({
      size: new KMap.Size(44, 44),
      imageOffset: new KMap.Pixel(0, 0),
      image: './images/node01.png',
      imageSize: new KMap.Size(44, 44)
    }),
    new KMap.Icon({
      size: new KMap.Size(48, 48),
      imageOffset: new KMap.Pixel(0, 0),
      image: './images/node02.png',
      imageSize: new KMap.Size(48, 48)
    }),
    new KMap.Icon({
      size: new KMap.Size(52, 52),
      imageOffset: new KMap.Pixel(0, 0),
      image: './images/node03.png',
      imageSize: new KMap.Size(52, 52)
    }),
    new KMap.Icon({
      size: new KMap.Size(56, 56),
      imageOffset: new KMap.Pixel(0, 0),
      image: './images/node04.png',
      imageSize: new KMap.Size(56, 56)
    })
  ]
});
```

```

        new KMap.Icon({
            size: new KMap.Size(60, 60),
            imageOffset: new KMap.Pixel(0, 0),
            image: './images/node05.png',
            imageSize: new KMap.Size(60, 60)
        })
    ],
    click : function(obj){
    }
});

```

参数:

名称	类型	描述
<code>map</code>	<code>Map</code>	地图对象
<code>options</code>	<code>optionsObject</code>	

optionsObject :

名称	类型	描述
<code>radius</code>	<code>number</code>	聚合半径, 单位像素, 默认 120
<code>minZoom</code>	<code>number</code>	最小比例尺, 默认 4
<code>maxZoom</code>	<code>number</code>	最大比例尺, 默认 17
<code>visible</code>	<code>boolean</code>	是否显示, 默认 true
<code>pointIcon</code>	<code>Icon</code>	数据点图标 <code>Icon</code> 对象
<code>clusterIcon</code>	<code>Array.<Icon></code>	聚合点图标 <code>Icon</code> 对象
<code>textFormat</code>	<code>string, function</code>	聚合数量显示格式, 固定格式:

名称	类型	描述
		数字[number],缩写[abbr] 函数: 返回值
click	function	点击事件, 回调函数
renderClusterMarker	function	渲染聚合图标, 回调函数
renderMarker	function	渲染数据图标, 回调函数

函数

```
setOptions(options)
```

描述:

-
-

设置参数

-
-

示例

```
cluster.setOptions({
  minZoom:5,
  maxZoom:20
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
radius	number	聚合半径，单位像素,默认 120
minZoom	number	最小比例尺，默认 4
maxZoom	number	最大比例尺，默认 17
visible	boolean	是否显示，默认 true
pointIcon	Icon	数据点图标 Icon 对象
clusterIcon	Array.<Icon>	聚合点图标 Icon 对象
textFormat	string,function	聚合数量显示格式，固定格式： 数字[number],缩写[abbr] 函数： 返回值
click	function	点击事件，回调函数
renderClusterMarker	function	渲染聚合图标，回调函数
renderMarker	function	渲染数据图标，回调函数

setRadius(radius)

描述:

-
-

设置聚合半径

-
-

示例

```
cluster.setRadius(150);
```

参数:

名称	类型	描述
radius	Number	聚合半径，单位像素

getRadius()

描述:

-
-

返回聚合半径

-
-

示例

```
let radius = cluster.getRadius();
```

返回:

聚合半径，单位像素

类型:

Number

setMaxZoom(zoom)

描述:

-
-

设置最大聚合比例尺

-
-

示例

```
cluster.setMaxZoom(20);
```

参数:

名称	类型	描述
zoom	Number	最大比例尺

```
getMaxZoom()
```

描述:

-
-

返回最大聚合比例尺

-
-

示例

```
cluster.getMaxZoom();
```

返回:

最大比例尺

类型:

number

setClusterIcon(*icons*)

描述:

-
-

设置聚合图标集[1-9, 10-99, 100-999, 1000-9999, ...]

-
-

示例

```
cluster.setClusterIcon([icon1,icon2,icon3]);
```

参数:

名称	类型	描述
icons	Array.<Icon>	图标集，数组索引按 1 位，10 位，100 位，1000 位显示

getClusterIcon()

描述:

-
-

返回聚合图标集

-
-

示例

```
cluster.getClusterIcon();
```

返回:

图标集

类型：
Array.<Icon>

setPointIcon(icon)

描述:

-
-

设置点图标

-
-

示例

```
cluster.setPointIcon(icon);
```

参数:

名称	类型	描述
icon	Icon	图标 Icon 对象

getPointIcon()

描述:

-
-

返回点图标

-
-

示例

```
cluster.getPointIcon();
```

返回:

图标 Icon 对象

类型:

Icon

setTextFormat(format)

描述:

-
-

设置文本显示样式

-
-

示例

```
cluster.setTextFormat('abbr');
cluster.setTextFormat(function(num){
    return num;
});
```

参数:

名称	类型	描述
format	string, function	显示格式，固定格式：数字[number],缩写 [abbr] 函数：返回值

setRenderClusterMarker(func)

描述:

-
-

设置聚合点显示回调

-
-

示例

```
cluster.setRenderClusterMarker(function(marker){  
  });
```

参数:

名称	类型	描述
func	function	回调函数

setRenderMarker(func)

描述:

-
-

设置点显示回调

-
-

示例

```
cluster.setRenderMarker(function(marker){  
  });
```

参数:

名称	类型	描述
func	function	回调函数

setData(features)

描述:

-
-

设置数据数组

-
-

示例

```
cluster.setData(features);
```

参数:

名称	类型	描述
features	Array.<feature>	数据数组，JSON Feature 对象数组

clear()

描述:

-
-

清除

-
-

示例

```
cluster.clear();
```

destroy()

描述:

-
-

销毁

-
-

示例

```
cluster.destroy();  
cluster = null;
```

show()

描述:

-
-

显示

-
-

示例

```
cluster.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
cluster.hide();
```

getClusters(bounds, zoom, callback)

描述:

-
-

获取范围框内的聚合点

-
-

示例

```
let bounds = new KMap.Bounds(new KMap.Point(113.71862493,22.45016411),new KMap.Point(114.42149174,22.77895389));
cluster.getClusters(bounds,11,function(features) {
    console.log(features);
});
```

参数:

名称	类型	描述
bounds	Bounds	范围框
zoom	number	比例尺级别
callback	Callback	回调

```
getChildren(clusterId,callback)
```

描述:

-
-

获取聚合点内的数据点

-
-

示例

```
cluster.getChildren(100001,function(features) {
    console.log(features);
});
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
<code>clusterId</code>	<code>number</code>	聚合点 ID
<code>callback</code>	<code>Callback</code>	回调

`getLeaves(clusterId,callback,limit,offset)`

描述:

-
-

获取聚合点下一级的子聚合集(可分页)

-
-

示例

```
//第 2 页
cluster.getLeaves(100001,function(features) {
    console.log(features);
},10,11);
```

参数:

名称	类型	描述
<code>clusterId</code>	<code>number</code>	聚合点 ID
<code>callback</code>	<code>Callback</code>	回调
<code>limit</code>	<code>number</code>	分页返回数量
<code>offset</code>	<code>number</code>	分页从第几个记录返回

```
getClusterExpansionZoom(clusterId,callback)
```

描述:

-
-

获取聚合点下一级合适的比例尺级别

-
-

示例

```
cluster.getClusterExpansionZoom(100001,function(zoom) {
    map.setCenterAndZoom(point,zoom);
});
```

参数:

名称	类型	描述
clusterId	number	聚合点 ID
callback	ZoomCallback	回调

图层/系统图层

BackgroundLayer

```
new BackgroundLayer (options)
```

描述:

-
-

系统背景层

-
-

示例

```
let backgroundLayer = new KMap.BackgroundLayer({
  zIndex : 1
});
```

参数:

名称	类型	描述
----	----	----

options	optionsObject
---------	---------------

optionsObject :

名称	类型	描述
----	----	----

id	String	图层 ID
----	--------	-------

zIndex	Number	图层层级，1-59
--------	--------	-----------

zooms	Array.<Number>	图层显示比例尺级别，如[10,16]在 10-16 比例尺级别显示
-------	----------------	-----------------------------------

opacity	Number	图层透明度，0-1，0 为全透明，1 为不透明，默认 1
---------	--------	------------------------------

visible	boolean	图层是否显示，默认 true
---------	---------	----------------

metroOnRoad	boolean	地铁线是否显示在路网上，默认 true
-------------	---------	---------------------

函数

setzIndex(zIndex)

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setzIndex(100);
```

参数:

名称	类型	描述
zIndex	Number	层级，1-59

setOpacity(opacity)

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

show()

描述:

-
-

显示

-
-

示例

```
layer.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-

-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
zooms	Array.<Number>	显示范围, [min,max], 如[10,16]在 10-16 比例尺级别显示

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getzIndex();
```

返回:

层级

类型:

number

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

Array.<number>

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

boolean

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型:

number

update()

描述:

-
-

更新图层

-
-

示例

```
layer.update();
```

render()

描述:

-
-

渲染图层

-
-

示例

```
layer.render();
```

属性

id

描述:

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

BuildingLayer

```
new BuildingLayer (options)
```

描述:

-
-

系统建筑物层

-
-

示例

```
let buildingLayer = new KMap.BuildingLayer({
  zIndex : 2
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
id	String	图层 ID

名称	类型	描述
<code>zIndex</code>	<code>Number</code>	图层层级，1-59
<code>zooms</code>	<code>Array.<Number></code>	图层显示比例尺级别，如 <code>[10,16]</code> 在 10-16 比例尺级别显示
<code>opacity</code>	<code>Number</code>	图层透明度，0-1，0 为全透明，1 为不透明，默认 1
<code>visible</code>	<code>boolean</code>	图层是否显示，默认 true

函数

```
setzIndex(zIndex)
```

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setzIndex(100);
```

参数:

名称	类型	描述
<code>zIndex</code>	<code>Number</code>	层级，1-59

setOpacity(*opacity*)

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

show()

描述:

-
-

显示

-
-

示例

```
layer.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-
-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
----	----	----

zooms	Array.<Number>	显示范围, [min,max], 如[10,16]在 10-16 比例尺级别显示
-------	----------------	--

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

```
getzIndex()
```

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getZIndex();
```

返回:

层级

类型:
number

```
getZooms()
```

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

`Array.<number>`

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

`boolean`

getOpacity()

描述:

-
-

获取透明度

-

-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型:

number

update()

描述:

-

-

更新图层

-

-

示例

```
layer.update();
```

render()

描述:

-

-

渲染图层

-

-

示例

```
layer.render();
```

属性

id

描述:

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

事件

mouseover/mouseout/click/rightclick/bdclick()

描述:

-
-

建筑物事件

-
-

参数:

名称	类型	描述
<code>originEvent</code>	<code>Object</code>	原始事件
<code>pixel</code>	<code>Pixel</code>	像素坐标
<code>position</code>	<code>Point</code>	地图坐标
<code>target</code>	<code>targetObject</code>	目标

targetObject :

名称	类型	描述
<code>type</code>	<code>String</code>	类型
<code>id</code>	<code>Number</code>	ID
<code>name</code>	<code>String</code>	名称
<code>point</code>	<code>Array.<Number></code>	点范围数组

HadLayer

```
new HadLayer (options)
```

描述:

-
-

系统高精度图层

-
-

示例

```
let hadLayer = new KMap.HadLayer({
  zIndex : 5
});
```

参数:

名称	类型	描述
----	----	----

options	optionsObject
---------	---------------

optionsObject :

名称	类型	描述
----	----	----

id	String	图层 ID
----	--------	-------

zIndex	Number	图层层级，1-59
--------	--------	-----------

zooms	Array.<Number>	图层显示比例尺级别，如[10,16]在 10-16 比例尺级别显示
-------	----------------	-----------------------------------

opacity	Number	图层透明度，0-1，0 为全透明，1 为不透明，默认 1
---------	--------	------------------------------

visible	boolean	图层是否显示，默认 true
---------	---------	----------------

函数

setzIndex(zIndex)

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setzIndex(100);
```

参数:

名称	类型	描述
zIndex	Number	层级，1-59

setOpacity(opacity)

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

show()

描述:

-
-

显示

-
-

示例

```
layer.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-

-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
zooms	Array.<Number>	显示范围, [min,max], 如[10,16]在 10-16 比例尺级别显示

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getzIndex();
```

返回:

层级

类型:

number

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

Array.<number>

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

boolean

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型:

number

update()

描述:

-
-

更新图层

-
-

示例

```
layer.update();
```

render()

描述:

-
-

渲染图层

-
-

示例

```
layer.render();
```

属性

id

描述:

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

IndoorLayer

```
new IndoorLayer (options)
```

描述:

-
-

系统室内图层

-
-

示例

```
let indoorLayer = new KMap.IndoorLayer({
  floorBarVisible : true,
  zIndex : 8,
  visible: true
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
----	----	----

名称	类型	描述
id	String	图层 ID
zIndex	Number	图层层级，1-59
zooms	Array.<Number>	图层显示比例尺级别，如[10,16]在10-16 比例尺级别显示
opacity	Number	图层透明度，0-1，0 为全透明，1 为不透明，默认 1
visible	boolean	图层是否显示，默认 true
position	Object	楼层控件位置， bottom,right,left,top 等位置设置
floorBarVisible	boolean	楼层控件是否显示，默认 true
level	Number	楼层控件显示层数，楼层最大显示层数，默认 11 层

函数

setFloorBarVisible(visible)

描述:

-
-

设置楼层控件是否显示

-
-

参数:

名称	类型	描述
visible	Boolean	

getFloorBarVisible()

描述:

-
-

获取楼层控件是否显示

-
-

返回:

类型:
*,boolean

getElement()

描述:

-
-

获取 element

-
-

返回:

类型：
HTMLElement

show()

描述:

-
-

显示层

-
-

hide()

描述:

-
-

隐藏层

-
-

setFloor(index)

描述:

-
-

设置楼层

-
-

参数:

名称	类型	描述
index		

```
trigger(map)
```

描述:

-
-

地图改变触发室内图

-
-

参数:

名称	类型	描述
map	Map	

```
destroy()
```

描述:

-
-

销毁

-
-

```
setzIndex(zIndex)
```

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setZIndex(100);
```

参数:

名称	类型	描述
zIndex	Number	层级，1-59

setOpacity(*opacity*)

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

setZooms(*zooms*)

描述:

-
-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
----	----	----

<code>zooms</code>	<code>Array.<Number></code>	显示范围, [min,max], 如[10,16]在 10-16 比例尺级别显示
--------------------	-----------------------------------	--

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getzIndex();
```

返回:

层级

类型:

`number`

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

`Array.<number>`

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

boolean

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型:

number

update()

描述:

-
-

更新图层

-
-

示例

```
layer.update();
```

render()

描述:

-
-

渲染图层

-
-

示例

```
layer.render();
```

属性

id

描述:

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

事件

```
mouseover/mouseout/click/rightclick/bdclick()
```

描述:

- -
- 室内图标注事件

-
-

参数:

名称	类型	描述
originEvent	Object	原始事件
pixel	Pixel	像素坐标
position	Point	地图坐标
target	targetObject	目标

targetObject :

名称	类型	描述
type	String	类型
id	Number	ID

名称	类型	描述
<code>name</code>	<code>String</code>	名称
<code>point</code>	<code>Array.<Number></code>	坐标点
<code>typeCode</code>	<code>Number</code>	POI 类型
<code>floor</code>	<code>Number</code>	所在楼层(楼上从 <code>0</code> 开始, 楼下从 <code>-1</code> 开始)

LabelLayer

```
new LabelLayer (options)
```

描述:

-
-

系统标注图层

-
-

示例

```
let labelLayer = new KMap.LabelLayer();
```

参数:

名称	类型	描述
<code>options</code>	<code>optionsObject</code>	

optionsObject :

名称	类型	描述
----	----	----

名称	类型	描述
<code>id</code>	<code>String</code>	图层 ID
<code>zIndex</code>	<code>Number</code>	图层层级，1-59
<code>zooms</code>	<code>Array.<Number></code>	图层显示比例尺级别，如 <code>[10,16]</code> 在 10-16 比例尺级别显示
<code>opacity</code>	<code>Number</code>	图层透明度，0-1，0 为全透明，1 为不透明，默认 1
<code>visible</code>	<code>boolean</code>	图层是否显示，默认 <code>true</code>

函数

`setZIndex(zIndex)`

描述:

-
-

设置 `zIndex`

-
-

示例

```
layer.setZIndex(100);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
zIndex	Number	层级，1-59

setOpacity(*opacity*)

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

show()

描述:

-
-

显示

-
-

示例

```
layer.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-
-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
----	----	----

zooms	Array.<Number>	显示范围, [min,max], 如[10,16]在 10-16 比例尺级别显示
-------	----------------	--

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getZIndex();
```

返回:

层级

类型:

number

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

`Array.<number>`

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

`boolean`

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型:

number

update()

描述:

-
-

更新图层

-
-

示例

```
layer.update();
```

render()

描述:

-
-

渲染图层

-
-

示例

```
layer.render();
```

属性

id

描述:

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

事件

mouseover/mouseout/click/rightclick/bdclick()

描述:

-
-

标注事件

-
-

参数:

名称	类型	描述
<code>originEvent</code>	<code>Object</code>	原始事件
<code>pixel</code>	<code>Pixel</code>	像素坐标
<code>position</code>	<code>Point</code>	地图坐标
<code>target</code>	<code>targetObject</code>	目标

targetObject :

名称	类型	描述
<code>type</code>	<code>String</code>	类型
<code>id</code>	<code>Number</code>	ID
<code>name</code>	<code>String</code>	名称
<code>point</code>	<code>Array.<Number></code>	坐标点
<code>typeCode</code>	<code>Number</code>	POI 类型

LandmarkLayer

new LandmarkLayer (options)

描述:

-
-

系统个性建筑图层

-
-

示例

```
let landmarkLayer = new KMap.LandmarkLayer({
  zIndex : 3,
  visible: true
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
id	String	图层 ID
zIndex	Number	图层层级，1-59
zooms	Array.<Number>	图层显示比例尺级别，如[10,16]在 10-16 比例尺级别显示
opacity	Number	图层透明度，0-1，0 为全透明，1 为不透明，默认 1

名称	类型	描述
<code>visible</code>	<code>boolean</code>	图层是否显示，默认 <code>true</code>

函数

```
setzIndex(zIndex)
```

描述:

-
-

设置 `zIndex`

-
-

示例

```
layer.setzIndex(100);
```

参数:

名称	类型	描述
<code>zIndex</code>	<code>Number</code>	层级，1-59

```
setOpacity(opacity)
```

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

show()

描述:

-
-

显示

-
-

示例

```
layer.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-
-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
zooms	Array.<Number>	显示范围，[min,max]，如[10,16]在 10-16 比例尺级别显示

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getzIndex();
```

返回:

层级

类型:

`number`

`getZooms()`

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

`Array.<number>`

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

boolean

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型:

number

update()

描述:

-
-

更新图层

-
-

示例

```
layer.update();
```

render()

描述:

-
-

渲染图层

-
-

示例

```
layer.render();
```

属性

id

描述:

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

事件

mouseover/mouseout/click/rightclick/bdclick()

描述:

-
-

个性建筑事件

-
-

参数:

名称	类型	描述
originEvent	Object	原始事件

名称	类型	描述
<code>pixel</code>	<code>Pixel</code>	像素坐标
<code>position</code>	<code>Point</code>	地图坐标
<code>target</code>	<code>targetObject</code>	目标

targetObject :

名称	类型	描述
<code>type</code>	<code>String</code>	类型
<code>id</code>	<code>Number</code>	ID
<code>name</code>	<code>String</code>	名称
<code>point</code>	<code>Array.<Number></code>	点坐标

RoadNetLayer

```
new RoadNetLayer (options)
```

描述:

-
-

系统路网图层

-
-

示例

```
let roadNetLayer = new KMap.RoadNetLayer({
  zIndex : 6
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
id	String	图层 ID
zIndex	Number	图层层级，1-59
zooms	Array.<Number>	图层显示比例尺级别，如[10,16]在 10-16 比例尺级别显示
opacity	Number	图层透明度，0-1，0 为全透明，1 为不透明，默认 1
visible	boolean	图层是否显示，默认 true

函数

```
setZIndex(zIndex)
```

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setZIndex(100);
```

参数:

名称	类型	描述
zIndex	Number	层级，1-59

setOpacity(*opacity*)

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

show()

描述:

-
-

显示

-
-

示例

```
layer.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-
-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
zooms	Array.<Number>	显示范围，[min,max]，如[10,16]在 10-16 比例尺级别显示

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getZIndex();
```

返回:

层级

类型:

number

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

Array.<number>

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

boolean

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型:

number

update()

描述:

-
-

更新图层

-
-

示例

```
layer.update();
```

render()

描述:

-
-

渲染图层

-
-

示例

```
layer.render();
```

属性

```
id
```

描述:

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

```
type
```

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

SatelliteLayer

```
new SatelliteLayer (options)
```

描述:

-
-

系统卫星图层

-
-

示例

```
let satellitelayer = new KMap.SatelliteLayer({
  zIndex : 4,
  visible: false
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
id	String	图层 ID
zIndex	Number	图层层级，1-59
zooms	Array.<Number>	图层显示比例尺级别，如[10,16]在 10-16 比例尺级别显示

名称	类型	描述
opacity	Number	图层透明度，0-1，0 为全透明，1 为不透明，默认 1
visible	boolean	图层是否显示，默认 true

函数

```
setzIndex(zIndex)
```

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setzIndex(100);
```

参数:

名称	类型	描述
zIndex	Number	层级，1-59

```
setOpacity(opacity)
```

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

```
show()
```

描述:

-
-

显示

-
-

示例

```
layer.show();
```

```
hide()
```

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

setZooms(zooms)

描述:

-
-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
zooms	Array.<Number>	显示范围, [min,max], 如[10,16]在 10-16 比例尺级别显示

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getZIndex();
```

返回:

层级

类型:

number

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

Array.<number>

getVisible()

描述:

-
-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

boolean

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型：
number

update()

描述：

-
-

更新图层

-
-

示例

layer.update();

render()

描述：

-
-

渲染图层

-
-

示例

layer.render();

属性

id

描述：

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

TrafficLayer

new TrafficLayer (options)

描述:

-
-

系统路况图层

-
-

示例

```
let trafficLayer = new KMap.TrafficLayer({
  autoRefresh: true,
  visible: false,
  zIndex: 7
});
```

```
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
id	String	图层 ID
zIndex	Number	图层层级, 1-59
zooms	Array.<Number>	图层显示比例尺级别, 如[10,16]在 10-16 比例尺级别显示
opacity	Number	图层透明度, 0-1, 0 为全透明, 1 为不透明, 默认 1
visible	boolean	图层是否显示, 默认 true
autoRefresh	boolean	是否自动刷新, 默认 true
interval	Number	刷新间隔时间, 单位秒, 默认 60 秒

函数

```
startFresh()
```

描述:

-
-

开启自动刷新

-
-

示例

```
trafficLayer.startFresh();
```

stopFresh()

描述:

-
-

停止自动刷新

-
-

示例

```
trafficLayer.stopFresh();
```

setInterval(value)

描述:

-
-

设置刷新闻隔时间

-
-

示例

```
trafficLayer.setInterval(30);
```

参数:

名称	类型	描述
value	Number	间隔时间，单位秒

setzIndex(zIndex)

描述:

-
-

设置 zIndex

-
-

示例

```
layer.setzIndex(100);
```

参数:

名称	类型	描述
zIndex	Number	层级，1-59

setOpacity(opacity)

描述:

-
-

设置透明度

-
-

示例

```
layer.setOpacity(0.8);
```

参数:

名称	类型	描述
opacity	Number	透明度，0-1，0 为全透明，1 为不透明

```
show()
```

描述:

-
-

显示

-
-

示例

```
layer.show();
```

```
hide()
```

描述:

-
-

隐藏

-
-

示例

```
layer.hide();
```

```
setZooms(zooms)
```

描述:

-
-

设置显示范围

-
-

示例

```
layer.setZooms([4,21]);
```

参数:

名称	类型	描述
----	----	----

<code>zooms</code>	<code>Array.<Number></code>	显示范围, [min,max], 如[10,16]在 10-16 比例尺级别显示
--------------------	-----------------------------------	--

destroy()

描述:

-
-

销毁图层

-
-

示例

```
layer.destroy();  
layer = null;
```

getzIndex()

描述:

-
-

获取层级

-
-

示例

```
let index= layer.getzIndex();
```

返回:

层级

类型:

`number`

getZooms()

描述:

-
-

获取显示范围

-
-

示例

```
let zooms = layer.getZooms();
```

返回:

显示范围,[min,max]

类型:

`Array.<number>`

getVisible()

描述:

-

-

获取是否显示

-
-

示例

```
let visible = layer.getVisible();
```

返回:

是否显示

类型:

boolean

getOpacity()

描述:

-
-

获取透明度

-
-

示例

```
let opacity = layer.getOpacity();
```

返回:

透明度

类型:

number

update()

描述:

-
-

更新图层

-
-

示例

```
layer.update();
```

render()

描述:

-
-

渲染图层

-
-

示例

```
layer.render();
```

属性

id

描述:

-
-

设置图层 ID

-
-

示例

```
layer.id = 'layerId';
```

type

描述:

-
-

设置类型

-
-

示例

```
layer.type = 'type';
```

服务

Driving

new Driving (options)

描述:

-
-

驾车规划路径服务

-
-

示例

```
let driving = new KMap.Driving({
  map:map,
  panel:'panel',
  licensePlate: '粤 B88888'
});
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
<code>options</code>	<code>optionsObject</code>	

optionsObject :

名称	类型	描述
<code>map</code>	<code>Map</code>	地图
<code>policy</code>	<code>DrivingPolicy</code>	算路条件， <code>DrivingPolicy</code> 枚举
<code>panel</code>	<code>string</code>	结果显示面板 ID
<code>showTraffic</code>	<code>boolean</code>	是否显示路况，默认 <code>true</code>
<code>autoFitView</code>	<code>boolean</code>	是否自动调整视图，默认 <code>true</code>
<code>licensePlate</code>	<code>string</code>	车牌号

函数

```
setPolicy(policy)
```

描述:

-
-

设置算路条件

-
-

示例

```
driving.setPolicy(KMap.DrivingPolicy.DEFAULT);
```

参数:

名称	类型	描述
policy	DrivingPolicy	条件参数，DrivingPolicy 枚举

getPolicy()

描述:

-
-

获取算路条件

-
-

示例

```
let policy = driving.getPolicy();
```

返回:

算路条件，DrivingPolicy 枚举

类型:

DrivingPolicy

setMulti(isMulti)

描述:

-
-

设置是否计算多条

-

-

示例

```
driving.setMulti(true);
```

参数:

名称	类型	描述
isMulti	boolean	是/否

setFence(fences)

描述:

-
-

设置避让围栏

-
-

示例

```
let fence = [new KMap.Point(114.0565284,22.54825997),new KMap.Point(114.05619479,22.54001982),new KMap.Point(114.07271424,22.54825997),new KMap.Point(114.05603374,22.54841005)];
driving.setFence([fence]);
```

参数:

名称	类型	描述
fences	Array.<Array.<Point>>	多围栏数组

clearFence()

描述:

-

-

清除避让围栏

-

-

示例

```
driving.clearFence([fence]);
```

setAvoidPoint(*avoids*)

描述:

-

-

设置避让点数组

-

-

示例

```
driving.setAvoidPoint([{lon:114.03571805,lat:22.53834028}]);
```

参数:

名称	类型	描述
<code>avoids</code>	<code>Array.<object></code>	避让点数组

avoids[]Object :

名称	类型	描述
<code>lon</code>	<code>number</code>	经度
<code>lat</code>	<code>number</code>	纬度

名称	类型	描述
roadId	number	道路 UID，没有传 0 或不传

```
clearAvoidPoint()
```

描述:

-
-

清除避让点

-
-

示例

```
driving.clearAvoidPoint();
```

```
setLicensePlate(licensePlate)
```

描述:

-
-

设置车牌号

-
-

示例

```
driving.setLicensePlate('粤 BNB001');
```

参数:

名称	类型	描述
licensePlate	string	车牌号

getLicensePlate()

描述:

-
-

获取车牌号

-
-

示例

```
let plate = driving.getLicensePlate();
```

返回:

车牌号

类型:

`string`

clear()

描述:

-
-

清除结果

-
-

示例

```
driving.clear();
```

destroy()

描述:

-

-

销毁对象

-

-

示例

```
driving.destroy();
```

```
search(origin,destination,options,callback)
```

描述:

-

-

计算路径

-

-

示例

```
let start = new KMap.Point(114.02306542,22.53496495);
let end = new KMap.Point(114.11779759,22.53132889);
driving.search(start,end,function(status,result) {
    if(status == 'complete') {
        //结果
        console.log(result);
    }
    else {
        //错误
        alert(result);
    }
});
```

参数:

名称	类型	描述
origin	Point, Array.<Object>	起点或点对象数组，可以传点或者名称对象数组

名称	类型	描述
<code>destination</code>	<code>Point</code>	终点
<code>options</code>	<code>optionsObject</code>	算路参数，经由点信息
<code>callback</code>	<code>Callback</code>	结果回调

origin[]Object :

名称	类型	描述
<code>point</code>	<code>Point</code>	点对象
<code>name</code>	<code>string</code>	点名称

optionsObject :

名称	类型	描述
<code>via</code>	<code>Array.<Point></code>	经由点数组

事件

```
error()
```

描述:

-
-

计算路径错误事件

-
-

参数:

名称	类型	描述
message	string	错误描述

complete()

描述:

-
-
- 计算路径完成事件
-
-

参数:

名称	类型	描述
result	DrivingResult	路径结果对象

resultObject :

名称	类型	描述
origin	object	起点信息
destination	object	终点信息
vias	Array.<object>	经由点信息数组

名称	类型	描述
<code>routes</code>	<code>Array.<object></code>	路径包数组

`error()`

描述:

-
-

计算路径错误事件

-
-

参数:

名称	类型	描述
<code>message</code>	<code>string</code>	错误描述

Correction

`new Correction (options)`

描述:

-
-

纠偏服务

-
-

示例

```
let corr = new KMap.Correction({map:map});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
map	Map	地图对象

函数

```
clear()
```

描述:

-
-

清除

-
-

示例

```
corr.clear();
```

```
pointGraspRoad(point,direction,callback)
```

描述:

-
-

点纠偏

-

-

示例

```
let p = new KMap.Point(114.026067,22.536282);
let dir = 90;
correction.pointGraspRoad(p,dir,function (status,result) {
    if(status == 'complete') {
        //显示
        console.log(result);
    }
    else {
        //错误
        alert(result);
    }
});
```

参数:

名称	类型	描述
point	Point	点参数
direction	number	角度,正北顺时针方向,单位度,无角度为-1
callback	PointCallback	回调结果,参数 status,result

trackGraspRoad(tracks,callback)

描述:

-
-

轨迹抓取道路

-
-

示例

```
let track = [{"lon":113.87038,"lat":23.07192306,"speed":20,"direction":202,"time":1
```

```
557314269},{ "lon":113.87026,"lat":23.07168639,"speed":11,"direction":183,"time":1557314275}]]];
correction.trackGraspRoad(track,function (status,result) {
    if(status == 'complete') {
        //显示
        console.log(result);
    }
    else {
        //错误
        alert(result);
    }
});
```

参数:

名称	类型	描述
tracks	<code>Array.<tracksObject></code>	轨迹数组
callback	<code>TrackCallback</code>	回调结果

tracksObject :

名称	类型	描述
lon	<code>number</code>	经度
lat	<code>number</code>	纬度
speed	<code>number</code>	速度，单位 KM/H
direction	<code>number</code>	方向，正北顺时针，单位度
time	<code>number</code>	时间戳，单位秒

Geocoder

new Geocoder (props)

描述:

-
-

地址/逆地址解析

-
-

示例

```
let geo = new KMap.Geocoder();
let geo = new KMap.Geocoder({lang: 'zh-cn'});
```

参数:

名称	类型	描述
props	propsObject	

propsObject :

名称	类型	描述
lang	String	语言, 支持 zh-cn, en

函数

getLocation(options, callback)

描述:

-
-

地理编码，将地址信息转化为经纬度坐标信息

-
-

示例

```
geo.getLocation({
  address: '广东省深圳市福田区车公庙深南大道 6023 号创建大厦 28 楼'
},(status,result) => {
  console.log('getLocation',status,result)
})
```

参数:

名称	类型	描述
options	optionsObject	
callback	function	回调方法，参数为 status,result

optionsObject :

名称	类型	描述
address	String	解析完整地址串，如：广东省深圳市福田区街道路*号大厦
city	String	所在城市，城市名或编码

```
getAddress(options,callback)
```

描述:

-
-

将经纬度坐标信息转化为的地址信息

-
-

示例

```
geo.getAddress({
  location: new KMap.Point(114.023149,22.534956)
},(status,result) => {
```

```
console.log('getAddress',status,result)
})
```

参数:

名称	类型	描述
options	optionsObject	
callback	function	回调方法，参数为 status,result

optionsObject :

名称	类型	描述
location	Point	经纬度坐标
range	String	检索范围，单位米，默认 1000 米
extension	String	检索类型，比如：poi,road,aoi，多个用半角逗号分隔

GeoCoderResult

```
new GeoCoderResult (json)
```

描述:

-
-

地址解析结果

-
-

参数:

名称	类型	描述
json	Object	

函数

```
destroy()
```

描述:

-
-

销毁结果

-
-

示例

```
geoResult.destroy();  
geoResult = null;
```

属性

```
list
```

描述:

-
-

获取结果列表

-
-

示例

```
geoResult.list;
```

ReGeoCoderResult

new ReGeoCoderResult (json)

描述:

-
-

逆地址结果

-
-

参数:

名称	类型	描述
json	Object	

函数

destroy()

描述:

-
-

销毁

-
-

示例

```
rgeoResult.destroy();
```

属性

list

描述:

-
-

获取列表

-
-

示例

```
rgeoResult.list;
```

GeoLocation

```
new GeoLocation ()
```

描述:

-
-

浏览器定位

-
-

函数

```
getPosition(callback,opts)
```

描述:

-
-

获取浏览器定位位置

-
-

示例

```
KMap.GeoLocation.getPosition((status,result)=>{
```

```
console.log(status,result)
},{enableHighAccuracy:false})
```

参数:

名称	类型	描述
callback		回调方法，参数为 status,result
opts	optsObject	

optsObject :

名称	类型	描述
enableHighAccuracy	Boolean	是否高精度
timeout	Number	超时时间，单位毫秒
maximumAge	Number	位置缓存时间,单位毫秒

```
watchPosition(callback,opts)
```

描述:

-
-

监控浏览器定位位置

-
-

示例

```
KMap.GeoLocation.watchPosition((status,result)=>{
  console.log(status,result)
},{enableHighAccuracy:false})
```

参数:

名称	类型	描述
callback		回调方法，参数为 status,result
opts	optsObject	

optsObject :

名称	类型	描述
enableHighAccuracy	Boolean	是否高精度
timeout	Number	超时时间，单位毫秒
maximumAge	Number	位置缓存时间,单位毫秒

```
clearWatch()
```

描述:

-
-

结束监控

-
-

示例

```
KMap.GeoLocation.clearWatch();
```

IpLocation

```
new IpLocation ()
```

描述:

-
-

IP 定位

-
-

函数

`getRegion(callback)`

描述:

-
-

获取定位

-
-

示例

```
IpLocation.getRegion((status,result)=>{console.log(status,result)})
```

参数:

名称	类型	描述
<code>callback</code>	<code>function</code>	回调方法，参数为 <code>status,result</code>

RegionSearch

函数

`findRegionByName(name,range)`

描述:

-
-

依据名称查找行政区信息：匹配规则：优先全匹配，在进行同长度字符匹配；
匹配顺序：城市，省份，区县；返回获取第一个符合的记录

-
-

参数:

名称	类型	描述
name	String	查询字符串
range	Array.<String>	限制范围，['province','city','county']

返回:

行政区信息，{"level": 级别,"pgb": 上级国标编码,"gb": 国标编码,"kld": 凯立德编码,"cn": 中文名,"en": 英文名,"xy": 坐标}

类型:
undefined, Object

```
getRegionByCode(code,range)
```

描述:

-
-

依据编码，获取行政区信息

-
-

参数:

名称	类型	描述
<code>code</code>	<code>Number</code>	行政区编码
<code>range</code>	<code>Array.<String></code>	匹配范围, <code>['gb','kld']</code>

返回:

行政区信息, `{"level": 级别,"pgb": 上级国标编码,"gb": 国标编码,"kld": 凯立德编码,"cn": 中文名,"en": 英文名,"xy": 坐标}`

类型:
`undefined, Object`

DrivingResult

属性

origin

描述:

-
-

起点

-
-

destination

描述:

-
-

终点

-
-

vias

描述:

-
-

经由点

-
-

routes

描述:

-
-

路径结果

-
-

RouteStep

属性

point

描述:

-
-

起点

-
-

distance

描述:

-
-

距离，单位米

-
-

description

描述:

-
-

导航描述

-
-

action

描述:

-
-

走完后转向动作

-
-

road

描述:

-
-

道路名

-
-

roadType

描述:

-
-

道路类型

-
-

path

描述:

-
-

形状点

-
-

TruckDriving

new TruckDriving (options)

描述:

-
-

货车规划路径服务

-

•

示例

```
let driving = new KMap.TruckDriving({
  map:map,
  panel:'panel',
  licensePlate: '粤 B88888',
  width: 2.5,
  height: 2,
  load: 10,
  axles: 4
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
map	Map	地图
policy	DrivingPolicy	算路条件
panel	string	结果显示面板 ID
showTraffic	boolean	是否显示路况，默认 true
autoFitView	boolean	是否自动调整视图，默认 true
width	Number	车宽，单位米，默认 2.5
height	Number	车高，单位米，默认 1.6

名称	类型	描述
load	Number	车载重，单位吨，默认 2.5
selfWeight	Number	车自重，单位吨，默认 10
axles	Number	车轴数，默认 2
type	VehicleType	车型，默认轻型货车
licensePlate	string	车牌号

函数

```
setPolicy(policy)
```

描述:

-
-

设置算路条件, DrivingPolicy

-
-

示例

```
driving.setPolicy(KMap.DrivingPolicy.DEFAULT);
```

参数:

名称	类型	描述
policy	DrivingPolicy	条件参数

getPolicy()

描述:

-
-

获取算路条件

-
-

示例

```
let policy = driving.getPolicy();
```

返回:

算路条件

类型:
DrivingPolicy

setMulti(isMulti)

描述:

-
-

设置是否计算多条

-
-

示例

```
driving.setMulti(true);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
isMulti	boolean	是/否

setFence(fences)

描述:

-
-

设置避让围栏

-
-

示例

```
let fence = [new KMap.Point(114.0565284,22.54825997),new KMap.Point(114.05619479,22.54001982),new KMap.Point(114.07271424,22.54825997),new KMap.Point(114.05603374,22.54841005)];
driving.setFence([fence]);
```

参数:

名称	类型	描述
fences	Array.<Array.<Point>>	围栏数组

clearFence()

描述:

-
-

清除避让围栏

-

-

示例

```
driving.clearFence([fence]);
```

setAvoidPoint(avoid)

描述:

-

-

设置避让点数组

-

-

示例

```
driving.setAvoidPoint([{lon:114.03571805,lat:22.53834028}]);
```

参数:

名称	类型	描述
----	----	----

avoid	Array.<object>	避让点数组
-------	----------------	-------

avoid[]Object :

名称	类型	描述
----	----	----

lon	number	经度
-----	--------	----

lat	number	纬度
-----	--------	----

roadId	number	道路 UID, 无传 0 或不设置
--------	--------	-------------------

clearAvoidPoint()

描述:

-
-

清除避让点

-
-

示例

```
driving.clearAvoidPoint();
```

setVehicle(vehicle)

描述:

-
-

设置车参数

-
-

示例

```
driving.setVehicle({
  licensePlate: '粤 B88888',
  width: 2.5,
  height: 2,
  load: 10,
  axles: 4
});
```

参数:

名称	类型	描述
vehicle	vehicleObject	车参数

vehicleObject :

名称	类型	描述
----	----	----

名称	类型	描述
<code>width</code>	<code>Number</code>	车宽，单位米
<code>height</code>	<code>Number</code>	车高，单位米
<code>load</code>	<code>Number</code>	车载重，单位吨
<code>selfWeight</code>	<code>Number</code>	车自重，单位吨
<code>axles</code>	<code>Number</code>	车轴数
<code>type</code>	<code>VehicleType</code>	车型， <code>VehicleType</code>
<code>licensePlate</code>	<code>string</code>	车牌号

`clear()`

描述:

-
-

清除结果

-
-

示例

```
driving.clear();
```

`destroy()`

描述:

-
-

销毁对象

-
-

示例

```
driving.destroy();
```

```
search(origin,destination,options,callback)
```

描述:

-
-

计算路径

-
-

示例

```
let start = new KMap.Point(114.02306542,22.53496495);
let end = new KMap.Point(114.11779759,22.53132889);
driving.search(start,end,function(status,result) {
    if(status == 'complete') {
        //结果
        console.log(result);
    }
    else {
        //错误
        alert(result);
    }
});
```

参数:

名称	类型	描述
origin	Point, Array.<Object>	起点或点对象数组

名称	类型	描述
<code>destination</code>	<code>Point</code>	终点
<code>options</code>	<code>optionsObject</code>	算路参数
<code>callback</code>	<code>Callback</code>	结果回调

`origin[]Object :`

名称	类型	描述
<code>point</code>	<code>Point</code>	点对象
<code>name</code>	<code>string</code>	点名称

`optionsObject :`

名称	类型	描述
<code>via</code>	<code>Array.<Point></code>	经由点数组

事件

```
complete()
```

描述:

-
-

计算路径完成事件

-
-

参数:

名称	类型	描述
result	TruckDrivingResult	路径结果

resultObject :

名称	类型	描述
origin	object	起点
destination	object	终点
vias	Array.<object>	经由点
routes	Array.<object>	路径包

```
error()
```

描述:

-
-
- 计算路径错误事件
-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
message	string	错误描述

TruckDrivingResult

属性

origin

描述:

-
-

起点

-
-

destination

描述:

-
-

终点

-
-

vias

描述:

-
-

经由点

-
-

routes

描述:

-
-

路径结果

-
-

Walking

new Walking (options)

描述:

-
-

步行规划路径服务

-
-

示例

```
let walking = new KMap.Walking({
  map:map,
  panel:'panel'
});
```

参数:

名称	类型	描述
----	----	----

options	optionsObject	
---------	---------------	--

optionsObject :

名称	类型	描述
<code>map</code>	<code>Map</code>	地图
<code>panel</code>	<code>string</code>	结果显示面板 ID

函数

```
clear()
```

描述:

-
-

清除结果

-
-

示例

```
walking.clear();
```

```
destroy()
```

描述:

-
-

销毁对象

-
-

示例

```
walking.destroy();
```

`search(origin,destination,options,callback)`

描述:

-
-

计算路径

-
-

示例

```
let start = new KMap.Point(114.02306542,22.53496495);
let end = new KMap.Point(114.11779759,22.53132889);
walking.search(start,end,function(status,result) {
    if(status == 'complete') {
        //结果
        console.log(result);
    }
    else {
        //错误
        alert(result);
    }
});
```

参数:

名称	类型	描述
origin	Point, Array.<Object>	起点或点对象数组
destination	Point	终点
options	optionsObject	算路参数
callback	Callback	结果回调

origin[]Object :

名称	类型	描述
<code>point</code>	<code>Point</code>	点对象
<code>name</code>	<code>string</code>	点名称

optionsObject :

名称	类型	描述
<code>via</code>	<code>Array.<Point></code>	经由点数组

事件

`complete()`

描述:

-
-

步行路径完成事件

-
-

参数:

名称	类型	描述
<code>result</code>	<code>WalkingResult</code>	路径结果

resultObject :

名称	类型	描述
----	----	----

名称	类型	描述
<code>origin</code>	<code>object</code>	起点
<code>destination</code>	<code>object</code>	终点
<code>vias</code>	<code>Array.<object></code>	经由点
<code>routes</code>	<code>Array.<object></code>	路径包

`error()`

描述:

-
-

步行路径错误事件

-
-

参数:

名称	类型	描述
<code>message</code>	<code>string</code>	错误描述

WalkingResult

属性

`origin`

描述:

-
-

起点

-
-

destination

描述:

-
-

终点

-
-

vias

描述:

-
-

经由点

-
-

routes

描述:

-
-

路径结果

-
-

AutoComplete

new AutoComplete (props)

描述:

-
-

输入提示

-
-

示例

```
let ac = new AutoComplete();
let ac = new KMap.AutoComplete({
  input: "searchResultPanel-input-text",
  adcode: '440300',
  selected: (e) => {
    map.setCenterAndZoom(e.location, 20, 1000)
  }
});
```

参数:

名称	类型	描述
props	propsObject	

propsObject :

名称	类型	描述
lang	String	语言设置, 可选 zh-cn,en
input	String, HTMLInputElement	输入框 ID

名称	类型	描述
<code>adcode</code>	<code>number</code>	行政区编码
<code>selected</code>	<code>function</code>	当选中项时回调方法

函数

```
search(params,callback)
```

描述:

- -
- 检索提示结果，可由用户实现输入提示

-
-

示例

```
ac.search({
  keyword: '莲花山',
  adcode: '440300',
},(status,result)=>{console.log(status,result)})
```

参数:

名称	类型	描述
<code>params</code>	<code>paramsObject</code>	
<code>callback</code>	<code>function</code>	回调方法，参数 <code>status,result</code>

paramsObject :

名称	类型	描述
----	----	----

名称	类型	描述
keyword	keyword	关键字
adcode	number	城市编码

返回:

类型: `Object`

`destroy()`

描述:

-
-

销毁

-
-

示例

```
ac.destroy();  
ac = null;
```

AutoCompleteResult

`new AutoCompleteResult (json,params)`

描述:

-
-

输入提示结果

-

-

参数:

名称	类型	描述
json	object	
params	object	

函数

```
getRequestParams()
```

描述:

-
-

获取请求参数

-
-

示例

```
acRes.getRequestParams();
```

返回:

类型:
Object

```
getCount()
```

描述:

-
-

返回 POI 记录数

-

-

示例

```
acRes = getCount();
```

返回:

类型:

number

getPoi(index)

描述:

-
-

依据下标获取 poi

-
-

示例

```
acRes = getPoi(0);
```

参数:

名称	类型	描述
index	number	下标

返回:

类型:

undefined,Object

getPois()

描述:

-
-

获取 poi 列表

-
-

示例

```
acRes = getPois();
```

返回:

类型:
Array.<Object>

```
destroy()
```

描述:

-
-

销毁

-
-

示例

```
acRes.destroy();  
acRes = null;
```

DistrictSearch

```
new DistrictSearch (props)
```

描述:

-
-

行政区检索

-
-

示例

```
let ds = new DistrictSearch();
```

参数:

名称	类型	描述
props	propsObject	

propsObject :

名称	类型	描述
lang	String	语言设置, 可选 zh-cn,en

函数

```
search(params,callback)
```

描述:

-
-

检索

-
-

示例

```
ds.search({
  keyword:"440000",
  subdistrict:2,
  return_polygon:1
},(status,result) => {
  console.log(status,result)
})
```

参数:

名称	类型	描述
params	paramsObject	

名称	类型	描述
callback	function	回调方法，参数 status,result

paramsObject :

名称	类型	描述
keyword	String	关键字
subdistrict	Number	子行政区(0: 不返回下级行政区，1: 返回下一级行政区，2: 返回下两级行政区，3: 返回下三级行政区)，默认 0
return_polygon	Number	是否需要返回行政区边界多边形(1: 只返回当前查询 district 的边界，不返回子节点的边界，0: 不返回多边形坐标)，默认 0

返回:

类型:
Object

destroy()

描述:

-
-

销毁

-
-

示例

```
ds.destroy();
ds = null;
```

DistrictSearchResult

```
new DistrictSearchResult (json,params)
```

描述:

-
-

行政区检索结果

-
-

参数:

名称	类型	描述
json	object	
params	object	

函数

```
getRequestParams()
```

描述:

-
-

获取检索参数

-
-

示例

```
dsResult.getRequestParams();
```

返回:

类型:

Object

getCount()

描述:

-
-

获取返回数据条数

-
-

示例

```
dsResult.getCount();
```

返回:

类型:

number

getResults()

描述:

-
-

获取结果列表

-
-

示例

```
dsResult.getResults();
```

返回:

类型:

Array.<Object>

destroy()

描述:

-
-

销毁

-
-

示例

```
dsResult.destroy();
dsResult = null;
```

PlaceSearch

new PlaceSearch (props)

描述:

-
-

地点检索

-
-

示例

```
let placeSearch = new KMap.PlaceSearch({
  map: map,
  panel: "searchResultPanel-list",
  lang: "zh-cn"
})
```

参数:

名称	类型	描述
props	propsObject	

propsObject :

名称	类型	描述
lang	String	语言设置, 可选 zh-cn,en
map	Map	呈现地图对象
panel	String, HTMLElement	呈现列表面板 ID
isFitView	Boolean	是否最佳视野呈现

函数

```
search(options,callback)
```

描述:

-
-

关键字全国检索

-
-

示例

```
placeSearch.search({
  category: '西餐厅|快餐',
  keyword: '肯德基',
  start: 0,
  count: 100,
},(status, result)=>{console.log(status, result);})
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
<code>options</code>	<code>optionsObject</code>	
<code>callback</code>	<code>function</code>	回调方法，参数 <code>status,result</code>

`optionsObject` :

名称	类型	描述
<code>category</code>	<code>string</code>	分类编码，多个分类及分类编码用 隔开
<code>keyword</code>	<code>keyword</code>	关键字
<code>start</code>	<code>number</code>	结果起始位置，从 0 开始，默认 0
<code>count</code>	<code>number</code>	返回结果数，默认 10

返回:

类型:

`Object`

`searchInRegion(options,callback)`

描述:

-
-

指定城市，关键字搜索

-
-

示例

```
placeSearch.searchInRegion({
```

```
city:'深圳市',
category:'西餐厅|快餐',
keyword:'肯德基',
start:0,
count:100,
},(status, result)=>{console.log(status, result);})
```

参数:

名称	类型	描述
<code>options</code>	<code>optionsObject</code>	
<code>callback</code>	<code>function</code>	回调方法，参数 <code>status,result</code>

optionsObject :

名称	类型	描述
<code>city</code>	<code>string</code>	查询城市，可输入城市名或行政区划代码(adcode)
<code>category</code>	<code>string</code>	分类编码，多个分类及分类编码用 隔开
<code>keyword</code>	<code>keyword</code>	关键字
<code>start</code>	<code>number</code>	结果起始位置，从 0 开始，默认 0
<code>count</code>	<code>number</code>	返回结果数，默认 10

返回:

类型:

`Object`

searchInBounds(*options,callback*)

描述:

-
-

指定范围关键字搜索

-
-

示例

```
placeSearch.searchInBounds({
  bounds:new KMap.Bounds(new
KMap.Point(113.94716877057552,22.527208100315054),new
KMap.Point(113.94851555841635,22.52788972057429))
  city:'深圳市',
  category:'西餐厅|快餐',
  keyword:'肯德基',
  start:0,
  count:100,
}),(status, result)=>{console.log(status, result);})
```

参数:

名称	类型	描述
options	optionsObject	
callback	function	回调方法，参数 status,result

optionsObject :

名称	类型	描述
bounds	Bounds	范围
city	string	查询城市，可输入城市名或行政区划分代码(adcode)
category	string	分类编码，多个分类及分类编码用 隔开
keyword	keyword	关键字

名称	类型	描述
start	number	结果起始位置，从 0 开始，默认 0
count	number	返回结果数，默认 10

返回:

类型:

Object

`searchAround(options,callback)`

描述:

-
-

中心点经纬度、半径以及关键字进行周边查询

-
-

示例

```
placeSearch.searchAround({
  circle: new KMap.Circle(
    {center: new KMap.Point(114.04242926984897,
22.527070016303945), radius: 10000}
  ),
  city: '深圳市',
  category: '西餐厅|快餐',
  keyword: '肯德基',
  start: 0,
  count: 100,
}, (status, result) => { console.log(status, result); })
```

参数:

名称	类型	描述
options	optionsObject	

名称	类型	描述
<code>callback</code>	<code>function</code>	回调方法，参数 <code>status,result</code>

optionsObject :

名称	类型	描述
<code>circle</code>	<code>Circle</code>	范围
<code>city</code>	<code>string</code>	查询城市，可输入城市名或行政区划分代码(adcode)
<code>category</code>	<code>string</code>	分类编码，多个分类及分类编码用 隔开
<code>keyword</code>	<code>keyword</code>	关键字
<code>start</code>	<code>number</code>	结果起始位置，从 0 开始，默认 0
<code>count</code>	<code>number</code>	返回结果数，默认 10

返回:

类型:

`Object`

`searchInPolygon(options,callback)`

描述:

-
-

在多边形区域内搜索

-
-

示例

```
placeSearch.searchInPolygon({
  polygon: new
KMap.Polygon({path:[[{lon":113.6865545627288,"lat":22.659696953037074,
"alt":0},{lon":114.20911508145574,"lat":22.659696953037074,"alt":0},{
lon":114.20911508145574,"lat":22.395224979652955,"alt":0},{lon":113.68
65545627288,"lat":22.395224979652955,"alt":0},{lon":113.6865545627288,
"lat":22.659696953037074,"alt":0}]]]}),
  city:'深圳市',
  category:'西餐厅|快餐',
  keyword:'肯德基',
  start:0,
  count:100,
}),(status, result)=>{console.log(status, result);})
```

参数:

名称	类型	描述
options	optionsObject	
callback	function	回调方法，参数 status,result

optionsObject :

名称	类型	描述
polygon	Polygon	多边形
city	string	查询城市，可输入城市名或行政区划分代码(adcode)
category	string	分类编码，多个分类及分类编码用 隔开
keyword	keyword	关键字
start	number	结果起始位置，从 0 开始，默认 0

名称	类型	描述
count	number	返回结果数，默认 10

返回:

类型:

Object

searchAlong(options,callback)

描述:

-
-

沿线搜索

-
-

示例

```
placeSearch.searchAlong({
  polyline: new
KMap.Polyline({path:[[114.02389949184,22.53858759938],[114.02796096652,
22.53556256734],[114.02326913347,22.53712754874],[114.02595124550,22.53
444285161],[114.03457877464,22.52998651301]]}),
  range: 1000,
  city: '深圳市',
  category: '西餐厅|快餐',
  keyword: '肯德基',
  start: 0,
  count: 100,
},(status, result)=>{console.log(status, result);})
```

参数:

名称	类型	描述
options	optionsObject	
callback	function	回调方法，参数 status,result

optionsObject :

名称	类型	描述
polyline	Polyline	线
city	string	查询城市，可输入城市名或行政区划分代码(adcode)
category	string	分类编码，多个分类及分类编码用 隔开
keyword	keyword	关键字
range	number	沿途搜索范围，单位米
start	number	结果起始位置，从 0 开始，默认 0
count	number	返回结果数，默认 10

返回:

类型:
Object

```
searchId(id,callback)
```

描述:

-
-

获取 Poi 详情

-
-

示例

```
placeSearch.searchId(2318745,(status,data) => {console.log(status,data);});
```

参数:

名称	类型	描述
id	String	poi 唯一 ID
callback	function	回调方法，参数 status,result

返回:

类型: Object

```
clearResults()
```

描述:

-
-

清除结果

-
-

示例

```
placeSearch.clearResults();
```

```
destroy()
```

描述:

-
-

销毁

-
-

示例

```
placeSearch.destroy();  
placeSearch = null;
```

PlaceSearchResult

new PlaceSearchResult (json,params)

描述:

-
-

地点检索结果

-
-

参数:

名称	类型	描述
json	object	
params	object	

函数

getRequestParams()

描述:

-
-

获取请求参数

-

-

示例

```
psResult.getRequestParams();
```

返回:

类型:

Object

getBounds()

描述:

-
-

返回请求结果视野

-
-

示例

```
psResult.getBounds();
```

返回:

类型:

Bounds

getTotal()

描述:

-
-

检索到总记录数

-
-

示例

```
psResult.getTotal();
```

返回:

类型:
Number

getCount()

描述:

-
-

返回 POI 记录数

-
-

示例

```
psResult.getCount();
```

返回:

类型:
number

getPoi(index)

描述:

-
-

依据下标获取 poi

-
-

示例

```
psResult.getPoi(0);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
index		

返回:

类型:
undefined, Object

```
getPois()
```

描述:

- -
- 获取返回的 POI 列表

-
-

示例

```
psResult.getPois();
```

返回:

类型:
Array.<Object>

```
getSuggestCites()
```

描述:

- -
- 获取建议检索城市

-
-

示例

```
psResult.getSuggestCites();
```

返回:

类型:

Array.<Object>

getArea()

描述:

-
-

获取检索结果区域信息

-
-

示例

```
psResult.getArea();
```

返回:

类型:

Object

destroy()

描述:

-
-

销毁

-
-

示例

```
psResult.destroy();  
psResult = null;
```

WeatherSearch

new WeatherSearch (props)

描述:

-
-

天气检索

-
-

示例

```
let weather = new WeatherSearch();
let weather = new WeatherSearch({lang:'zh-cn'});
```

参数:

名称	类型	描述
props	propsObject	

propsObject :

名称	类型	描述
lang	String	语言设置, 可选 zh-cn,en

函数

search(params,callback)

描述:

-
-

检索

-
-

示例

```
weather.search({adcode: '440300'}, (status, result) => { console.log(status, result); })
```

参数:

名称	类型	描述
params	paramsObject	
callback	function	回调方法，参数 status,result

paramsObject :

名称	类型	描述
adcode	String	行政区名称，或者行政区编码,全国传入空

返回:

类型:
Object

destroy()

描述:

-
-

销毁

-
-

示例

```
weather.destroy();  
destroy = null;
```

WeatherSearchResult

new WeatherSearchResult (json,params)

描述:

-
-

天气检索结果

-
-

参数:

名称	类型	描述
json	object	
params	object	

函数

getRequestParams()

描述:

-
-

获取请求参数

-
-

示例

`weaResult.getRequestParams();`

返回:

类型:
Object

getCount()

描述:

-
-

获取检索返回结果数量

-
-

示例

```
weaResult.getCount();
```

返回:

类型:

number

getResults()

描述:

-
-

获取返回结果列表

-
-

示例

```
weaResult.getResults();
```

返回:

类型:

Array.<Object>

Transit

new Transit (options)

描述:

-
-

公交地铁规划服务

-
-

示例

```
let transit = new KMap.Transit({
  map : map,
  panel: 'panel',
  draggable : true
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
map	Map	地图
policy	TransitPolicy	公交规划策略
panel	string	结果显示面板 ID
autoFitView	boolean	是否自动调整视图，默认 true
city	string	城市

名称	类型	描述
<code>draggable</code>	<code>boolean</code>	是否可以拖动图标重新规划，默认 <code>false</code>

函数

```
clear()
```

描述:

-
-

清空结果和搜索参数

-
-

示例

```
transit.clear();
```

```
destroy()
```

描述:

-
-

销毁

-
-

示例

```
transit.destroy();
```

```
setDraggable(draggable)
```

描述:

-
-

是否可拖动

-
-

示例

```
transit.setDraggable(true);
```

参数:

名称	类型	描述
draggable	boolean	是否可拖动改变路线

```
setCity(city)
```

描述:

-
-

设置城市

-
-

示例

```
transit.setCity('深圳市');
```

参数:

名称	类型	描述
city	string	城市名称或编码

setPolicy(policy)

描述:

-
-

设置策略

-
-

示例

```
transit.setPolicy(KMap.TransitPolicy.LEAST_TRANSFER);
```

参数:

名称	类型	描述
policy	TransitPolicy	策略, TransitPolicy 枚举

search(origin,destination,callback)

描述:

-
-

规划路线

-
-

示例

```
let start = new KMap.Point(114.02306542,22.53496495);
let end = new KMap.Point(114.11779759,22.53132889);
transit.search(start,end,function(status,result) {
    if(status == 'complete') {
        //结果
        console.log(result);
    }
    else {
```

```
        //错误
        alert(result);
    }
});
```

参数:

名称	类型	描述
origin	Point, Array.<Array.<object>>	起点或起终点数组信息
destination	Point	终点
callback	Callback	结果回调

TransitStep

属性

```
type
```

描述:

-
-

段类别 (walk, bus, metro)

-
-

```
id
```

描述:

-
-

线路 ID(步行为 0)

-
-

name

描述:

-
-

线路名称

-
-

start

描述:

-
-

起点

-
-

end

描述:

-
-

终点

-
-

time

描述:

-
-

时间，单位秒

-
-

distance

描述:

-
-

里程，单位米

-
-

stations

描述:

-
-

经过的站点

-
-

path

描述:

-
-

形状点

-

-

color

描述:

-
-

线颜色

-
-

TransitRoute

属性

id

描述:

-
-

方案 ID

-
-

name

描述:

-
-

名称

-
-

origin

描述:

-
-

上车站

-
-

destination

描述:

-
-

下车站

-
-

distance

描述:

-
-

总里程，单位米

-
-

walkDistance

描述:

-
-

步行里程，单位米

-
-

time

描述:

-
-

总时间，单位秒

-
-

tolls

描述:

-
-

费用，单位分

-
-

tag

描述:

-
-

标签

-
-

steps

描述:

-
-

行程

-
-

TransferResult

属性

origin

描述:

-
-

起点

-
-

destination

描述:

-
-

终点

-
-

routes

描述:

-
-

方案

-
-

DrivingRoute

属性

id

描述:

-
-

路径 ID

-
-

distance

描述:

-
-

里程，单位米

-
-

time

描述:

-
-

时间，单位秒

-
-

tolls

描述:

-
-

人工收费，单位分

-
-

etc

描述:

-
-

ETC 收费，单位分

-
-

trafficLight

描述:

-
-

交通灯数

-

-

tag

描述:

-
-

标签

-
-

steps

描述:

-
-

行程段

-
-

segments

描述:

-
-

道路段

-
-

districts

描述:

-

-

城市段

-
-

restricts

描述:

-
-

限行信息

-
-

tips

描述:

-
-

提示信息

-
-

mainRoads

描述:

-
-

主要经过路段

-
-

TruckDrivingRoute

属性

id

描述:

-
-

路径 ID

-
-

distance

描述:

-
-

里程，单位米

-
-

time

描述:

-
-

时间，单位秒

-
-

tolls

描述:

-
-

人工收费，单位分

-
-

etc

描述:

-
-

ETC 收费，单位分

-
-

trafficLight

描述:

-
-

交通灯数

-
-

tag

描述:

-
-

标签

-
-

steps

描述:

-
-

行程段

-
-

segments

描述:

-
-

道路段

-
-

districts

描述:

-
-

城市段

-
-

restricts

描述:

-
-

限行信息

-
-

tips

描述:

-
-

提示信息

-
-

mainRoads

描述:

-
-

主要经过路段

-
-

覆盖物

Contextmenu

new Contextmenu (props)

描述:

-
-

右键菜单

-
-

示例

```
let contextmenu = new KMap.Contextmenu({isCustom: false})
contextmenu.addItem("放大一级", (e) => {let zoom =
map.getZoom();map.setZoom(zoom + 1, 500);map.close(contextmenu)}));
map.open(contextmenu,new KMap.Point(114.02308, 22.53474, 0));
map.close(contextmenu);
```

参数:

名称	类型	描述
props	propsObject	

propsObject :

名称	类型	描述
id	String	对象 ID 值。默认值由系统分配
isCustom	Boolean	是否自定义菜单，默认: false，设为 true 时，菜单内容完全按照 content 所设的值添加
content	String	显示内容，可以是 HTML 要素字符串或者 HTMLElement 对象

函数

addItem(text,fn,index)

描述:

-
-

添加选项

-
-

示例

```
contextmenu.addItem("缩小一级", () => {let zoom = map.getZoom();map.setZoom(zoom - 1, 500);map.close(contextmenu)});
contextmenu.addItem("旋转 15 度", () => {let rotate = map.getRotation();map.setRotation(rotate + 15, 500);map.close(contextmenu)}}, 1);
```

参数:

名称	类型	描述
text	String	文本
fn	function	回调方法
index	Number	插入位置,从 0 开始

removeItem(text)

描述:

-
-

移除选项

-

-

示例

```
contextmenu.removeItem("旋转 15 度");
```

参数:

名称	类型	描述
text	String	

```
getItems()
```

描述:

-
-

获取设置的选项

-
-

示例

```
contextmenu.getItems();
```

返回:

类型:
Object

```
setContent(content)
```

描述:

-
-

设置内容

-
-

示例

```
contextmenu.setContent("<div>这是右键菜单</div>");
```

参数:

名称	类型	描述
content	string, HTMLElement	

```
getContent()
```

描述:

-
-

获取内容

-
-

示例

```
map.getContent();
```

返回:

类型:
string, HTMLElement

```
getId()
```

描述:

-
-

返回覆盖物 ID

-
-

返回:

类型：
string

destroy()

描述:

-
-

销毁

-
-

示例

obj.destroy();

getPosition()

描述:

-
-

获取对象纬度度坐标

-
-

示例

obj.getPosition();

返回:

类型：
Point

setExtData(extData)

描述:

-

-

设置用户自定义属性

-
-

示例

```
obj.setExtData(Any)
```

参数:

名称	类型	描述
----	----	----

extData	Any	
---------	-----	--

```
getExtData()
```

描述:

-
-

获取用户自定义属性

-
-

示例

```
obj.getExtData()
```

返回:

类型:

Any

Icon

```
new Icon (props)
```

描述:

-
-

图标

-
-

示例

```
let icon = new KMap.Icon({
  size: new KMap.Size(36 / 2, 36 / 2),
  imageOffset: new KMap.Pixel(-260 / 2, -396 / 2),
  image: '/assets/images/compass.png',
  imageSize: new KMap.Size(696 / 2, 540 / 2)
});
```

参数:

名称	类型	描述
props	propsObject	

propsObject :

名称	类型	描述
size	Size, Vector2	图标尺寸，默认值(36,36)
imageOffset	Pixel, Vector2	图标取图偏移量
image	URL	图标的取图地址
imageSize	URL	图标所用图片大小

函数

setImageSize(size)

描述:

-
-

设置图标尺寸

-
-

示例

```
icon.setImageSize(new KMap.Size(696,540));
icon.setImageSize([696,540]);
```

参数:

名称	类型	描述
size	Size, Vector2	

```
getImageSize()
```

描述:

-
-

获取图片大小

-
-

示例

```
icon.getImageSize();
```

返回:

类型:
Size

```
setSize(size)
```

描述:

-

-

设置图标尺寸

-
-

示例

```
icon.setSize(new KMap.Size(36,36));  
icon.setSize([36,36]);
```

参数:

名称	类型	描述
size	Size, Vector2	

getSize()

描述:

-
-

获取图标尺寸

-
-

示例

```
icon.getSize();
```

返回:

类型:

Size

setImageOffset(offset)

描述:

-
-

设置图标取图偏移量

-
-

示例

```
icon.setImageOffset(new KMap.Pixel(-260,-396));  
icon.setImageOffset([-260,-396]);
```

参数:

名称	类型	描述
offset	Pixel, Vector2	

```
getImageOffset()
```

描述:

-
-

获取图标取图偏移量

-
-

示例

```
icon.getImageOffset();
```

返回:

类型:
Pixel

```
setImage(image)
```

描述:

-
-

设置图标的取图地址

-
-

示例

```
icon.setSize("http://www.careland.com.cn/assets/images/compass.png");
```

参数:

名称	类型	描述
image	URL	

```
getImage()
```

描述:

-
-

获取图标的取图地址

-
-

示例

```
icon.getImage();
```

返回:

类型:
URL

InfoWindow

```
new InfoWindow (props)
```

描述:

-
-

信息窗口

-
-

示例

```
let infowin = new KMap.InfoWindow({
  position: new KMap.Point(114.02308, 22.53474, 0),
  anchor: 'bottom-center',
  closeWhenClickMap: false,
  closeButton: true,
  offset: new KMap.Pixel(0, -42),
  content: '<div style="height:100px;width:200px;">这是
infowindow<br><input onkeypress="console.log(this.value)"
style="zIndex:123"> </div>'
});
map.open(infowin,new KMap.Point(114.02308, 22.53474, 0));
map.close(infowin);
```

参数:

名称	类型	描述
props	propsObject	

propsObject :

名称	类型	描述
id	String	对象 ID 值。默认值由系统分配
isCustom	Boolean	是否自定义窗体，默认: false ，设为 true 时，窗体内容完全按照 content 所设的值添加，否则，在窗体外框中显示内容
content	String	显示内容，可以是 HTML 要素字符串或者 HTMLElement 对象
autoMove	Boolean	是否自动调整窗体到视野内

名称	类型	描述
<code>avoid</code>	<code>Array.<Number></code>	是否自动平移，默认： <code>true</code> ， <code>autoMove</code> 为 <code>true</code> 时，自动平移到视野内后的上右下左的避让宽度
<code>closeWhenClickMap</code>	<code>Boolean</code>	鼠标点击地图后关闭信息窗体，默认： <code>false</code>
<code>size</code>	<code>Size</code>	窗体尺寸(<code>isCustom</code> 为 <code>true</code> 时，该属性无效)，默认： <code>undefined</code>
<code>anchor</code>	<code>String</code>	信息窗体锚点，默认： <code>top-left</code> ， 可选值： <code>'top-left' 'top-center' 'top-right' 'middle-left' 'center' 'middle-right' 'bottom-left' 'bottom-center' 'bottom-right'</code>
<code>offset</code>	<code>Pixel, Vector2</code>	窗体显示位置偏移量，默认： <code>new Pixel(0, 0)</code>

函数

`setPosition(position)`

描述:

-

-

设置悬停位置

-
-

示例

```
infowin.setPosition(new KMap.new KMap.Point(114.02308, 22.53474, 0));  
infowin.setPosition([114.02308, 22.53474]);
```

参数:

名称	类型	描述
----	----	----

position	Point, Vector2	
----------	----------------	--

getIsOpen()

描述:

-
-

获取信息窗体是否打开

-
-

示例

```
infowin.getIsOpen();
```

返回:

类型:
Boolean

setSize(size)

描述:

-
-

设置信息窗体大小（isCustom 为 false 时有效）

-
-

示例

```
infowin.setSize(new KMap.Size(500,450));
infowin.setSize([500,450]);
```

参数:

名称	类型	描述
size	Size, Vector2	

```
getSize()
```

描述:

-
-

获取信息窗体大小

-
-

示例

```
infowin.getSize();
```

返回:

类型: Size

```
setContent(content)
```

描述:

-
-

设置自定义内容

-
-

示例

```
infowin.setContent("<div>info window content</div>");
```

参数:

名称	类型	描述
content	String, HTMLElement	

getContent()

描述:

-
-

获取自定义内容

-
-

示例

```
infowin.getContent();
```

返回:

类型:

String, HTMLElement

setAnchor(anchor)

描述:

-
-

设置信息窗体锚点

-
-

示例

```
infowin.setAnchor("top-center");
```

参数:

名称	类型	描述
anchor	String	

```
getAnchor()
```

描述:

-
-

获取锚点位置

-
-

示例

```
infowin.getAnchor();
```

返回:

类型: String

```
setOffset(offset)
```

描述:

-
-

设置偏移量

-
-

示例

```
infowin.setOffset(new KMap.Pixel(-10,-10));
```

```
infowin.setOffset([-10,-10]);
```

参数:

名称	类型	描述
offset	Pixel, Vector2	

```
getOffset()
```

描述:

-
-

获取偏移量

-
-

示例

```
infowin.getOffset();
```

返回:

类型:
Pixel

```
getId()
```

描述:

-
-

返回覆盖物 ID

-
-

返回:

类型:
string

destroy()

描述:

-
-

销毁

-
-

示例

```
obj.destroy();
```

getPosition()

描述:

-
-

获取对象纬度度坐标

-
-

示例

```
obj.getPosition();
```

返回:

类型:

Point

setExtData(extData)

描述:

-
-

设置用户自定义属性

-
-

示例

```
obj.setExtData(Any)
```

参数:

名称	类型	描述
extData	Any	

```
getExtData()
```

描述:

-
-

获取用户自定义属性

-
-

示例

```
obj.getExtData()
```

返回:

类型:
Any

Marker

```
new Marker (props)
```

描述:

-
-

点标记

-
-

示例

```
marker = new KMap.Marker({
  position: new KMap.Point(114.02308, 22.53474,0),
  anchor: 'center',
  angle: 0,
  offset: new KMap.Pixel(0, 0),
  title: '我是 title',
  cursor: 'pointer',
  bubble: true,
  visible: true,
  zooms: [4, 21],
  icon: new KMap.Icon({
    size: new KMap.Size(13, 13),
    imageOffset: new KMap.Pixel(-2, -2),
    image: '/assets/images/stadiometry.png',
    imageSize: new KMap.Size(36, 15)
  }),
  content: '',
  label: {
    content: '<div style="color:red;background:
#ffffff;border:1px solid #ccc;padding:5px;">这是 marker label</div>',
    direction: 'top'
  },
  topWhenClick: true
});
map.add(marker);
```

参数:

名称	类型	描述
props	propsObject	

propsObject :

名称	类型	描述
id	String	对象 ID 值，默认值由系统分配
position	Point, Vector2	纬度度坐标，默认值为空
title	string	鼠标滑过时的文字提示，默认值为空

名称	类型	描述
<code>visible</code>	<code>Boolean</code>	是否可见，默认值: <code>true</code>
<code>zooms</code>	<code>Vector2</code>	显示的缩放级别范围
<code>zIndex</code>	<code>Number</code>	叠加顺序，默认值: <code>12</code>
<code>offset</code>	<code>Pixel, Vector2</code>	显示位置偏移量，默认值为: <code>[0,0]</code>
<code>anchor</code>	<code>String</code>	锚点，默认值: <code>'top-left'</code> ，可选值: <code>'top-left'</code> , <code>'top-center'</code> , <code>'top-right'</code> , <code>'middle-left'</code> , <code>'center'</code> , <code>'middle-right'</code> , <code>'bottom-left'</code> , <code>'bottom-center'</code> , <code>'bottom-right'</code>
<code>angle</code>	<code>Number</code>	旋转角度，默认值: <code>0</code>
<code>cursor</code>	<code>String</code>	鼠标悬停时样式，默认值: <code>pointer</code>
<code>bubble</code>	<code>Boolean</code>	事件是否冒泡，默认为 <code>false</code>
<code>clickable</code>	<code>Boolean</code>	点标记是否可点击，默认值: <code>true</code>
<code>draggable</code>	<code>Boolean</code>	是否可拖拽移动，默认值: <code>false</code>

名称	类型	描述
<code>topWhenClick</code>	<code>Boolean</code>	点击时是否置顶，默认值： <code>false</code>
<code>icon</code>	<code>Icon, URL</code>	图标，可以是图标地址，也可以传 <code>Icon</code> 对象，当设置了合法的 <code>Content</code> 时，此属性无效
<code>content</code>	<code>String, HTMLElement</code>	显示内容
<code>label</code>	<code>propsLabelObject</code>	标注
<code>extData</code>	<code>Any</code>	自定义属性

propsLabelObject :

名称	类型	描述
<code>content</code>	<code>String, HTMLElement</code>	标注内容
<code>offset</code>	<code>Pixel, Vector2</code>	偏移量，默认值： <code>new Pixel(0, 0)</code>
<code>direction</code>	<code>String</code>	标注方位，默认值： <code>right</code> ，可选值： 'top' 'right' 'bottom' 'left' 'center'

函数

`setPosition(position)`

描述:

-
-

设置在地图上显示的位置

-
-

示例

```
marker.setPosition(new KMap.new KMap.Point(114.02308, 22.53474, 0));
marker.setPosition([114.02308, 22.53474]);
```

参数:

名称	类型	描述
position	Point	

```
setIcon(icon)
```

描述:

-
-

设置点标记的显示图标

-
-

示例

```
marker.setIcon(new KMap.Icon({
  size: new KMap.Size(23, 29),
  imageOffset: new KMap.Pixel(0, 0),
  image: 'https://open.careland.com.cn/docs/new/image/poiMarker.png',
  imageSize: new KMap.Size(23, 29)
}));
marker.setIcon('https://open.careland.com.cn/docs/new/image/poiMarker.png');
```

参数:

名称	类型	描述
icon	Icon, URL	

getIcon()

描述:

-
-

获取图标

-
-

示例

```
marker.getIcon();
```

返回:

类型:

Icon, URL

setContent(content)

描述:

-
-

设置点标记显示的自定义内容

-
-

示例

```
marker.setContent('<div>Marker Content</div>')
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
icon	HTMLElement, String	

getContent()

描述:

-
-

获取点标记显示的自定义内容

-
-

示例

```
marker.getContent()
```

返回:

类型:

string

setTitle(title)

描述:

-
-

设置鼠标滑过点标记时的文字提示

-
-

示例

```
marker.setTile("Marker Title");
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
<code>title</code>	<code>String</code>	

`getTitle()`

描述:

-
-

获取鼠标滑过点标记时的文字提示

-
-

示例

```
marker.getTitle();
```

返回:

类型:

`string`

`show()`

描述:

-
-

显示覆盖物

-
-

示例

```
marker.show();
```

`hide()`

描述:

-
-

隐藏覆盖物

-
-

示例

```
marker.hide();
```

setTop(isTop)

描述:

-
-

设置是否置顶

-
-

示例

```
marker.setTop(true);
```

参数:

名称	类型	描述
isTop	Boolean	

getTop()

描述:

-
-

获取该点标记是否置顶

-

-

示例

```
marker.getTop();
```

```
setTopWhenClick(isTop)
```

描述:

-
-

设置是否点击置顶

-
-

示例

```
marker.setTopWhenClick(false);
```

参数:

名称	类型	描述
----	----	----

isTop	Boolean	
-------	---------	--

```
setZIndex(zindex)
```

描述:

-
-

设置叠加顺序

-
-

示例

```
marker.setZIndex(20);
```

参数:

名称	类型	描述
zindex	Number	

getZIndex()

描述:

-
-

获取叠加顺序

-
-

示例

```
marker.getZIndex();
```

返回:

类型:

Number

setOffset(offset)

描述:

-
-

设置覆偏移量

-
-

示例

```
marker.setOffset(new KMap.Pixel(10,10));  
marker.setOffset([10,10]);
```

参数:

名称	类型	描述
offset	Pixel, Vector2	

getOffset()

描述:

-
-

获取偏移量

-
-

示例

```
marker.getOffset();
```

返回:

类型:

Pixel

setAnchor(anchor)

描述:

-
-

设置锚点位置

-
-

示例

```
marker.setAnchor('center');
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
anchor	String	

getAnchor()

描述:

-
-

获取锚点位置

-
-

示例

```
marker.getAnchor();
```

返回:

类型:

String

setAngle(*angle*)

描述:

-
-

设置旋转角度

-
-

示例

```
marker.setAngle(90);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
angle	Number	

getAngle()

描述:

-
-

获取覆盖物旋转角度

-
-

示例

marker.getAngle();

返回:

类型:

Number

setClickable(clickable)

描述:

-
-

设置点标记是否支持鼠标单击事件

-
-

示例

marker.setClickable(false);

参数:

名称	类型	描述
----	----	----

名称	类型	描述
clickable	Boolean	

getClickable()

描述:

-
-

获取点标记是否支持鼠标单击事件

-
-

示例

```
marker.getClickable();
```

返回:

类型:
Boolean

setDraggable(draggable)

描述:

-
-

设置点标记对象是否可拖拽移动

-
-

示例

```
marker.setDraggable(false);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
draggable	Boolean	

getDraggable()

描述:

-
-

获取点标记对象是否可拖拽移动

-
-

示例

```
marker.getDraggable();
```

返回:

类型:
Boolean

setBubble(bubble)

描述:

-
-

设置事件是否冒泡

-
-

示例

```
marker.setBubble(true);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
bubble	Boolean	

getBubble()

描述:

-
-

获取事件是否冒泡

-
-

示例

```
marker.getBubble();
```

返回:

类型:

Boolean

setZooms(min,max)

描述:

-
-

设置点标记显示的层级范围

-
-

示例

```
marker.setZooms(10,21)
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
<code>min</code>	<code>Boolean</code>	最小值
<code>max</code>	<code>Boolean</code>	最大值

```
getZooms()
```

描述:

-
-

获取点标记显示的层级范围

-
-

示例

```
marker.getZooms();
```

返回:

类型:

`Vector2`

```
setCursor(cursor)
```

描述:

-
-

设置指定鼠标悬停时样式

-
-

示例

```
marker.setCursor("pointer");
```

参数:

名称	类型	描述
cursor	String	

getCursor()

描述:

-
-

获取指定鼠标悬停时样式

-
-

示例

```
marker.getCursor();
```

返回:

类型:
String

setLabel(label)

描述:

-
-

设置文本标注

-
-

示例

```
marker.setLabel({content: '<div style="color:blueviolet">marker 9050</div>', direction: 'right'})
marker.setLabel(null);
```

参数:

名称	类型	描述
label	Object	

getLabel()

描述:

-
-

获取文本标注

-
-

示例

```
marker.getLabel();
```

返回:

类型: Object

getId()

描述:

-
-

返回覆盖物 ID

-
-

返回:

类型: string

destroy()

描述:

-
-

销毁

-
-

示例

```
obj.destroy();
```

getPosition()

描述:

-
-

获取对象纬度度坐标

-
-

示例

```
obj.getPosition();
```

返回:

类型:

Point

setExtData(*extData*)

描述:

-
-

设置用户自定义属性

-
-

示例

```
obj.setExtData(Any)
```

参数:

名称	类型	描述
extData	Any	

```
getExtData()
```

描述:

-
-

获取用户自定义属性

-
-

示例

```
obj.getExtData()
```

返回:

类型:
Any

事件

```
moving()
```

描述:

-
-

覆盖物在执行 moveTo, moveAlong 动画时触发事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
moveend()
```

描述:

-
-

覆盖物执行 moveTo 动画结束时触发事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
movealong()
```

描述:

-
-

覆盖物执行 moveAlong 动画一次后触发事件

-
-

参数:

名称	类型	描述
e	Object	结果

TextMarker

`new TextMarker (props)`

描述:

-
-

文本标记

-
-

示例

```
let textMarker = new KMap.TextMarker({
  position: new KMap.Point(114.02308, 22.53474, 0),
  anchor: 'left-top',
  angle: 0,
  offset: new KMap.Pixel(0, 0),
  title: '我是 title',
  visible: true,
  draggable: true,
  cursor: 'move',
  zooms: [4, 21],
  text: '我是 Text',
  style: {background: '#cccccc',padding: '10px',color: '#f00',display:
'block'},
  topWhenClick: true
});
map.add(textMarker);
```

参数:

名称	类型	描述
props	propsObject	

propsObject :

名称	类型	描述
<code>id</code>	<code>String</code>	对象 ID 值，默认值由系统分配
<code>position</code>	<code>Point, Vector2</code>	纬度度坐标，默认值为空
<code>title</code>	<code>string</code>	鼠标滑过时的文字提示，默认值为空
<code>visible</code>	<code>Boolean</code>	是否可见，默认值： <code>true</code>
<code>zooms</code>	<code>Vector2</code>	显示的缩放级别范围
<code>zIndex</code>	<code>Number</code>	叠加顺序，默认值： <code>12</code>
<code>offset</code>	<code>Pixel, Vector2</code>	显示位置偏移量，默认值： <code>[0,0]</code>
<code>anchor</code>	<code>String</code>	锚点，默认值： <code>'top-left'</code> ，可选值： <code>'top-left'</code> , <code>'top-center'</code> , <code>'top-right'</code> , <code>'middle-left'</code> , <code>'center'</code> , <code>'middle-right'</code> , <code>'bottom-left'</code> , <code>'bottom-center'</code> , <code>'bottom-right'</code>
<code>angle</code>	<code>Number</code>	旋转角度，默认值： <code>0</code>
<code>cursor</code>	<code>String</code>	鼠标悬停时样式，默认值： <code>'pointer'</code>

名称	类型	描述
<code>bubble</code>	<code>Boolean</code>	事件是否冒泡，默认为 <code>false</code>
<code>clickable</code>	<code>Boolean</code>	点标记是否可点击，默认值: <code>true</code>
<code>draggable</code>	<code>Boolean</code>	是否可拖拽移动，默认值: <code>false</code>
<code>topWhenClick</code>	<code>Boolean</code>	点击时是否置顶，默认值: <code>false</code>
<code>text</code>	<code>String</code>	显示内容
<code>style</code>	<code>Object</code>	文本样式， <code>Object</code> 同 <code>css</code> 样式表

函数

`setPosition(position)`

描述:

-
-

设置悬停位置

-
-

示例

```
textMarker.setPosition(new KMap.new KMap.Point(114.02308, 22.53474, 0));
textMarker.setPosition([114.02308, 22.53474]);
```

参数:

名称	类型	描述
<code>position</code>	<code>Point, Vector2</code>	

`setText(text)`

描述:

-
-

设置点标记显示的自定义内容

-
-

示例

```
textMarker.setText('这里是 text');
```

参数:

名称	类型	描述
<code>text</code>	<code>String</code>	

`getText()`

描述:

-
-

获取点标记显示的自定义内容

-
-

示例

```
textMarker.getText();
```

返回:

类型:

String

setTitle(title)

描述:

-
-

设置点标记的文字提示

-
-

示例

```
textMarker.setTitle("这里是 Title");
```

参数:

名称	类型	描述
----	----	----

title	String	
-------	--------	--

getTitle()

描述:

-
-

获取点标记的文字提示

-
-

示例

```
textMarker.getTitle();
```

返回:

类型:

String

show()

描述:

-
-

显示

-
-

示例

```
textMarker.show();
```

hide()

描述:

-
-

隐藏

-
-

示例

```
textMarker.hide();
```

setTop(isTop)

描述:

-
-

设置是否置顶该点标记

-
-

示例

```
textMarker.setTop(false);
```

参数:

名称	类型	描述
isTop	Boolean	

```
getTop()
```

描述:

-
-

获取是否置顶

-
-

示例

```
textMarker.getTop();
```

```
setTopWhenClick(isTop)
```

描述:

-
-

鼠标点击时是否置顶

-
-

示例

```
textMarker.setTopWhenClick(true);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
isTop	Boolean	

setZIndex(zindex)

描述:

-
-

设置叠加顺序

-
-

示例

```
textMarker.setZIndex(21);
```

参数:

名称	类型	描述
zindex	Number	

getZIndex()

描述:

-
-

获取叠加顺序

-
-

示例

```
textMarker.getZIndex();
```

返回:

类型：
Number

setOffset(offset)

描述:

-
-

设置偏移量

-
-

示例

```
textMarker.setOffset(new KMap.Pixel(10,10));  
textMarker.setOffset([10,10]);
```

参数:

名称	类型	描述
offset	Pixel, Vector2	

getOffset()

描述:

-
-

获取偏移量

-
-

示例

```
textMarker.getOffset();
```

返回:

类型:

Pixel,Vector2

setAnchor(anchor)

描述:

-
-

设置锚点位置

-
-

示例

textMarker.setAnchor('center');

参数:

名称	类型	描述
anchor	String	

getAnchor()

描述:

-
-

获取锚点位置

-
-

示例

textMarker.getAnchor();

返回:

类型:
String

setAngle(*angle*)

描述:

-
-

设置旋转角度

-
-

示例

```
textMarker.setAngle(45);
```

参数:

名称	类型	描述
angle	Number	

getAngle()

描述:

-
-

获取旋转角度

-
-

示例

```
textMarker.getAngle();
```

返回:

类型:
Number

setClickable(*clickable*)

描述:

-
-

设置点标记是否支持鼠标单击事件

-
-

示例

```
textMarker.setClickable(true);
```

参数:

名称	类型	描述
clickable	Boolean	

```
getClickable()
```

描述:

-
-

获取点标记是否支持鼠标单击事件

-
-

示例

```
textMarker.getClickable();
```

返回:

类型: Boolean

```
setDraggable(draggable)
```

描述:

-
-

设置点标记对象是否可拖拽移动

-
-

示例

```
textMarker.setDraggable(false);
```

参数:

名称	类型	描述
<code>draggable</code>	<code>Boolean</code>	

`getDraggable()`

描述:

-
-

获取点标记对象是否可拖拽移动

-
-

示例

```
textMarker.getDraggable();
```

返回:

类型:

`Boolean`

`setBubble(bubble)`

描述:

-
-

设置事件是否冒泡

-
-

示例

```
textMarker.setBubble(false);
```

参数:

名称	类型	描述
bubble	Boolean	

```
getBubble()
```

描述:

-
-

获取事件是否冒泡

-
-

示例

```
textMarker.getBubble();
```

返回:

类型:
Boolean

```
setZooms(min,max)
```

描述:

-
-

设置点标记显示的缩放层级范围

-
-

示例

```
textMarker.setZooms(10,21);
```

参数:

名称	类型	描述
min	number	最小值
max	number	最大值

```
getZooms()
```

描述:

-
-

获取点标记显示的层级范围

-
-

示例

```
textMarker.getZooms();
```

返回:

类型:

Vector2

```
setCursor(cursor)
```

描述:

-
-

设置指定鼠标悬停时样式

-
-

示例

```
textMarker.setCursor('pointer');
```

参数:

名称	类型	描述
cursor	String	

```
getCursor()
```

描述:

-
-

获取指定鼠标悬停时样式

-
-

示例

```
textMarker.getCursor();
```

返回:

类型:
string

```
setStyle(style)
```

描述:

-
-

设置样式

-
-

示例

```
textMarker.setStyle({color: '#f00',background: '#cccccc'});
```

参数:

名称	类型	描述
style	Object	

getStyle()

描述:

-
-

获取样式

-
-

示例

```
textMarker.getStyle();
```

返回:

类型:
*

getId()

描述:

-
-

返回覆盖物 ID

-
-

返回:

类型:

`string`

`destroy()`

描述:

-
-

销毁

-
-

示例

```
obj.destroy();
```

`getPosition()`

描述:

-
-

获取对象纬度度坐标

-
-

示例

```
obj.getPosition();
```

返回:

类型:

`Point`

`setExtData(extData)`

描述:

-
-

设置用户自定义属性

-
-

示例

```
obj.setExtData(Any)
```

参数:

名称	类型	描述
extData	Any	

```
getExtData()
```

描述:

-
-

获取用户自定义属性

-
-

示例

```
obj.getExtData()
```

返回:

类型:

Any

Circle

```
new Circle (options)
```

描述:

-
-

构造圆形对象，通过 options 指定多边形样式

-
-

示例

```
let circle = new KMap.Circle({
  center : new KMap.Point(114.02308, 22.53474),
  radius : 100,
  height: 20,
  cursor: 'pointer',
  bubble : true,
  style : {
    borderStyle: 'dashed',
    borderDasharray : [10,5],
    borderWidth:5,
    borderColor:"#000FFF7F",
    fillColor: "#FF00007F",
  }
})
map.add(circle);
```

参数:

名称	类型	描述
options	optionsObject	参数对象

optionsObject :

名称	类型	描述
id	String	ID 编号
zIndex	Number	覆盖物的叠加顺序
style	Array.<PolygonStyle>	圆样式表达数组，0 是顶面样式，其他为侧面样式 [{fillColor:"#FF00007F",borderWidth:2,borderColor:"#000FFF7F"},{fillColor:

名称	类型	描述
		<code>"rgba(0,255,55,0.5)"}]</code>
<code>height</code>	<code>Number</code>	立面高度，米 <code>eq:10</code>
<code>cursor</code>	<code>String</code>	鼠标移入样式,参考 <code>css</code> 鼠标样式
<code>bubble</code>	<code>Bubble</code>	事件是否冒泡
<code>tip</code>	<code>optionsTip Object</code>	鼠标移入提示
<code>draggable</code>	<code>Boolean</code>	否可拖拽移动，默认: <code>false</code> , <code>true</code> :可以拖拽， <code>false</code> :不可以拖拽
<code>visible</code>	<code>Boolean</code>	是否可见， <code>true</code> :显示， <code>false</code> :隐藏，默认显示
<code>zooms</code>	<code>Array.<Number></code>	显示级别范围，数组[最小比例尺，最大比例尺]， <code>eq:[4,20]</code>
<code>extData</code>	<code>Any</code>	自定义属性，用户自有属性，会原样返回
<code>center</code>	<code>Point</code>	圆心坐标， <code>eq: new KMap.Point(114.11555554,23.45662,50)</code>
<code>radius</code>	<code>Number</code>	圆半径，米,默认: <code>0</code> <code>eq:50</code>

名称	类型	描述
rotation:		[x:number,y:number:z:number] 旋转角度，弧度单位，eq: [Math.PI / 5,Math.PI / 2,Math.PI / 3]
revolution:		[x:number,y:number:z:number] 公转角度，弧度单位 eq: [Math.PI / 5,Math.PI / 2,Math.PI / 3]
position Offset:		[x:number,y:number:z:number] 位置偏移量，单位米 eq: [-200, 0, 0]

optionsStyleObject :

名称	类型	描述
borderColor	string	边线颜色 默认:#8aa8f5, eq: #FF00007F, rgba(0,255,55,0.5)
borderStyle	string	边线样式 solid,dashed,默认:solid
borderDasharray	Array.<Number>	虚线段 eq:[5,10]
borderWidth	Number	边线宽度,默认:3
fillColor	String	填充颜色 eq: #FF00007F, rgba(0,255,55,0.5) ,默认:#8af5bd

名称	类型	描述
<code>images</code>	<code>Array.<String></code>	纹理 url 地址数组， eq: ['url', 'url1']
<code>repeatWidth</code>	<code>Number</code>	纹理平铺宽度。 <code>-1</code> :拉伸， <code>0</code> :使用图片宽度平铺（默认）， 大于 <code>0</code> :指定平铺宽度
<code>repeatHeight</code>	<code>Number</code>	纹理平铺高度。 <code>-1</code> :拉伸（默认）， <code>0</code> :使用图片高度平铺， 大于 <code>0</code> :指定平铺高度

optionsTipObject :

名称	类型	描述
<code>text</code>	<code>String</code>	提示文字
<code>style</code>	<code>StyleSheet</code>	样式，参考 css 样式

函数

setCenter(center,noUpdate)

描述:

-
-

设置圆心

-

-

示例

```
circle.setCenter(new KMap.Point(114.02308, 22.53474));
```

参数:

名称	类型	描述
----	----	----

<code>center</code>	<code>Point</code>	圆心, eq: new KMap.Point(114.1155554,23.45662,50)
---------------------	--------------------	--

getCenter()

描述:

-

-

获取圆心

-

-

示例

```
circle.getCenter();
```

返回:

类型:

`Point`

setRadius(radius,noUpdate)

描述:

-

-

设置半径(米)

-
-

示例

```
circle.setRadius(100);
```

参数:

名称	类型	描述
radius	Number	半径，单位：米

```
getRadius()
```

描述:

-
-

获取半径，单位：米

-
-

示例

```
circle.getRadius();
```

返回:

类型:
Number

```
getHeight()
```

描述:

-
-

获取立面高度，单位：米

-

-

示例

```
circle.getHeight();
```

返回:

类型:

Number

getHeight(*height*)

描述:

-

-

设置立面高度，单位：米，eq: 100

-

-

示例

```
circle.setHeight(100);
```

参数:

名称	类型	描述
height	Number	立面高度

contains(*point*)

描述:

-

-

判断点是否在圆内

-

-

示例

```
circle.contains(new KMap.Point(114.02308, 22.53474));
```

参数:

名称	类型	描述
point	Point, Array.<Number>	经纬度 ， eq:new KMap.Point(114.02308, 22.53474)

返回:

类型:
boolean

getArea()

描述:

-
-

获取圆的面积 $S = \pi r^2$

-
-

示例

```
circle.getArea();
```

返回:

类型:
number

getBounds()

描述:

-
-

获取外框

-
-

示例

```
circle.getBounds();
```

返回:

类型:
 Bounds

getDepthStatus()

描述:

-
-

获取深度开关

-
-

返回:

类型:
 boolean

setDepthStatus(value)

描述:

-
-

设置深度开关

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
value	boolean	

setDraggable(*b*)

描述:

-
-

设置是否可以拖拽

-
-

参数:

名称	类型	描述
b		

getDraggable()

描述:

-
-

获取是否可以拖拽

-
-

返回:

类型:
boolean,*

setOptions(*opts*)

描述:

-
-

设置参数

-
-

参数:

名称	类型	描述
opts		

setTip(*tip*)

描述:

-
-

设置鼠标移入提示

-
-

示例

```
polyline.setTip({text: '提示', style: {color:'red',padding:'5px'},  
offset: new KMap.Pixel(0,10)})
```

参数:

名称	类型	描述
tip	Object	

getRotation()

描述:

-
-

获取逆时针倾斜度

-
-

返回:

类型:

`Array.<number>`

`getTip()`

描述:

-
-

获取鼠标提示

-
-

返回:

类型:

`*`

`getOptions()`

描述:

-
-

返回参数

-
-

返回:

类型: *

setCursor(cursor)

描述:

-
-

设置鼠标悬停样式

-
-

参数:

名称	类型	描述
cursor		

getCursor()

描述:

-
-

获取鼠标悬停样式

-
-

返回:

类型: *

setRotation(rotation)

描述:

-
-

设置逆时针旋转

-
-

示例

```
gltfSource.setRotation([1,2,3]);
```

参数:

名称	类型	描述
rotation	Array.<Number>	旋转弧度[x,y,z]

```
setPositionOffset(offset)
```

描述:

-
-

设置位置偏移

-
-

示例

```
gltfSource.setPositionOffset([1,2,3]);
```

参数:

名称	类型	描述
offset	Array.<Number>	单位米[x,y,x]

setRevolution(revolution)

描述:

-
-

设置公转

-
-

示例

```
gltfSource.setRevolution([1,2,3]);
```

参数:

名称	类型	描述
revolution	Array.<Number>	旋转弧度[x,y,z]

setAngle(angle)

描述:

-
-

设置水平方向角度

-
-

示例

```
labelMarker.setAngle(60);
```

参数:

名称	类型	描述
angle	Number	

getAngle()

描述:

-
-

获取水平角度

-
-

返回:

类型:

`number`

getId()

描述:

-
-

获取 ID

-
-

返回:

类型:

`*,string`

setId(id)

描述:

-
-

设置 ID

-
-

参数:

名称	类型	描述
id		

getzIndex()

描述:

-
-

获取层级

-
-

返回:

类型:

number,string

setzIndex(zIndex)

描述:

-
-

设置层级

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
zIndex		

```
hide()
```

描述:

-
-

隐藏

-
-

```
show()
```

描述:

-
-

显示

-
-

```
setZooms(zooms)
```

描述:

-
-

设置比例尺范围

-
-

参数:

名称	类型	描述
zooms		

getZooms()

描述:

-
-

获取比例尺范围

-
-

返回:

类型: `Array.<Number>`

getExtData()

描述:

-
-

获取用户自定义属性

-
-

返回:

类型: `*,null`

setExtData(extData)

描述:

-
-

设置用户自定义属性

-
-

参数:

名称	类型	描述
extData		

```
getStyle()
```

描述:

-
-

获取样式表达

-
-

```
setStyle(style)
```

描述:

-
-

设置样式表达

-
-

参数:

名称	类型	描述
style	PointStyle, PolylineStyle, PolygonStyle	

on(type,listener)

描述:

-
-

添加事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型:

BaseEvent

off(type,listener)

描述:

-
-

删除事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型:
BaseEvent

destroy()

描述:

- -
- 销毁自己

-
-

属性

bubble

描述:

- -
- 获取是否冒泡

-
-

draggable

描述:

-
-

获取是否可以拖拽

-
-

positionOffset

描述:

-
-

获取位置偏移

-
-

示例

```
gltfSource.positionOffset;
```

rotation

描述:

-
-

获取旋转

-
-

CircleMarker

new CircleMarker (options)

描述:

-
-

圆形标注

-
-

示例

```
let circlemarker = new KMap.CircleMarker({
  center : new KMap.Point(114.02308, 22.53474),
  radius : 100,
  height: 20,
  cursor: 'pointer',
  bubble : true,
  sizeAttenuation:false,
  followingCamera:false,
  style : {
    borderStyle: 'dashed',
    borderDasharray : [10,5],
    borderWidth:5,
    borderColor:"#000FFF7F",
    fillColor: "#FF00007F",
  }
})
map.add(circlemarker);
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
id	String	ID
zIndex	Number	覆盖物的叠加顺序
style	PolygonStyle	圆样式表达

名称	类型	描述
<code>cursor</code>	<code>String</code>	鼠标移入样式,参考 <code>css</code> 鼠标样式
<code>bubble</code>	<code>Bubble</code>	事件是否冒泡
<code>tip</code>	<code>optionsTipObject</code>	鼠标移入提示
<code>draggable</code>	<code>Boolean</code>	否可拖拽移动, 默认: <code>false</code> , <code>true</code> :可以拖拽, <code>false</code> :不可以拖拽
<code>visible</code>	<code>Boolean</code>	是否可见, <code>true</code> :显示, <code>false</code> :隐藏, 默认显示
<code>zooms</code>	<code>Array.<Number></code>	显示级别范围 <code>eq:[4,20]</code> , 数组[最小比例尺, 最大比例尺], <code>eq:[4,20]</code>
<code>extData</code>	<code>Any</code>	自定义属性, 用户自有属性, 会原样返回
<code>center</code>	<code>Point</code>	圆心坐标, <code>eq: new KMap.Point(114.11555554, 23.45662, 50)</code>
<code>radius</code>	<code>Number</code>	圆半径, 单位像素, 默认: <code>0</code> <code>eq:50</code>
<code>sizeAttenuation</code>	<code>Boolean</code>	是否近大远小

名称	类型	描述
<code>followingCamera</code>	<code>Boolean</code>	是否跟随摄像头, <code>true</code> : 随地图改变而改变相对屏幕倾斜角, <code>false</code> :永远面对屏幕, 不随地图旋转改变倾斜角度
<code>rotation:</code>		<code>[x:number,y:number,z:number]</code> 旋转角度, 弧度单位, eq: <code>[Math.PI / 5,Math.PI / 2,Math.PI / 3]</code>
<code>revolution:</code>		<code>[x:number,y:number,z:number]</code> 公转角度, 弧度单位 eq: <code>[Math.PI / 5,Math.PI / 2,Math.PI / 3]</code>
<code>positionOffset:</code>		<code>[x:number,y:number,z:number]</code> 位置偏移量, 单位米 eq: <code>[-200, 0, 0]</code>

optionsStyleObject :

名称	类型	描述
<code>borderColor</code>	<code>string</code>	边线颜色 默认: <code>#8aa8f5</code> , eq: <code>#FF00007F,rgba(0,255,55,0.5)</code>
<code>borderStyle</code>	<code>string</code>	边线样式 <code>solid,dashed</code> ,默认: <code>solid</code>
<code>borderDasharray</code>	<code>Array.<Number></code>	虚线段 eq: <code>[5,10]</code>
<code>borderWidth</code>	<code>Number</code>	边线宽度,默认: <code>3</code>

名称	类型	描述
<code>fillColor</code>	<code>String</code>	填充颜色 eq: #FF00007F, rgba(0,255,55,0.5) , 默认: #8af5bd
<code>images</code>	<code>Array.<String></code>	纹理 url 地址数组, eq: ['url', 'url1']
<code>repeatWidth</code>	<code>Number</code>	纹理平铺宽度。 -1: 拉伸, 0: 使用图片宽度平铺（默认），大于 0: 指定平铺宽度
<code>repeatHeight</code>	<code>Number</code>	纹理平铺高度。 -1: 拉伸（默认），0: 使用图片高度平铺，大于 0: 指定平铺高度

optionsTipObject :

名称	类型	描述
<code>text</code>	<code>String</code>	提示文字
<code>style</code>	<code>StyleSheet</code>	样式, 参考 css 样式

函数

setCenter(center,noUpdate)

描述:

-
-

设置圆心

-
-

参数:

名称	类型	描述
center		

```
setPosition(p)
```

描述:

-
-

设置坐标点

-
-

参数:

名称	类型	描述
p		

```
getCenter()
```

描述:

-
-

获取圆心

-
-

返回:

类型:
*

```
setRadius(radius,noUpdate)
```

描述:

-
-

设置半径 (px)

-
-

参数:

名称	类型	描述
----	----	----

radius

```
getRadius()
```

描述:

-
-

获取半径 (px)

-
-

返回:

类型:
*

setText(*text*)

描述:

-
-

设置文本

-
-

参数:

名称	类型	描述
text		

getText()

描述:

-
-

获取文本

-
-

返回:

类型: *

getCanvasWidth()

描述:

-
-

获取画布宽度

-
-

返回:

类型:

*

getFollowingCamera()

描述:

-
-

获取是否跟随摄像头，true：随地图改变而改变相对屏幕倾斜角，false:面朝屏幕，不随地图旋转改变倾斜角度

-
-

返回:

类型:

Boolean

update()

描述:

-
-

更新

-
-

getDepthStatus()

描述:

-
-

获取深度开关

-
-

返回:

类型:

boolean

```
setDepthStatus(value)
```

描述:

-
-

设置深度开关

-
-

参数:

名称	类型	描述
----	----	----

value	boolean	
-------	---------	--

```
setDraggable(b)
```

描述:

-
-

设置是否可以拖拽

-
-

参数:

名称	类型	描述
b		

```
getDraggable()
```

描述:

-
-

获取是否可以拖拽

-
-

返回:

类型:
boolean,*

```
setOptions(opts)
```

描述:

-
-

设置参数

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
opts		

```
setTip(tip)
```

描述:

- -
- 设置鼠标移入提示

-
-

示例

```
polyline.setTip({text: '提示', style: {color:'red',padding:'5px'},  
offset: new KMap.Pixel(0,10)})
```

参数:

名称	类型	描述
tip	Object	

```
getRotation()
```

描述:

- -
- 获取逆时针倾斜度

-
-

返回:

类型:
Array.<number>

getTip()

描述:

-
-

获取鼠标提示

-
-

返回:

类型: *

getOptions()

描述:

-
-

返回参数

-
-

返回:

类型: *

setCursor(cursor)

描述:

-
-

设置鼠标悬停样式

-
-

参数:

名称	类型	描述
cursor		

```
getCursor()
```

描述:

-
-

获取鼠标悬停样式

-
-

返回:

类型: *

```
setRotation(rotation)
```

描述:

-
-

设置逆时针旋转

-
-

示例

```
gltfSource.setRotation([1,2,3]);
```

参数:

名称	类型	描述
rotation	Array.<Number>	旋转弧度[x,y,z]

setPositionOffset(*offset*)

描述:

-
-

设置位置偏移

-
-

示例

```
gltfSource.setPositionOffset([1,2,3]);
```

参数:

名称	类型	描述
offset	Array.<Number>	单位米[x,y,x]

setRevolution(*revolution*)

描述:

-
-

设置公转

-
-

示例

```
gltfSource.setRevolution([1,2,3]);
```

参数:

名称	类型	描述
revolution	Array.<Number>	旋转弧度[x,y,z]

setAngle(*angle*)

描述:

-
-

设置水平方向角度

-
-

示例

```
labelMarker.setAngle(60);
```

参数:

名称	类型	描述
angle	Number	

getAngle()

描述:

-
-

获取水平角度

-
-

返回:

类型：
number

getId()

描述:

-
-

获取 ID

-
-

返回:

类型：
*,string

setId(*id*)

描述:

-
-

设置 ID

-
-

参数:

名称	类型	描述
id		

getzIndex()

描述:

-
-

获取层级

-
-

返回:

类型:
number,string

```
setzIndex(zIndex)
```

描述:

-
-

设置层级

-
-

参数:

名称	类型	描述
zIndex		

```
hide()
```

描述:

-
-

隐藏

-
-

show()

描述:

-
-

显示

-
-

setZooms(zooms)

描述:

-
-

设置比例尺范围

-
-

参数:

名称	类型	描述
----	----	----

zooms		
-------	--	--

getZooms()

描述:

-
-

获取比例尺范围

-
-

返回:

类型:
`Array.<Number>`

`getExtData()`

描述:

-
-

获取用户自定义属性

-
-

返回:

类型:
`*,null`

`setExtData(extData)`

描述:

-
-

设置用户自定义属性

-
-

参数:

名称	类型	描述
----	----	----

<code>extData</code>		
----------------------	--	--

`getStyle()`

描述:

-
-

获取样式表达

-
-

```
setStyle(style)
```

描述:

-
-

设置样式表达

-
-

参数:

名称	类型	描述
----	----	----

<code>style</code>	<code>PointStyle, PolylineStyle, PolygonStyle</code>	
--------------------	--	--

```
on(type,listener)
```

描述:

-
-

添加事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型:
BaseEvent

```
off(type,listener)
```

描述:

-
-

删除事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型:
BaseEvent

```
destroy()
```

描述:

-
-

销毁自己

-
-

属性

bubble

描述:

-
-

获取是否冒泡

-
-

draggable

描述:

-
-

获取是否可以拖拽

-
-

positionOffset

描述:

-
-

获取位置偏移

-
-

示例

```
gltfSource.positionOffset;
```

rotation

描述:

-
-

获取旋转

-
-

Ellipse

new Ellipse (options)

描述:

-
-

椭圆类

-
-

示例

```
let ellipse = new KMap.Ellipse({
  center : new KMap.Point(114.02308, 22.53474),
  radius : [10,20],
  height: 20,
  cursor: 'pointer',
  bubble : true,
  style : [{
    borderStyle: 'dashed',
    borderDasharray : [10,5],
    borderWidth:5,
    borderColor:"#000FFF7F",
    fillColor: "#FF00007F",
  },{
    fillColor: "rgba(0,255,55,0.5)",
  }
])
```

```
    ]  
  })  
  map.add(ellipse);
```

参数:

名称	类型	描述
<code>options</code>	<code>optionsObject</code>	

optionsObject :

名称	类型	描述
<code>id</code>	<code>String</code>	ID
<code>zIndex</code>	<code>Number</code>	覆盖物的叠加顺序
<code>style</code>	<code>Array.<PolygonStyle></code>	表达数组，0 是顶面样式，其他为侧面样式 [<code>{fillColor:"#FF00007F",borderWidth:2,bo rderColor:"#000FFF7F"}},{fillColor: "rgba(0,255,55,0.5)"}</code>]
<code>height</code>	<code>Number</code>	立面高度，米 eq:10
<code>cursor</code>	<code>String</code>	鼠标移入样式,参考 <code>css</code> 鼠标样式
<code>bubble</code>	<code>Bubble</code>	事件是否冒泡
<code>tip</code>	<code>optionsTip Object</code>	鼠标移入提示
<code>draggable</code>	<code>Boolean</code>	可否拖拽移动，默认: <code>false</code> , <code>true</code> :可以拖拽, <code>false</code> :不可以拖拽

名称	类型	描述
visible	Boolean	是否可见, true :显示, false :隐藏, 默认显示
zooms	Array.<Number>	显示级别范围, 数组[最小比例尺, 最大比例尺], eq:[4,20]
extData	Any	自定义属性, 用户自有属性, 会原样返回
center	Point	圆心坐标, eq: new KMap.Point(114.11555554,23.45662,50)
radius	Array.<Number>	圆心半径, 米,默认: 0 eq:50
rotation:		[x:number,y:number:z:number] 旋转角度, 弧度单位, eq: [Math.PI / 5,Math.PI / 2,Math.PI / 3]
revolution:		[x:number,y:number:z:number] 公转角度, 弧度单位 eq: [Math.PI / 5,Math.PI / 2,Math.PI / 3]
position Offset:		[x:number,y:number:z:number] 位置偏移量, 单位米 eq: [-200, 0, 0]

optionsStyleObject :

名称	类型	描述
----	----	----

名称	类型	描述
<code>borderColor</code>	<code>string</code>	边线颜色 默认: <code>#8aa8f5</code> , eq: <code>#FF00007F,rgba(0,255,55,0.5)</code>
<code>borderStyle</code>	<code>string</code>	边线样式 <code>solid,dashed</code> ,默认: <code>solid</code>
<code>borderDasharray</code>	<code>Array.<Number></code>	虚线段 eq: <code>[5,10]</code>
<code>borderWidth</code>	<code>Number</code>	边线宽度,默认: <code>3</code>
<code>fillColor</code>	<code>String</code>	填充颜色 eq: <code>#FF00007F,rgba(0,255,55,0.5)</code> ,默认: <code>#8af5bd</code>

optionsTipObject :

名称	类型	描述
<code>text</code>	<code>String</code>	提示文字
<code>style</code>	<code>StyleSheet</code>	样式, 参考 <code>css</code> 样式

函数

`setCenter(center,noUpdate)`

描述:

-
-

设置圆心

-
-

示例

```
ellipse.setCenter(new KMap.Point(114.02308, 22.53474));
```

参数:

名称	类型	描述
center	Point	

```
getCenter()
```

描述:

-
-

获取圆心

-
-

示例

```
ellipse.getCenter();
```

返回:

类型: Point

```
setRadius(radius,noUpdate)
```

描述:

-
-

设置半径模向半径, 纵向半径

-
-

示例

```
ellipse.setRadius([100,50]);
```

参数:

名称	类型	描述
radius	Array. <Number>	

getRadius()

描述:

-
-

获取半径模向半径, 纵向半径

-
-

示例

```
ellipse.getRadius();
```

返回:

类型:
Number

getHeight()

描述:

-
-

获取立面高度

-
-

示例

```
ellipse.getHeight();
```

返回:

类型:
Number

```
setHeight(height)
```

描述:

-
-

设置立面高度

-
-

示例

```
ellipse.setHeight(20);
```

参数:

名称	类型	描述
height	Number	立面高度

```
contains(point)
```

描述:

-
-

判断点是否在椭圆内

-
-

示例

```
ellipse.contains(new KMap.Point(114.02308, 22.53474));
```

参数:

名称	类型	描述
point	Point, Array.<Number>	经纬度

返回:

类型:
boolean

```
getArea()
```

描述:

-
-

获取椭圆的面积 $S = \pi \times r_x \times r_y$

-
-

示例

```
ellipse.getArea();
```

返回:

类型:
number

```
getBounds()
```

描述:

-
-

获取外框

-

-

示例

```
ellipse.getBounds();
```

返回:

类型:

Bounds

getScopes()

描述:

-
-

获取切线

-
-

示例

```
ellipse.getScopes();
```

返回:

类型:

Array.<Array.<Number>>

getDepthStatus()

描述:

-
-

获取深度开关

-
-

返回:

类型:

boolean

setDepthStatus(value)

描述:

-
-

设置深度开关

-
-

参数:

名称	类型	描述
value	boolean	

setDraggable(b)

描述:

-
-

设置是否可以拖拽

-
-

参数:

名称	类型	描述
b		

getDraggable()

描述:

-
-

获取是否可以拖拽

-
-

返回:

类型:
boolean,*

```
setOptions(opts)
```

描述:

-
-

设置参数

-
-

参数:

名称	类型	描述
opts		

```
setTip(tip)
```

描述:

-
-

设置鼠标移入提示

-
-

示例

```
polyline.setTip({text: '提示', style: {color:'red',padding:'5px'},  
offset: new KMap.Pixel(0,10)})
```

参数:

名称	类型	描述
tip	Object	

```
getRotation()
```

描述:

-
-

获取逆时针倾斜度

-
-

返回:

类型:
Array.<number>

```
getTip()
```

描述:

-
-

获取鼠标提示

-
-

返回:

类型:
*

getOptions()

描述:

-
-

返回参数

-
-

返回:

类型: *

setCursor(cursor)

描述:

-
-

设置鼠标悬停样式

-
-

参数:

名称	类型	描述
----	----	----

cursor		
--------	--	--

getCursor()

描述:

-
-

获取鼠标悬停样式

-
-

返回:

类型:
*

setRotation(rotation)

描述:

-
-

设置逆时针旋转

-
-

示例

gltfSource.setRotation([1,2,3]);

参数:

名称	类型	描述
rotation	Array.<Number>	旋转弧度[x,y,z]

setPositionOffset(offset)

描述:

-
-

设置位置偏移

-

-

示例

```
gltfSource.setPositionOffset([1,2,3]);
```

参数:

名称	类型	描述
offset	Array.<Number>	单位米[x,y,x]

```
setRevolution(revolution)
```

描述:

-
-

设置公转

-
-

示例

```
gltfSource.setRevolution([1,2,3]);
```

参数:

名称	类型	描述
revolution	Array.<Number>	旋转弧度[x,y,z]

```
setAngle(angle)
```

描述:

-
-

设置水平方向角度

-
-

示例

```
labelMarker.setAngle(60);
```

参数:

名称	类型	描述
angle	Number	

getAngle()

描述:

-
-

获取水平角度

-
-

返回:

类型:
number

getId()

描述:

-
-

获取 ID

-
-

返回:

类型：
*,string

setId(id)

描述:

-
-

设置 ID

-
-

参数:

名称	类型	描述
id		

getzIndex()

描述:

-
-

获取层级

-
-

返回:

类型：
number,string

setzIndex(zIndex)

描述:

-
-

设置层级

-
-

参数:

名称	类型	描述
zIndex		

```
hide()
```

描述:

-
-

隐藏

-
-

```
show()
```

描述:

-
-

显示

-
-

```
setZooms(zooms)
```

描述:

-
-

设置比例尺范围

-
-

参数:

名称	类型	描述
zooms		

```
getZooms()
```

描述:

-
-

获取比例尺范围

-
-

返回:

类型:
Array.<Number>

```
getExtData()
```

描述:

-
-

获取用户自定义属性

-
-

返回:

类型:

**,null*

setExtData(*extData*)

描述:

-
-

设置用户自定义属性

-
-

参数:

名称

类型

描述

extData

getStyle()

描述:

-
-

获取样式表达

-
-

setStyle(*style*)

描述:

-
-

设置样式表达

-
-

参数:

名称	类型	描述
style	PointStyle, PolylineStyle, PolygonStyle	

on(type,listener)

描述:

-
-

添加事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型: BaseEvent

off(type,listener)

描述:

-
-

删除事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型:
BaseEvent

```
destroy()
```

描述:

-
-

销毁自己

-
-

属性

```
bubble
```

描述:

-
-

获取是否冒泡

-

-

draggable

描述:

-
-

获取是否可以拖拽

-
-

positionOffset

描述:

-
-

获取位置偏移

-
-

示例

```
gltfSource.positionOffset;
```

rotation

描述:

-
-

获取旋转

-
-

Feature

函数

getDepthStatus()

描述:

-
-

获取深度开关

-
-

返回:

类型:
boolean

setDepthStatus(value)

描述:

-
-

设置深度开关

-
-

参数:

名称	类型	描述
----	----	----

value	boolean	
-------	---------	--

setDraggable(b)

描述:

-
-

设置是否可以拖拽

-
-

参数:

名称	类型	描述
b		

```
getDraggable()
```

描述:

-
-

获取是否可以拖拽

-
-

返回:

类型:
boolean,*

```
setOptions(opts)
```

描述:

-
-

设置参数

-
-

参数:

名称	类型	描述
opts		

setTip(*tip*)

描述:

-
-

设置鼠标移入提示

-
-

示例

```
polyline.setTip({text: '提示', style: {color:'red',padding:'5px'},  
offset: new KMap.Pixel(0,10)})
```

参数:

名称	类型	描述
tip	Object	

getRotation()

描述:

-
-

获取逆时针倾斜度

-
-

返回:

类型：
Array.<number>

getTip()

描述：

-
-

获取鼠标提示

-
-

返回：

类型：

*

getOptions()

描述：

-
-

返回参数

-
-

返回：

类型：

*

setCursor(cursor)

描述：

-
-

设置鼠标悬停样式

-
-

参数:

名称	类型	描述
cursor		

getCursor()

描述:

-
-

获取鼠标悬停样式

-
-

返回:

类型:
*

setRotation(rotation)

描述:

-
-

设置逆时针旋转

-
-

示例

gltfSource.setRotation([1,2,3]);

参数:

名称	类型	描述
rotation	Array.<Number>	旋转弧度[x,y,z]

setPositionOffset(offset)

描述:

- -
- 设置位置偏移

-
-

示例

```
gltfSource.setPositionOffset([1,2,3]);
```

参数:

名称	类型	描述
offset	Array.<Number>	单位米[x,y,x]

setRevolution(revolution)

描述:

- -
- 设置公转

-
-

示例

```
gltfSource.setRevolution([1,2,3]);
```

参数:

名称	类型	描述
revolution	Array.<Number>	旋转弧度[x,y,z]

setAngle(*angle*)

描述:

-
-

设置水平方向角度

-
-

示例

```
labelMarker.setAngle(60);
```

参数:

名称	类型	描述
angle	Number	

getAngle()

描述:

-
-

获取水平角度

-
-

返回:

类型:

number

getId()

描述:

-
-

获取 ID

-
-

返回:

类型:

*,string

setId(id)

描述:

-
-

设置 ID

-
-

参数:

名称

类型

描述

id

getzIndex()

描述:

-
-

获取层级

-
-

返回:

类型:
number, string

```
setzIndex(zIndex)
```

描述:

-
-

设置层级

-
-

参数:

名称	类型	描述
----	----	----

zIndex

```
hide()
```

描述:

-
-

隐藏

-
-

```
show()
```

描述:

-
-

显示

-
-

```
setZooms(zooms)
```

描述:

-
-

设置比例尺范围

-
-

参数:

名称	类型	描述
zooms		

```
getZooms()
```

描述:

-
-

获取比例尺范围

-
-

返回:

类型:
`Array.<Number>`

```
getExtData()
```

描述:

-
-

获取用户自定义属性

-
-

返回:

类型:
`*,null`

```
setExtData(extData)
```

描述:

-
-

设置用户自定义属性

-
-

参数:

名称	类型	描述
----	----	----

extData		
---------	--	--

getStyle()

描述:

-
-

获取样式表达

-
-

setStyle(*style*)

描述:

-
-

设置样式表达

-
-

参数:

名称	类型	描述
----	----	----

<code>style</code>	<code>PointStyle, PolylineStyle, PolygonStyle</code>	
--------------------	--	--

on(*type*,*listener*)

描述:

-
-

添加事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型:
BaseEvent

```
off(type,listener)
```

描述:

-
-
- 删除事件
-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型:
BaseEvent

```
destroy()
```

描述:

-
-

销毁自己

-
-

属性

bubble

描述:

-
-

获取是否冒泡

-
-

draggable

描述:

-
-

获取是否可以拖拽

-
-

positionOffset

描述:

-
-

获取位置偏移

-
-

示例

```
gltfSource.positionOffset;
```

rotation

描述:

-
-

获取旋转

-
-

GeoJSON

new GeoJSON (options)

描述:

-
-

展现 geojson 数据对象到地图显示

-
-

参数:

名称	类型	描述
options	optionsObject	属性

optionsObject :

名称	类型	描述
----	----	----

名称	类型	描述
geoJson	Object	geojson 数据
onLoad	Function	加载时回调函数
pointStyle	Array.<PointStyle>	点样式表达数组
polylineStyle	Array.<PolylineStyle>	线样式表达数组
polygonStyle	Array.<PolygonStyle>	多边形样式表达数组

函数

```
setGeoJson(geoJson)
```

描述:

-
-

设置 geojson 数据

-
-

参数:

名称	类型	描述
geoJson	object	数据

getGeoJson()

描述:

-
-

获取 geojson 数据

-
-

返回:

类型: Object

getAllStyle(*type*)

描述:

-
-

获取所有样式

-
-

参数:

名称	类型	描述
----	----	----

type		
------	--	--

返回:

类型: Object

setPointStyle(*style, zoom*)

描述:

-
-

设置点样式

-
-

参数:

名称	类型	描述
style	PointStyle	点样式表达
zoom	number	比例尺级别，设置对应比例尺的样式 0:默认样式 4-20 是比例尺样式

getPointSize(zoom)

描述:

-
-

获取点样式

-
-

参数:

名称	类型	描述
zoom		比例尺级别，对应比例尺的样式

返回:

类型:

PointStyle

```
setPolylineStyle(style, zoom)
```

描述:

-
-

设置线样式

-
-

参数:

名称	类型	描述
style	PolylineStyle	线样式表达
zoom	number	比例尺级别，设置对应比例尺的样式 0:默认样式 4-20 是比例尺样式

```
getPolylineStyle(zoom)
```

描述:

-
-

获取线样式

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
zoom		比例尺级别，对应比例尺的样式

返回:

类型:
PolylineStyle

```
setPolygonStyle(style, zoom)
```

描述:

- -
- 设置多边形样式

-
-

参数:

名称	类型	描述
style	PolygonStyle	多边形样式表达
zoom	number	比例尺级别，设置对应比例尺的样式 0:默认样式 4-20 是比例尺样式

```
getPolygonStyle(zoom)
```

描述:

- -
- 获取多边形样式

-
-

参数:

名称	类型	描述
zoom		比例尺级别，对应比例尺的样式

返回:

类型:
PolygonStyle

```
update()
```

描述:

-
-

更新

-
-

Gltf

```
new Gltf (options)
```

描述:

-
-

GLTF 模型

-
-

示例

```
let gltf = new KMap.Gltf({
  url: './models/gltf/RobotExpressive.glb',
  scale: 3,
  position: new KMap.Point(114.0278946, 22.6124434, 0),
  rotation: [x:number,y:number:z:number],
  color: 'rgba(255, 128, 0)',
  positionOffset:[x:number,y:number:z:number],
  rotationOffset:[x:number,y:number:z:number],
  opacity: 1
});
map.add(gltf);
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
scale	Number	缩放比例
position	Point	经纬度坐标
rotation:		[x:number,y:number:z:number] 旋转角度，弧度单位，eq: [Math.PI / 5,Math.PI / 2,Math.PI / 3]
revolution:		[x:number,y:number:z:number] 公转角度，弧度单位 eq: [Math.PI / 5,Math.PI / 2,Math.PI / 3]
positionOffset:		[x:number,y:number:z:number]

名称	类型	描述
		位置偏移量，单位米 eq: [-200, 0, 0]
color	String	颜色 eq:rgba(255, 128, 0), #FF00007F
opacity	Number	透明度，默认：1，最大1不透明，最小0完全透明，eq: 0.5: 半透明
visible	Boolean	显显示隐藏，默认：true,true:显示，false:隐藏
zooms	Array.<Number>	显示级别范围，数组[最小比例尺，最大比例尺]，eq:[4,20]
draggable	Boolean	否可拖拽移动，默认：false, true:可以拖拽，false:不可以拖拽

函数

setMaterial(options)

描述:

-
-

设置材质

-
-

示例

```
gltfSource.setMaterial({
  body: {
    color: 0xffddcc, metalness: 1.0, roughness: 0.8,
    clearcoat: 1.0, clearcoatRoughness: 0.2, transparent: 1,
  }
});
```

参数:

名称	类型	描述
props	Object	

setAngle(*angle*)

描述:

-
-

设置角度

-
-

示例

```
gltfSource.setAngle(90)
```

参数:

名称	类型	描述
angle	Number	正北顺时针角度，0-360

setRotationOffset(*rotationOffset*)

描述:

-
-

设置旋转修正

-
-

示例

```
gltfSource.setRotationOffset([1,1,1]);
```

参数:

名称	类型	描述
rotation	Array.<Number>	修正旋转弧度[x,y,z]

setScale(scale)

描述:

-
-

设置缩放

-
-

示例

```
gltfSource.setRotation(10);
```

参数:

名称	类型	描述
scale	Number	缩放比例

setColor(color)

描述:

-
-

设置颜色

-
-

示例

```
gltfSource.setColor('rgba(255, 128, 0)');
```

参数:

名称	类型	描述
color	String	格式 rgba(255, 128, 0)

```
setOpacity(opacity)
```

描述:

-
-

设置透明度

-
-

示例

```
gltfSource.setOpacity(0.5);
```

参数:

名称	类型	描述
opacity	Number	透明度, 0-1, 0 为全透明, 1 为不透明, 默认 1

setVisible(visible)

描述:

-
-

设置显示隐藏

-
-

示例

```
gltfSource.visible(true);
```

参数:

名称	类型	描述
visible	Boolean	是否

setPosition(position)

描述:

-
-

设置位置

-
-

示例

```
gltfSource.setPosition(new KMap.Point(114.0278946, 22.6124434, 0));
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
<code>position</code>	<code>Point</code>	坐标点

`getPosition()`

描述:

-
-

获取位置

-
-

返回:

坐标点

类型:

`Point`

`setZooms(zooms)`

描述:

-
-

设置显示比例尺级别

-
-

示例

```
gltfSource.setZooms([12,21]);
```

参数:

名称	类型	描述
zooms	Number	比例尺级别，范围 4-21

`play(animationName)`

描述:

-
-

播放动画

-
-

示例

```
gltfSource.play("Walking");
```

参数:

名称	类型	描述
animationName	string	动画名

`stop()`

描述:

-
-

停止动画

-
-

示例

```
gltfSource.stop();
```

getDepthStatus()

描述:

-
-

获取深度开关

-
-

返回:

类型:

boolean

setDepthStatus(value)

描述:

-
-

设置深度开关

-
-

参数:

名称	类型	描述
----	----	----

value	boolean	
-------	---------	--

setDraggable(b)

描述:

-
-

设置是否可以拖拽

-
-

参数:

名称	类型	描述
b		

```
getDraggable()
```

描述:

-
-

获取是否可以拖拽

-
-

返回:

类型:
boolean,*

```
setTip(tip)
```

描述:

-
-

设置鼠标移入提示

-
-

示例

```
polyline.setTip({text: '提示', style: {color:'red',padding:'5px'},
offset: new KMap.Pixel(0,10)})
```

参数:

名称	类型	描述
tip	Object	

getRotation()

描述:

- -
- 获取逆时针倾斜度

-
-

返回:

类型:
Array.<number>

getTip()

描述:

- -
- 获取鼠标提示

-
-

返回:

类型:
*

getOptions()

描述:

-
-

返回参数

-
-

返回:

类型:
*

setCursor(cursor)

描述:

-
-

设置鼠标悬停样式

-
-

参数:

名称

类型

描述

cursor

getCursor()

描述:

-
-

获取鼠标悬停样式

-
-

返回:

类型:
*

setRotation(rotation)

描述:

-
-

设置逆时针旋转

-
-

示例

```
gltfSource.setRotation([1,2,3]);
```

参数:

名称	类型	描述
rotation	Array.<Number>	旋转弧度[x,y,z]

setPositionOffset(offset)

描述:

-
-

设置位置偏移

-

-

示例

```
gltfSource.setPositionOffset([1,2,3]);
```

参数:

名称	类型	描述
offset	Array.<Number>	单位米[x,y,x]

```
setRevolution(revolution)
```

描述:

-
-

设置公转

-
-

示例

```
gltfSource.setRevolution([1,2,3]);
```

参数:

名称	类型	描述
revolution	Array.<Number>	旋转弧度[x,y,z]

```
getAngle()
```

描述:

-
-

获取水平角度

-
-

返回:

类型:
number

```
getId()
```

描述:

-
-

获取 ID

-
-

返回:

类型:
*,string

```
setId(id)
```

描述:

-
-

设置 ID

-
-

参数:

名称	类型	描述
id		

getzIndex()

描述:

-
-

获取层级

-
-

返回:

类型:

number,string

setzIndex(zIndex)

描述:

-
-

设置层级

-
-

参数:

名称

类型

描述

zIndex

hide()

描述:

-
-

隐藏

-
-

```
show()
```

描述:

-
-

显示

-
-

```
getZooms()
```

描述:

-
-

获取比例尺范围

-
-

返回:

类型: `Array.<Number>`

```
getExtData()
```

描述:

-
-

获取用户自定义属性

-
-

返回:

类型:
`*,null`

```
setExtData(extData)
```

描述:

-
-

设置用户自定义属性

-
-

参数:

名称	类型	描述
extData		

```
getStyle()
```

描述:

-
-

获取样式表达

-
-

```
setStyle(style)
```

描述:

-
-

设置样式表达

-
-

参数:

名称	类型	描述
style	PointStyle, PolylineStyle, PolygonStyle	

```
on(type,listener)
```

描述:

-
-

添加事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型: BaseEvent

```
off(type,listener)
```

描述:

-
-

删除事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型:
BaseEvent

```
destroy()
```

描述:

-
-

销毁自己

-
-

```
setOptions(options)
```

描述:

-
-

设置参数

-
-

参数:

名称	类型	描述
opts		

属性

url

描述:

-
-

模型 URL

-
-

示例

gltfSource.url;

material

描述:

-
-

材质

-
-

示例

gltfSource.material;

rotationOffset

描述:

-
-

旋转弧度修正

-
-

示例

```
gltfSource.rotation;
```

scale

描述:

-
-

获取缩放比例

-
-

示例

```
gltfSource.scale;
```

color

描述:

-
-

获取颜色

-
-

示例

```
gltfSource.color;
```

opacity

描述:

-
-

获取透明度

-
-

示例

```
gltfSource.opacity;
```

position

描述:

-
-

获取位置

-
-

示例

```
gltfSource.position;
```

scene

bubble

描述:

-
-

获取是否冒泡

-
-

draggable

描述:

-
-

获取是否可以拖拽

-
-

positionOffset

描述:

-
-

获取位置偏移

-
-

示例

```
gltfSource.positionOffset;
```

rotation

描述:

-
-

获取旋转

-
-

事件

```
moving()
```

描述:

- -
- 覆盖物在执行 moveTo, moveAlong 动画时触发事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
moveend()
```

描述:

- -
- 覆盖物执行 moveTo 动画结束时触发事件

-
-

参数:

名称	类型	描述
e	Object	结果

moveAlong()

描述:

-
-

覆盖物执行 moveAlong 动画一次后触发事件

-
-

参数:

名称	类型	描述
e	Object	结果

LabelMarker

new LabelMarker (options)

描述:

-
-

矢量标注类

-
-

示例

```
let labelmarker = new KMap.LabelMarker({
  position : new KMap.Point(114.02308, 22.53474),
  cursor: 'pointer',
  style : {
    width:23,
    height:29,
    borderWidth:1,
    borderColor:"rgba(0,13,255,0.5)",
```

```

        fillColor: "rgba(255,0,0,0.5)",
        posStyle: 0,
        textSize:10,
        textColor:'rgba(255, 0, 0, 1)',
        images:['./images/poiMarker.png']
    },
    text:'你好凯立德',
    zIndex:100,
    sizeAttenuation: false,//是否近大远小
    followingCamera: true,//是否跟随（地图）摄像头
    rotation:[0,0,0]
  })
  map.add(labelmarker);

```

参数:

名称	类型	描述
----	----	----

options	optionsObject
---------	---------------

optionsObject :

名称	类型	描述
----	----	----

id	String	ID
----	--------	----

zIndex	Number	覆盖物的叠加顺序
--------	--------	----------

style	PointStyle	圆样式表达
-------	------------	-------

position	Point	坐标
----------	-------	----

cursor	String	鼠标移入样式,参考 css 鼠标样式
--------	--------	--------------------

bubble	Bubble	事件是否冒泡
--------	--------	--------

tip	optionsTipObject	鼠标移入提示
-----	------------------	--------

名称	类型	描述
<code>draggable</code>	<code>Boolean</code>	否可拖拽移动，默认： <code>false</code> ， <code>true</code> :可以拖拽， <code>false</code> :不可以拖拽
<code>visible</code>	<code>Boolean</code>	是否可见， <code>true</code> :显示， <code>false</code> :隐藏，默认显示
<code>zooms</code>	<code>Array.<Number></code>	显示级别范围 eq:[4,20]
<code>extData</code>	<code>Any</code>	自定义属性，用户自有属性，会原样返回
<code>rotation:</code>		[x:number,y:number:z:number] 旋转角度，弧度单位，eq: [Math.PI / 5,Math.PI / 2,Math.PI / 3]
<code>revolution:</code>		[x:number,y:number:z:number] 公转角度，弧度单位 eq: [Math.PI / 5,Math.PI / 2,Math.PI / 3]
<code>positionOffset:</code>		[x:number,y:number:z:number] 位置偏移量，单位米 eq: [-200, 0, 0]

optionsStyleObject :

名称	类型	描述
<code>textColor</code>	<code>String</code>	文本颜色 eq: #48a8fc
<code>textSize</code>	<code>Number</code>	文本大小（单位：磅）

optionsTipObject :

名称	类型	描述
<code>text</code>	<code>String</code>	提示文字
<code>style</code>	<code>StyleSheet</code>	样式, 参考 <code>css</code> 样式

函数

`setPosition(point,noUpdate)`

描述:

-
-

设置坐标点

-
-

参数:

名称	类型	描述
<code>point</code>	<code>Point</code>	坐标点

getPosition()

描述:

-
-

获取坐标点

-
-

返回:

类型:

Point

setText(text,noUpdate)

描述:

-
-

设置文本信息

-
-

参数:

名称	类型	描述
text	String	文本信息

getText()

描述:

-

-

获取文本信息

-

-

参数:

名称	类型	描述
文本信息	String	

update()

描述:

-

-

更新

-

-

getDepthStatus()

描述:

-

-

获取深度开关

-

-

返回:

类型:

boolean

setDepthStatus(value)

描述:

-
-

设置深度开关

-
-

参数:

名称	类型	描述
value	boolean	

setDraggable(b)

描述:

-
-

设置是否可以拖拽

-
-

参数:

名称	类型	描述
b		

getDraggable()

描述:

-

-

获取是否可以拖拽

-
-

返回:

类型:
`boolean,*`

```
setOptions(opts)
```

描述:

-
-

设置参数

-
-

参数:

名称	类型	描述
opts		

```
setTip(tip)
```

描述:

-
-

设置鼠标移入提示

-
-

示例

```
polyline.setTip({text: '提示', style: {color:'red',padding:'5px'},
offset: new KMap.Pixel(0,10)})
```

参数:

名称	类型	描述
tip	Object	

```
getRotation()
```

描述:

-
-

获取逆时针倾斜度

-
-

返回:

类型: `Array.<number>`

```
getTip()
```

描述:

-
-

获取鼠标提示

-
-

返回:

类型: `*`

getOptions()

描述:

-
-

返回参数

-
-

返回:

类型: *

setCursor(cursor)

描述:

-
-

设置鼠标悬停样式

-
-

参数:

名称	类型	描述
----	----	----

cursor		
--------	--	--

getCursor()

描述:

-
-

获取鼠标悬停样式

-
-

返回:

类型:
*

setRotation(rotation)

描述:

-
-

设置逆时针旋转

-
-

示例

gltfSource.setRotation([1,2,3]);

参数:

名称	类型	描述
rotation	Array.<Number>	旋转弧度[x,y,z]

setPositionOffset(offset)

描述:

-
-

设置位置偏移

-

-

示例

```
gltfSource.setPositionOffset([1,2,3]);
```

参数:

名称	类型	描述
offset	Array.<Number>	单位米[x,y,x]

```
setRevolution(revolution)
```

描述:

-
-

设置公转

-
-

示例

```
gltfSource.setRevolution([1,2,3]);
```

参数:

名称	类型	描述
revolution	Array.<Number>	旋转弧度[x,y,z]

```
setAngle(angle)
```

描述:

-
-

设置水平方向角度

-
-

示例

```
labelMarker.setAngle(60);
```

参数:

名称	类型	描述
angle	Number	

```
getAngle()
```

描述:

-
-

获取水平角度

-
-

返回:

类型:
number

```
getId()
```

描述:

-
-

获取 ID

-
-

返回:

类型：
*,string

setId(id)

描述:

-
-

设置 ID

-
-

参数:

名称	类型	描述
id		

getzIndex()

描述:

-
-

获取层级

-
-

返回:

类型：
number,string

setzIndex(zIndex)

描述:

-
-

设置层级

-
-

参数:

名称	类型	描述
zIndex		

```
hide()
```

描述:

-
-

隐藏

-
-

```
show()
```

描述:

-
-

显示

-
-

```
setZooms(zooms)
```

描述:

-
-

设置比例尺范围

-
-

参数:

名称	类型	描述
zooms		

```
getZooms()
```

描述:

-
-

获取比例尺范围

-
-

返回:

类型:
Array.<Number>

```
getExtData()
```

描述:

-
-

获取用户自定义属性

-
-

返回:

类型:

**,null*

setExtData(*extData*)

描述:

-
-

设置用户自定义属性

-
-

参数:

名称

类型

描述

extData

getStyle()

描述:

-
-

获取样式表达

-
-

setStyle(*style*)

描述:

-
-

设置样式表达

-
-

参数:

名称	类型	描述
style	PointStyle, PolylineStyle, PolygonStyle	

on(type,listener)

描述:

-
-

添加事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型: BaseEvent

off(type,listener)

描述:

-
-

删除事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型:
BaseEvent

```
destroy()
```

描述:

-
-

销毁自己

-
-

属性

```
bubble
```

描述:

-
-

获取是否冒泡

-

-

draggable

描述:

-
-

获取是否可以拖拽

-
-

positionOffset

描述:

-
-

获取位置偏移

-
-

示例

```
gltfSource.positionOffset;
```

rotation

描述:

-
-

获取旋转

-
-

事件

moving()

描述:

-
-

覆盖物在执行 moveTo, moveAlong 动画时触发事件

-
-

参数:

名称	类型	描述
e	Object	结果

moveend()

描述:

-
-

覆盖物执行 moveTo 动画结束时触发事件

-
-

参数:

名称	类型	描述
e	Object	结果

movealong()

描述:

-
-

覆盖物执行 moveAlong 动画一次后触发事件

-
-

参数:

名称	类型	描述
e	Object	结果

MassMarks

new MassMarks (data,options)

描述:

-
-

海量点类

-
-

示例

```
let data = [{"point":[117.190182,39.125596],"name":"天津市",
,"style":0},{ "point":[116.405285,39.904989],"name":"北京市","style":0}];
let style = [{
  images: './images/mass0.png',
  width:12,
  height:12
}, {
  images: './images/mass1.png',
  width:10,
  height:10
}, {
  images: './images/mass2.png',
```

```
        width:8,
        height:8
    }]);
let mass = new KMap.MassMarks(data,{
    opacity: 0.8,
    style : style,
    coordinates:'GCJ02'
});
map.add(mass);
```

参数:

名称	类型	描述
<code>data</code>	<code>Array.<Object></code>	
<code>options</code>	<code>optionsObject</code>	

data[]Object :

名称	类型	描述
<code>point</code>	<code>Array.<Number></code>	数据点坐标数组，格式[lon,lat]
<code>name</code>	<code>String</code>	数据点名称
<code>style</code>	<code>Number</code>	数据点显示样式索引

optionsObject :

名称	类型	描述
<code>opacity</code>	<code>Number</code>	透明度，0-1，0 为全透明，1 为不透明，默认 1
<code>style</code>	<code>Array.<PointStyle></code>	点样式数组，数组中用 <code>style:0</code> 指定显示样式

名称	类型	描述
coordinates	String	坐标系，支持 WGS84,GCJ02，默认 GCJ02

函数

```
setData(data)
```

描述:

-
-

设置数据

-
-

示例

```
mass.setData([
  {"point": [117.190182, 39.125596], "name": "天津市", "style": 0},
  {"point": [116.405285, 39.904989], "name": "北京市", "style": 0}
]);
```

参数:

名称	类型	描述
data	Array.<Object>	

data[]Object :

名称	类型	描述
point	Array.<Number>	数据点坐标数组，格式[lon,lat]
name	String	数据点名称

名称	类型	描述
<code>style</code>	<code>Number</code>	数据点显示样式索引

`getData()`

描述:

-
-

获取数据

-
-

示例

```
let data = mass.getData();
```

返回:

数据

类型:

`Array.<Object>`

`setStyle(style, zoom)`

描述:

-
-

设置样式

-
-

示例

```
let style = [{
```

```

        images: './images/mass0.png',
        width:12,
        height:12
    }, {
        images: './images/mass1.png',
        width:10,
        height:10
    }, {
        images: './images/mass2.png',
        width:8,
        height:8
    }
    ]];
let mass = new KMap.MassMarks(data,{
    opacity: 0.8,
    style : style,
    coordinates:'GCJ02'
});
mass.setStyle(style);

```

参数:

名称	类型	描述
<code>style</code>	<code>Array.<PointStyle></code>	点样式数组
<code>zoom</code>	<code>Number</code>	样式对应比例尺级别，0 为所有级别，不传默认为 0

getStyle(zoom)

描述:

-
-

获取样式

-
-

示例

```
let styles = mass.getStyle();
```

参数:

名称	类型	描述
zoom	Number	样式对应比例尺级别，0 为所有级别，不传默认为 0

返回:

样式数组

类型:

Array.<PointStyle>

update()

描述:

-
-

更新

-
-

示例

```
mass.update();
```

Plane

new Plane (options)

描述:

-
-

平面

-
-

示例

```
let plane = new KMap.Plane({
  position:new KMap.Point(114.02308, 22.53474),
  width:100,
  height:100,
  zIndex:100,
  style:{
    "fillColor":"rgb(255,0,0)",
    "images":["./textures/CarelandLogo.png"],
  }
})
map.add(plane);
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
id	String	ID
zIndex	Number	覆盖物的叠加顺序
style	PolygonStyle	表达
width	Number	宽度，单位：米
height	Number	高度，单位：米

名称	类型	描述
<code>cursor</code>	<code>String</code>	鼠标移入样式
<code>bubble</code>	<code>Bubble</code>	事件是否冒泡，默认： <code>false</code>
<code>tip</code>	<code>optionsTipObject</code>	鼠标移入提示
<code>draggable</code>	<code>Boolean</code>	否可拖拽移动，默认： <code>false</code> ， <code>true</code> :可以拖拽， <code>false</code> :不可以拖拽
<code>visible</code>	<code>Boolean</code>	是否可见， <code>true</code> :显示， <code>false</code> :隐藏，默认显示
<code>zooms</code>	<code>Array.<Number></code>	显示级别范围 <code>eq:[4,20]</code>
<code>extData</code>	<code>Any</code>	自定义属性 自定义属性，用户自有属性，会原样返回
<code>position</code>	<code>Point</code>	经纬度坐标

optionsStyleObject :

名称	类型	描述
<code>fillColor</code>	<code>String</code>	填充 <code>eq:#8af5bd</code>
<code>borderWidth</code>	<code>Number</code>	边线宽度

名称	类型	描述
borderColor	String	边线颜色
borderStyle	String	边线样式，实线:solid，虚线:dashed，虚边线:borderDashed
borderDasharray	Number	虚线段 eq:[15,5]

optionsTipObject :

名称	类型	描述
text	String	提示文字
style	StyleSheet	样式，参考 css 样式

函数

```
setPosition(point,noUpdate)
```

描述:

-
-

设置坐标点

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
<code>point</code>	<code>Point</code>	坐标点

`getPosition()`

描述:

-
-

获取坐标点

-
-

返回:

类型:
`Point`

`getSize()`

描述:

-
-

获取立面高度

-
-

示例

```
rectangle.getHeight();
```

返回:

类型:
`Number`

setSize(width,height)

描述:

-
-

设置立面高度

-
-

示例

```
rectangle.setHeight(100);
```

参数:

名称	类型	描述
height	Number	单位米

getBounds()

描述:

-
-

获取外框

-
-

示例

```
circle.getBounds();
```

返回:

类型:
 Bounds

contains(point)

描述:

-
-

判断点是否在内

-
-

示例

```
rectangle.contains(new KMap.Point(114.02308, 22.53474));
```

参数:

名称	类型	描述
point	Point, Array. <Number>	经纬度

返回:

类型:

boolean

getDepthStatus()

描述:

-
-

获取深度开关

-
-

返回:

类型:

boolean

setDepthStatus(value)

描述:

-
-

设置深度开关

-
-

参数:

名称	类型	描述
value	boolean	

setDraggable(b)

描述:

-
-

设置是否可以拖拽

-
-

参数:

名称	类型	描述
b		

getDraggable()

描述:

-

-

获取是否可以拖拽

-
-

返回:

类型:
`boolean,*`

```
setOptions(opts)
```

描述:

-
-

设置参数

-
-

参数:

名称	类型	描述
opts		

```
setTip(tip)
```

描述:

-
-

设置鼠标移入提示

-
-

示例

```
polyline.setTip({text: '提示', style: {color:'red',padding:'5px'},
offset: new KMap.Pixel(0,10)})
```

参数:

名称	类型	描述
tip	Object	

```
getRotation()
```

描述:

-
-

获取逆时针倾斜度

-
-

返回:

类型: `Array.<number>`

```
getTip()
```

描述:

-
-

获取鼠标提示

-
-

返回:

类型: `*`

getOptions()

描述:

-
-

返回参数

-
-

返回:

类型: *

setCursor(cursor)

描述:

-
-

设置鼠标悬停样式

-
-

参数:

名称	类型	描述
----	----	----

cursor

getCursor()

描述:

-
-

获取鼠标悬停样式

-
-

返回:

类型:
*

setRotation(rotation)

描述:

-
-

设置逆时针旋转

-
-

示例

```
gltfSource.setRotation([1,2,3]);
```

参数:

名称	类型	描述
rotation	Array.<Number>	旋转弧度[x,y,z]

setPositionOffset(offset)

描述:

-
-

设置位置偏移

-

-

示例

```
gltfSource.setPositionOffset([1,2,3]);
```

参数:

名称	类型	描述
offset	Array.<Number>	单位米[x,y,x]

```
setRevolution(revolution)
```

描述:

-
-

设置公转

-
-

示例

```
gltfSource.setRevolution([1,2,3]);
```

参数:

名称	类型	描述
revolution	Array.<Number>	旋转弧度[x,y,z]

```
setAngle(angle)
```

描述:

-
-

设置水平方向角度

-
-

示例

```
labelMarker.setAngle(60);
```

参数:

名称	类型	描述
angle	Number	

getAngle()

描述:

-
-

获取水平角度

-
-

返回:

类型:
number

getId()

描述:

-
-

获取 ID

-
-

返回:

类型：
*,string

setId(id)

描述:

-
-

设置 ID

-
-

参数:

名称	类型	描述
----	----	----

id		
----	--	--

getzIndex()

描述:

-
-

获取层级

-
-

返回:

类型：
number,string

setzIndex(zIndex)

描述:

-
-

设置层级

-
-

参数:

名称	类型	描述
zIndex		

```
hide()
```

描述:

-
-

隐藏

-
-

```
show()
```

描述:

-
-

显示

-
-

```
setZooms(zooms)
```

描述:

-
-

设置比例尺范围

-
-

参数:

名称	类型	描述
zooms		

```
getZooms()
```

描述:

-
-

获取比例尺范围

-
-

返回:

类型:
Array.<Number>

```
getExtData()
```

描述:

-
-

获取用户自定义属性

-
-

返回:

类型:

**,null*

setExtData(*extData*)

描述:

-
-

设置用户自定义属性

-
-

参数:

名称	类型	描述
----	----	----

extData		
---------	--	--

getStyle()

描述:

-
-

获取样式表达

-
-

setStyle(*style*)

描述:

-
-

设置样式表达

-
-

参数:

名称	类型	描述
style	PointStyle, PolylineStyle, PolygonStyle	

on(type,listener)

描述:

-
-

添加事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型: BaseEvent

off(type,listener)

描述:

-
-

删除事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型:
BaseEvent

```
destroy()
```

描述:

-
-

销毁自己

-
-

属性

```
bubble
```

描述:

-
-

获取是否冒泡

-

-

draggable

描述:

-
-

获取是否可以拖拽

-
-

positionOffset

描述:

-
-

获取位置偏移

-
-

示例

```
gltfSource.positionOffset;
```

rotation

描述:

-
-

获取旋转

-
-

Polygon

new Polygon (options)

描述:

-
-

多边形

-
-

示例

```
let polygon = new KMap.Polygon({
  path : [
    [114.0223514, 22.5325797],
    [114.0238530, 22.5321540],
    [114.0236549, 22.5310677],
    [114.0220558, 22.5298341],
    [114.0208129, 22.5305882],
    [114.0210315, 22.5317664],
    [114.0223514, 22.5325797]
  ],
  height: 20,
  cursor: 'pointer',
  bubble : true,
  style : {
    borderStyle: 'dashed',
    borderDasharray : [10,5],
    borderWidth:5,
    borderColor:"#000FFF7F",
    fillColor: "#FF00007F",
  },
  tip:{text:'polyObj2',style:{color:'red'}}
})
map.add(polygon);
```

参数:

名称	类型	描述
----	----	----

optionsObject :

名称	类型	描述
id	String	ID 唯一标识

名称	类型	描述
zIndex	Number	覆盖物的叠加顺序
style	Array.<PolygonStyle>	表达数组，0 是顶面样式，其他为侧面样式 [{fillColor:"#FF00007F",borderWidth:2,borderColor:"#000FFF7F"} },{fillColor: "rgba(0,255,55,0.5)"}]
height	Number	立面高度，单位：米，eq: 20
cursor	String	鼠标移入样式,参考 css 鼠标样式
bubble	Bubble	事件是否冒泡
tip	optionsTipObject	鼠标移入提示
draggable	Boolean	否可拖拽移动，默认: false, true:可以拖拽, false:不可以拖拽
visible	Boolean	是否可见，true:显示，false:隐藏，默认显示
zooms	Array.<Number>	显示级别范围，数组[最小比例尺，最大比例尺]，eq:[4,20]
extData	Any	自定义属性，用户自有属性，会原样返回

名称	类型	描述
path:	<code>Array.<Array.<(Point Array.<Number>>></code>	
rotation:		<code>[x:number,y:number,z:number]</code> 旋转角度, 弧度单位, eq: <code>[Math.PI / 5,Math.PI / 2,Math.PI / 3]</code>
revolution:		<code>[x:number,y:number,z:number]</code> 公转角度, 弧度单位 eq: <code>[Math.PI / 5,Math.PI / 2,Math.PI / 3]</code>
positionOffset:		<code>[x:number,y:number,z:number]</code> 位置偏移量, 单位米 eq: <code>[-200, 0, 0]</code>

optionsStyleObject :

名称	类型	描述
fillColor	<code>String</code>	填充颜色 默认为 #8af5bd
borderWidth	<code>Number</code>	边线宽度,默认为 3
borderColor	<code>String</code>	边线颜色,默认为 #8aa8f5
borderStyle	<code>String</code>	边线样式, 默认: solid, 实线:solid, 虚线:dashed, 虚边线:borderDashed

名称	类型	描述
<code>borderDasharray</code>	<code>Number</code>	虚线段 <code>eq:[15,5]</code>

optionsTipObject :

名称	类型	描述
<code>text</code>	<code>String</code>	提示文字
<code>style</code>	<code>StyleSheet</code>	样式，参考 <code>css</code> 样式

函数

`getPath()`

描述:

-
-

获取多边形节点数组

-
-

示例

```
polygon.getPath();
```

返回:

类型:

`Array.<Array.<(Array.<Number>|Point)>>`

`setHeight(height)`

描述:

-

-

设置多边形立面高度

-

-

示例

```
polygon.setHeight(100)
```

参数:

名称	类型	描述
height	Number	立面高度

```
getHeight()
```

描述:

-

-

获取多边形立面高度

-

-

示例

```
polygon.getHeight()
```

返回:

类型:
number

```
getBounds()
```

描述:

-

-

获取当前折线的矩形范围对象

-
-

示例

```
polygon.getBounds();
```

返回:

类型:
 Bounds

```
setPath(p,noUpdate)
```

描述:

-
-

设置多边形形状点

-
-

参数:

名称	类型	描述
----	----	----

p	Array.<Array.<(Point Array.<Number>>>>
---	--

返回:

```
polygon.setPath([ [114.0223514, 22.5325797], [114.0238530, 22.5321540],  
[114.0236549, 22.5310677], [114.0220558, 22.5298341], [114.0208129,  
22.5305882], [114.0210315, 22.5317664] ]);
```

类型:
 boolean

```
getArea()
```

描述:

-
-

获取多边形的面积（单位：平方米）

-
-

示例

```
polygon.getArea();
```

返回:

类型:
Number

```
contains(point)
```

描述:

-
-

判断点是否在多边形内

-
-

示例

```
polygon.contains(new KMap.Point(114.02308, 22.53474));
```

参数:

名称	类型	描述
point	Point, Array.<Number>	经纬度

返回:

类型:
boolean

getDepthStatus()

描述:

-
-

获取深度开关

-
-

返回:

类型:

boolean

setDepthStatus(value)

描述:

-
-

设置深度开关

-
-

参数:

名称

类型

描述

value

boolean

setDraggable(b)

描述:

-
-

设置是否可以拖拽

-
-

参数:

名称	类型	描述
b		

```
getDraggable()
```

描述:

-
-

获取是否可以拖拽

-
-

返回:

类型:
boolean,*

```
setTip(tip)
```

描述:

-
-

设置鼠标移入提示

-
-

示例

```
polyline.setTip({text: '提示', style: {color:'red',padding:'5px'},
offset: new KMap.Pixel(0,10)})
```

参数:

名称	类型	描述
tip	Object	

getRotation()

描述:

-
-

获取逆时针倾斜度

-
-

返回:

类型: `Array.<number>`

getTip()

描述:

-
-

获取鼠标提示

-
-

返回:

类型: `*`

getOptions()

描述:

-
-

返回参数

-
-

返回:

类型:
*

setCursor(cursor)

描述:

-
-

设置鼠标悬停样式

-
-

参数:

名称	类型	描述
----	----	----

cursor		
--------	--	--

getCursor()

描述:

-
-

获取鼠标悬停样式

-
-

返回:

类型:
*

setRotation(rotation)

描述:

-
-

设置逆时针旋转

-
-

示例

```
gltfSource.setRotation([1,2,3]);
```

参数:

名称	类型	描述
rotation	Array.<Number>	旋转弧度[x,y,z]

setPositionOffset(offset)

描述:

-
-

设置位置偏移

-

-

示例

```
gltfSource.setPositionOffset([1,2,3]);
```

参数:

名称	类型	描述
offset	Array.<Number>	单位米[x,y,x]

```
setRevolution(revolution)
```

描述:

-
-

设置公转

-
-

示例

```
gltfSource.setRevolution([1,2,3]);
```

参数:

名称	类型	描述
revolution	Array.<Number>	旋转弧度[x,y,z]

```
setAngle(angle)
```

描述:

-
-

设置水平方向角度

-
-

示例

```
labelMarker.setAngle(60);
```

参数:

名称	类型	描述
angle	Number	

```
getAngle()
```

描述:

-
-

获取水平角度

-
-

返回:

类型:
number

```
getId()
```

描述:

-
-

获取 ID

-
-

返回:

类型：
*,string

setId(id)

描述:

-
-

设置 ID

-
-

参数:

名称	类型	描述
id		

getzIndex()

描述:

-
-

获取层级

-
-

返回:

类型：
number,string

setzIndex(zIndex)

描述:

-
-

设置层级

-
-

参数:

名称	类型	描述
----	----	----

zIndex		
--------	--	--

`hide()`

描述:

-
-

隐藏

-
-

`show()`

描述:

-
-

显示

-
-

`setZooms(zooms)`

描述:

-
-

设置比例尺范围

-
-

参数:

名称	类型	描述
zooms		

```
getZooms()
```

描述:

-
-

获取比例尺范围

-
-

返回:

类型:
Array.<Number>

```
getExtData()
```

描述:

-
-

获取用户自定义属性

-
-

返回:

类型:

**,null*

setExtData(*extData*)

描述:

-
-

设置用户自定义属性

-
-

参数:

名称

类型

描述

extData

getStyle()

描述:

-
-

获取样式表达

-
-

setStyle(*style*)

描述:

-
-

设置样式表达

-
-

参数:

名称	类型	描述
style	PointStyle, PolylineStyle, PolygonStyle	

on(type,listener)

描述:

-
-

添加事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型: BaseEvent

off(type,listener)

描述:

-
-

删除事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型:
BaseEvent

```
destroy()
```

描述:

-
-

销毁自己

-
-

属性

```
bubble
```

描述:

-
-

获取是否冒泡

-

-

draggable

描述:

-
-

获取是否可以拖拽

-
-

positionOffset

描述:

-
-

获取位置偏移

-
-

示例

```
gltfSource.positionOffset;
```

rotation

描述:

-
-

获取旋转

-
-

Polyline

new Polyline (options)

描述:

-
-

折线

-
-

示例

```
let polyline = new KMap.Polyline({
  path : [
    [114.0223514, 22.5325797],
    [114.0238530, 22.5321540],
    [114.0236549, 22.5310677],
    [114.0220558, 22.5298341],
    [114.0208129, 22.5305882],
    [114.0210315, 22.5317664]
  ],
  height: 20,
  cursor: 'pointer',
  bubble : true,
  style : {
    strokeStyle: 'dashed',
    strokeDasharray : [10,5],
    strokeWidth:5,
    strokeColor:"#000FFF7F",
    borderStyle: 'dashed',
    borderDasharray : [10,5],
    borderWidth:5,
    borderColor:"#000FFF7F"
  },
  tip:{text:'polyObj2',style:{color:'red'}}
})
map.add(polyline);
```

参数:

名称	类型	描述
options	optionsObject	对象参数

optionsObject :

名称	类型	描述
id	String	唯一 ID
zIndex	Number	覆盖物的叠加顺序
style	PolylineStyle	表达样式， {strokeColor:"#FF00007F",borderWidth:2,borderColor:"#000FFF7F"}
height	Number	海拔高度，米 eq:10
cursor	String	鼠标移入样式,参考 css 鼠标样式
bubble	Boolean	事件是否冒泡
tip	optionsTipObject	鼠标移入提示
draggable	Boolean	否可拖拽移动，默认: false, true:可以拖拽，false:不可以拖拽
visible	Boolean	是否可见，true:显示，false:隐藏，默认显示
zooms	Array.<Number>	显示级别范围，数组[最小比例尺，最大比例尺]，eq:[4,20]

名称	类型	描述
<code>extData</code>	<code>Any</code>	自定义属性，用户自有属性，会原样返回
<code>path</code>	<code>Array.<Array.<(Point Array.<Number>>></code>	形状点信息
<code>showDir</code>	<code>Boolean, Number</code>	是否延路径显示白色方向箭头,默认 false 。建议折线宽度大于 6 时使用
<code>continuousDir</code>	<code>Boolean, Number</code>	是否连续的延路径显示白色方向箭头,默认 false 。建议折线宽度大于 6 时使用
<code>rotation:</code>		<code>[x:number,y:number,z:number]</code> 旋转角度，弧度单位，eq: <code>[Math.PI / 5,Math.PI / 2,Math.PI / 3]</code>
<code>revolution:</code>		<code>[x:number,y:number,z:number]</code> 公转角度，弧度单位 eq: <code>[Math.PI / 5,Math.PI / 2,Math.PI / 3]</code>
<code>positionOffset:</code>		<code>[x:number,y:number,z:number]</code> 位置偏移量，单位米 eq: <code>[-200, 0, 0]</code>

optionsStyleObject :

名称	类型	描述
<code>strokeColor</code>	<code>string</code>	线颜色 默认: <code>#8af5bd</code> ,

名称	类型	描述
		eq:#FF00007F,rgba(0,255,55,0.5))
strokeStyle	string	线样式 solid,dashed,默认:solid
strokeDasharray	Array.<Number>	虚线段 eq:[5,10]
strokeWidth	Number	线宽度,默认:6
borderColor	string	边线颜色 eq:#FF00007F,rgba(0,255,55,0.5))
borderStyle	string	边线样式 solid,dashed,默认:solid
borderDasharray	Array.<Number>	虚线段 eq:[5,10]
borderWidth	Number	边线宽度,默认:0

optionsTipObject :

名称	类型	描述
text	String	提示文字
style	StyleSheet	样式, 参考 css 样式

函数

getPath()

描述:

-
-

获取折线节点数组

-
-

示例

```
polyline.getPath();
```

返回:

类型:

```
Array.<Array.<(Array.<Number>|Point)>>>
```

setHeight(*height*)

描述:

-
-

设置折线海拔高度

-
-

示例

```
polyline.setHeight(100);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
height	Number	海拔高度，当坐标的 z 没设置时，会以 height 为主

getHeight()

描述:

-
-

获取折线海拔高度

-
-

示例

```
polyline.getHeight();
```

返回:

类型:

Number

getBounds()

描述:

-
-

获取当前折线的矩形范围对象

-
-

示例

```
polyline.getBounds();
```

返回:

类型:

Bounds

setPath(p,noUpdate)

描述:

-
-

设置折线路径的节点数组

-
-

示例

```

polygon.setPath([
    [114.0223514, 22.5325797],
    [114.0238530, 22.5321540],
    [114.0236549, 22.5310677],
    [114.0220558, 22.5298341],
    [114.0208129, 22.5305882],
    [114.0210315, 22.5317664]
]);

```

参数:

名称	类型	描述
----	----	----

p	Array.<Array.<(Point Array.<Number>>>>	
---	--	--

getLength()

描述:

-
-

获取折线长度

-
-

示例

```

polyline.getLength();

```

返回:

类型:

number

contains(point,tolerance)

描述:

-
-

判断坐标是否在折线上

-
-

示例

```
polyline.contains(new KMap.Point(114.02308, 22.53474),10);
```

参数:

名称

类型

描述

point

Point, Array. <Number>

tolerance

Number

容忍度，单位米

返回:

类型:

boolean

getDepthStatus()

描述:

-
-

获取深度开关

-
-

返回:

类型:

boolean

```
setDepthStatus(value)
```

描述:

-
-

设置深度开关

-
-

参数:

名称	类型	描述
value	boolean	

```
setDraggable(b)
```

描述:

-
-

设置是否可以拖拽

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
b		

getDraggable()

描述:

-
-

获取是否可以拖拽

-
-

返回:

类型:
boolean,*

setTip(*tip*)

描述:

-
-

设置鼠标移入提示

-
-

示例

```
polyline.setTip({text: '提示', style: {color:'red',padding:'5px'},  
offset: new KMap.Pixel(0,10)})
```

参数:

名称	类型	描述
tip	Object	

getRotation()

描述:

-
-

获取逆时针倾斜度

-
-

返回:

类型:

Array.<number>

getTip()

描述:

-
-

获取鼠标提示

-
-

返回:

类型:

*

getOptions()

描述:

-
-

返回参数

-
-

返回:

类型:
*

```
setCursor(cursor)
```

描述:

-
-

设置鼠标悬停样式

-
-

参数:

名称	类型	描述
cursor		

```
getCursor()
```

描述:

-
-

获取鼠标悬停样式

-
-

返回:

类型:
*

setRotation(rotation)

描述:

-
-

设置逆时针旋转

-
-

示例

```
gltfSource.setRotation([1,2,3]);
```

参数:

名称	类型	描述
rotation	Array.<Number>	旋转弧度[x,y,z]

setPositionOffset(offset)

描述:

-
-

设置位置偏移

-
-

示例

```
gltfSource.setPositionOffset([1,2,3]);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
offset	Array.<Number>	单位米[x,y,x]

setRevolution(revolution)

描述:

-
-

设置公转

-
-

示例

```
gltfSource.setRevolution([1,2,3]);
```

参数:

名称	类型	描述
revolution	Array.<Number>	旋转弧度[x,y,z]

setAngle(angle)

描述:

-
-

设置水平方向角度

-
-

示例

```
labelMarker.setAngle(60);
```

参数:

名称	类型	描述
angle	Number	

getAngle()

描述:

-
-

获取水平角度

-
-

返回:

类型:
number

getId()

描述:

-
-

获取 ID

-
-

返回:

类型:
*,string

setId(id)

描述:

-
-

设置 ID

-
-

参数:

名称	类型	描述
----	----	----

id		
----	--	--

getzIndex()

描述:

-
-

获取层级

-
-

返回:

类型: number, string

setzIndex(zIndex)

描述:

-
-

设置层级

-
-

参数:

名称	类型	描述
zIndex		

```
hide()
```

描述:

-
-

隐藏

-
-

```
show()
```

描述:

-
-

显示

-
-

```
setZooms(zooms)
```

描述:

-
-

设置比例尺范围

-
-

参数:

名称	类型	描述
zooms		

```
getZooms()
```

描述:

-
-

获取比例尺范围

-
-

返回:

类型: `Array.<Number>`

```
getExtData()
```

描述:

-
-

获取用户自定义属性

-
-

返回:

类型: `*,null`

setExtData(*extData*)

描述:

-
-

设置用户自定义属性

-
-

参数:

名称	类型	描述
extData		

getStyle()

描述:

-
-

获取样式表达

-
-

setStyle(*style*)

描述:

-
-

设置样式表达

-
-

参数:

名称	类型	描述
style	PointStyle, PolylineStyle, PolygonStyle	

on(type,listener)

描述:

-
-

添加事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型: BaseEvent

off(type,listener)

描述:

-
-

删除事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型:
BaseEvent

destroy()

描述:

- -
- 销毁自己

-
-

属性

bubble

描述:

- -
- 获取是否冒泡

-
-

draggable

描述:

-
-

获取是否可以拖拽

-
-

positionOffset

描述:

-
-

获取位置偏移

-
-

示例

```
gltfSource.positionOffset;
```

rotation

描述:

-
-

获取旋转

-
-

Rectangle

new Rectangle (options)

描述:

-
-

矩形

-
-

示例

```
let rectangle = new KMap.Rectangle({
  bounds : new KMap.Bounds(new KMap.Point(114.022134,22.533947),
new KMap.Point(114.024,22.535420)),
  height: 20,
  cursor: 'pointer',
  bubble : true,
  style : {
    borderStyle: 'dashed',
    borderDasharray : [10,5],
    borderWidth:5,
    borderColor:"#000FFF7F",
    fillColor: "#FF00007F",
  },
  tip:{text:'polyObj2',style:{color:'red'}}
})
map.add(rectangle);
```

参数:

名称	类型	描述
----	----	----

options	optionsObject
---------	---------------

optionsObject :

名称	类型	描述
----	----	----

id	String	ID
----	--------	----

zIndex	Number	覆盖物的叠加顺序
--------	--------	----------

style	Array.<PolygonStyle>	样式表达数组，0 是顶面样式，其他为侧面样式 [{fillColor:"#FF00007F",borderWidth:2,borderColor:"#000FFF7F"},{fillColor:
-------	----------------------	--

名称	类型	描述
		<code>"rgba(0,255,55,0.5)"}]</code>
<code>height</code>	<code>Number</code>	立面高度，单位：米 <code>eq:10</code>
<code>cursor</code>	<code>String</code>	鼠标移入样式,参考 <code>css</code> 鼠标样式
<code>bubble</code>	<code>Bubble</code>	事件是否冒泡
<code>tip</code>	<code>optionsTip Object</code>	鼠标移入提示
<code>draggable</code>	<code>Boolean</code>	否可拖拽移动，默认： <code>false</code> ， <code>true</code> :可以拖拽， <code>false</code> :不可以拖拽
<code>visible</code>	<code>Boolean</code>	是否可见， <code>true</code> :显示， <code>false</code> :隐藏，默认显示
<code>zooms</code>	<code>Array.<Number></code>	显示级别范围 ，数组[最小比例尺，最大比例尺]， <code>eq:[4,20]</code>
<code>extData</code>	<code>Any</code>	自定义属性，用户自有属性，会原样返回
<code>bounds:</code>	<code>Bounds</code>	矩形外框
<code>alt:</code>	<code>Number</code>	矩形高程，单位：米
<code>rotation:</code>		<code>[x:number,y:number,z:number]</code> 旋转角度，弧

名称	类型	描述
		度单位, eq: [Math.PI / 5,Math.PI / 2,Math.PI / 3]
revolution:		[x:number,y:number,z:number] 公转角度, 弧度单位 eq: [Math.PI / 5,Math.PI / 2,Math.PI / 3]
position Offset:		[x:number,y:number,z:number] 位置偏移量, 单位米 eq: [-200, 0, 0]

optionsStyleObject :

名称	类型	描述
fillColor	String	填充 eq:#8af5bd
borderWidth	Number	边线宽度,默认:3
borderColor	String	边线颜色 默认:#8aa8f5, eq:#FF00007F,rgba(0,255,55,0.5)
borderStyle	String	边线样式, 实线:solid, 虚线:dashed, 虚边 线:borderDashed
borderDasharray	Number	虚线段 eq:[15,5]

optionsTipObject :

名称	类型	描述
<code>text</code>	<code>String</code>	提示文字
<code>style</code>	<code>StyleSheet</code>	样式，参考 <code>css</code> 样式

函数

```
setBounds(bounds,noUpdate)
```

描述:

-
-

设置框

-
-

示例

```
rectangle.setBounds(new KMap.Bounds(new KMap.Point(114.022134,22.533947), new KMap.Point(114.024,22.535420)));
```

参数:

名称	类型	描述
<code>bounds</code>	<code>Bounds</code>	框

```
getBounds()
```

描述:

-
-

返回框

-
-

示例

```
rectangle.getBounds();
```

返回:

类型:
 Bounds

setAlt(alt)

描述:

-
-

设置矩形框高程 单位米

-
-

参数:

名称	类型	描述
alt	Number	

getAlt()

描述:

-
-

获取矩形框高程，单位米

-
-

返回:

类型：
Number

getHeight()

描述：

-
-

获取立面高度

-
-

示例

```
rectangle.getHeight();
```

返回：

类型：
Number

setHeight(*height*)

描述：

-
-

设置立面高度

-
-

示例

```
rectangle.setHeight(100);
```

参数：

名称	类型	描述
----	----	----

名称	类型	描述
height	Number	单位米

contains(*point*)

描述:

-
-

判断点是否在矩形内

-
-

示例

```
rectangle.contains(new KMap.Point(114.02308, 22.53474));
```

参数:

名称	类型	描述
point	Point, Array.<Number>	经纬度

返回:

类型:

boolean

getDepthStatus()

描述:

-
-

获取深度开关

-

-

返回:

类型:

boolean

```
setDepthStatus(value)
```

描述:

-
-

设置深度开关

-
-

参数:

名称	类型	描述
value	boolean	

```
setDraggable(b)
```

描述:

-
-

设置是否可以拖拽

-
-

参数:

名称	类型	描述
b		

getDraggable()

描述:

-
-

获取是否可以拖拽

-
-

返回:

类型: `boolean,*`

setOptions(*opts*)

描述:

-
-

设置参数

-
-

参数:

名称	类型	描述
----	----	----

<code>opts</code>

setTip(*tip*)

描述:

-
-

设置鼠标移入提示

-
-

示例

```
polyline.setTip({text: '提示', style: {color:'red',padding:'5px'},  
offset: new KMap.Pixel(0,10)})
```

参数:

名称	类型	描述
tip	Object	

```
getRotation()
```

描述:

-
-

获取逆时针倾斜度

-
-

返回:

类型:
Array.<number>

```
getTip()
```

描述:

-
-

获取鼠标提示

-

-

返回:

类型: *

```
getOptions()
```

描述:

-
-

返回参数

-
-

返回:

类型: *

```
setCursor(cursor)
```

描述:

-
-

设置鼠标悬停样式

-
-

参数:

名称	类型	描述
cursor		

getCursor()

描述:

-
-

获取鼠标悬停样式

-
-

返回:

类型: *

setRotation(rotation)

描述:

-
-

设置逆时针旋转

-
-

示例

```
gltfSource.setRotation([1,2,3]);
```

参数:

名称	类型	描述
rotation	Array.<Number>	旋转弧度[x,y,z]

setPositionOffset(offset)

描述:

-
-

设置位置偏移

-
-

示例

```
gltfSource.setPositionOffset([1,2,3]);
```

参数:

名称	类型	描述
offset	Array.<Number>	单位米[x,y,x]

setRevolution(revolution)

描述:

-
-

设置公转

-
-

示例

```
gltfSource.setRevolution([1,2,3]);
```

参数:

名称	类型	描述
revolution	Array.<Number>	旋转弧度[x,y,z]

setAngle(angle)

描述:

-
-

设置水平方向角度

-
-

示例

```
labelMarker.setAngle(60);
```

参数:

名称	类型	描述
angle	Number	

```
getAngle()
```

描述:

-
-

获取水平角度

-
-

返回:

类型:
number

```
getId()
```

描述:

-
-

获取 ID

-
-

返回:

类型:
`*,string`

`setId(id)`

描述:

-
-

设置 ID

-
-

参数:

名称	类型	描述
----	----	----

id		
----	--	--

`getzIndex()`

描述:

-
-

获取层级

-
-

返回:

类型：
number, string

setzIndex(zIndex)

描述:

-
-

设置层级

-
-

参数:

名称	类型	描述
----	----	----

zIndex		
--------	--	--

hide()

描述:

-
-

隐藏

-
-

show()

描述:

-
-

显示

-
-

setZooms(zooms)

描述:

-
-

设置比例尺范围

-
-

参数:

名称

类型

描述

zooms

getZooms()

描述:

-
-

获取比例尺范围

-
-

返回:

类型:

Array.<Number>

getExtData()

描述:

-
-

获取用户自定义属性

-
-

返回:

类型:
`*,null`

```
setExtData(extData)
```

描述:

-
-

设置用户自定义属性

-
-

参数:

名称	类型	描述
----	----	----

extData

```
getStyle()
```

描述:

-
-

获取样式表达

-
-

setStyle(style)

描述:

-
-

设置样式表达

-
-

参数:

名称	类型	描述
----	----	----

style	PointStyle, PolylineStyle, PolygonStyle	
-------	---	--

on(type,listener)

描述:

-
-

添加事件

-
-

参数:

名称	类型	描述
----	----	----

type		
------	--	--

listener		
----------	--	--

返回:

类型:

BaseEvent

off(type,listener)

描述:

-
-

删除事件

-
-

参数:

名称	类型	描述
----	----	----

type		
------	--	--

listener		
----------	--	--

返回:

类型:

BaseEvent

destroy()

描述:

-
-

销毁自己

-
-

属性

bubble

描述:

-
-

获取是否冒泡

-
-

draggable

描述:

-
-

获取是否可以拖拽

-
-

positionOffset

描述:

-
-

获取位置偏移

-
-

示例

```
gltfSource.positionOffset;
```

rotation

描述:

-
-

获取旋转

-
-

Tube

new Tube (options)

描述:

-
-

管道

-
-

示例

```
let Tube = new KMap.Tube({
  path : [
    [114.0223514, 22.5325797,0],
    [114.0238530, 22.5321540,30],
    [114.0236549, 22.5310677,50],
    [114.0220558, 22.5298341,60],
    [114.0208129, 22.5305882,70],
    [114.0210315, 22.5317664,80],
    [114.0223514, 22.5325797,100]
  ],
  style : {
    fillColor: "#FF00007F",
  }
})
map.add(polygon);
```

参数:

名称	类型	描述
options	optionsObject	对象参数

optionsObject :

名称	类型	描述
----	----	----

名称	类型	描述
<code>id</code>	<code>String</code>	ID 唯一 ID
<code>zIndex</code>	<code>Number</code>	管道的叠加顺序, eq: 1000
<code>tubularSegments</code>	<code>Number</code>	组成这一管道的分段数, 默认值为 64 , 数值越大越精细, 太大性能会下降
<code>radius</code>	<code>Number</code>	管道的半径, 默认值: 1
<code>radialSegments</code>	<code>Number</code>	管道横截面的分段数目, 默认值为 8 , , 数值越大越精细, 太大性能会下降
<code>side</code>	<code>String</code>	可见显示面: front : 正面可见, double : 双面可见
<code>closed</code>	<code>Boolean</code>	管道的两端是否闭合, 默认值为 false
<code>style</code>	<code>optionsStyleObject</code>	表达信息
<code>cursor</code>	<code>String</code>	鼠标移入样式, 参考 <code>css</code> 鼠标样式

名称	类型	描述
<code>bubble</code>	<code>Bubble</code>	事件是否冒泡
<code>materialType</code>	<code>String</code>	材质类型，如： <code>MeshStandardMaterial</code> , 其他材质需要配合环境纹理或是环境色使用，使用材质需要开启场景： <code>map.setEnvironment('*.hdr');</code>
<code>tip</code>	<code>optionsTipObject</code>	鼠标移入提示
<code>draggable</code>	<code>Boolean</code>	否可拖拽移动，默认： <code>false</code> , <code>true</code> :可以拖拽， <code>false</code> :不可以拖拽
<code>visible</code>	<code>Boolean</code>	是否可见， <code>true</code> :显示， <code>false</code> :隐藏，默认显示
<code>zooms</code>	<code>Array.<Number></code>	显示级别范围，数组[最小比例尺，最大比例尺]， <code>eq:[4,20]</code>
<code>extData</code>	<code>Any</code>	自定义属性 自定义属性，用户自有属性，会原样返回

名称	类型	描述
path:	<code>Array.<(Point Array.<Number>)></code>	形状点信息 eq: <code>[[114.022, 22.5325, 0], [114.023, 22.532, 30], [114.023, 22.531, 50], [114.022, 22.529, 60], [114.020, 22.530, 70]]</code>
rotation:		<code>[x:number,y:number,z:number]</code> 旋转角度，弧度单位，eq: <code>[Math.PI / 5, Math.PI / 2, Math.PI / 3]</code>
revolution:		<code>[x:number,y:number,z:number]</code> 公转角度，弧度单位 eq: <code>[Math.PI / 5, Math.PI / 2, Math.PI / 3]</code>
positionOffset:		<code>[x:number,y:number,z:number]</code> 位置偏移量，单位米 eq: <code>[-200, 0, 0]</code>

optionsStyleObject :

名称	类型	描述
fillColor	<code>String</code>	填充 eq: <code>#8af5bd</code>

optionsTipObject :

名称	类型	描述
text	String	提示文字
style	StyleSheet	样式，参考 css 样式

函数

```
getTubularSegments()
```

描述:

- -
- 组成管道的分段数

-
-

返回:

类型:
number

```
setTubularSegments(tubularSegments)
```

描述:

- -
- 组成管道的分段数，默认值为 64

-
-

参数:

名称	类型	描述
tubularSegments	number	

getRadius()

描述:

-
-

获取管道的半径

-
-

返回:

类型:

number

setRadius(radius)

描述:

-
-

设置管道的半径

-
-

参数:

名称	类型	描述
radius	number	

getRadialSegments()

描述:

-
-

设置管道横截面的分段目

-
-

返回:

类型:
number

setRadialSegments(radialSegments)

描述:

-
-

设置管道横截面的分段目

-
-

参数:

名称	类型	描述
radialSegments	number	

getClosed()

描述:

-
-

获取管道的两端是否闭合

-
-

返回:

类型:

boolean

setClosed(*closed*)

描述:

-
-

设置管道的两端是否闭合

-
-

参数:

名称	类型	描述
----	----	----

closed	boolean	
--------	---------	--

getSide()

描述:

-
-

获取可见面

-
-

返回:

类型：
String

setSide(side)

描述:

-
-

设置管道的正面可见 ， front： 正面可见， double： 双面可见

-
-

参数:

名称	类型	描述
side	String	

getMaterialType()

描述:

-
-

获取材质类型

-
-

返回:

类型：
String

setMaterialType(materialType)

描述:

-
-

设置材质类型，如:MeshStandardMaterial, 其他材质需要配合环境纹理或是环境色使用

-
-

参数:

名称	类型	描述
radialSegments	number	

```
getPath()
```

描述:

-
-

获取多边形节点数组

-
-

示例

```
polygon.getPath();
```

返回:

类型:
Array.<Array.<(Array.<Number>|Point)>>

```
setStyle(styles)
```

描述:

-
-

设置样式表达

-
-

示例

```
polygon.setStyle({
    borderStyle: 'dashed',
    borderDasharray : [10,5],
    borderWidth:5,
    borderColor:"#000FFF7F",
    fillColor: "#FF00007F",
});
```

参数:

名称	类型	描述
styles	PolygonStyle	多边形样式

```
getBounds()
```

描述:

-
-

获取当前折线的矩形范围对象

-
-

示例

```
polygon.getBounds();
```

返回:

类型: Bounds

```
setPath(p,noUpdate)
```

描述:

-
-

设置多边形形状点

-
-

参数:

名称	类型	描述
----	----	----

p	Array.<Array.<(Point Array.<Number>>>>	
---	--	--

返回:

```
polygon.setPath([ [114.0223514, 22.5325797], [114.0238530, 22.5321540],  
[114.0236549, 22.5310677], [114.0220558, 22.5298341], [114.0208129,  
22.5305882], [114.0210315, 22.5317664] ]);
```

类型:
boolean

getArea()

描述:

-
-

获取多边形的面积（单位：平方米）

-
-

示例

```
polygon.getArea();
```

返回:

类型:
Number

contains(point)

描述:

-
-

判断点是否在多边形内

-
-

示例

```
polygon.contains(new KMap.Point(114.02308, 22.53474));
```

参数:

名称	类型	描述
point	Point, Array. <Number>	经纬度

返回:

类型:

boolean

getDepthStatus()

描述:

-
-

获取深度开关

-
-

返回:

类型:

boolean

setDepthStatus(value)

描述:

-
-

设置深度开关

-
-

参数:

名称	类型	描述
----	----	----

value	boolean	
-------	---------	--

setDraggable(b)

描述:

-
-

设置是否可以拖拽

-
-

参数:

名称	类型	描述
----	----	----

b		
---	--	--

getDraggable()

描述:

-

-

获取是否可以拖拽

-
-

返回:

类型:

`boolean,*`

`setTip(tip)`

描述:

-
-

设置鼠标移入提示

-
-

示例

```
polyline.setTip({text: '提示', style: {color:'red',padding:'5px'},  
offset: new KMap.Pixel(0,10)})
```

参数:

名称	类型	描述
----	----	----

<code>tip</code>	<code>Object</code>	
------------------	---------------------	--

`getRotation()`

描述:

-
-

获取逆时针倾斜度

-
-

返回:

类型:

`Array.<number>`

`getTip()`

描述:

-
-

获取鼠标提示

-
-

返回:

类型:

`*`

`getOptions()`

描述:

-
-

返回参数

-
-

返回:

类型:

`*`

setCursor(cursor)

描述:

-
-

设置鼠标悬停样式

-
-

参数:

名称	类型	描述
----	----	----

cursor		
--------	--	--

getCursor()

描述:

-
-

获取鼠标悬停样式

-
-

返回:

类型: *

setRotation(rotation)

描述:

-
-

设置逆时针旋转

-
-

示例

```
gltfSource.setRotation([1,2,3]);
```

参数:

名称	类型	描述
rotation	Array.<Number>	旋转弧度[x,y,z]

setPositionOffset(offset)

描述:

-
-

设置位置偏移

-
-

示例

```
gltfSource.setPositionOffset([1,2,3]);
```

参数:

名称	类型	描述
offset	Array.<Number>	单位米[x,y,x]

setRevolution(revolution)

描述:

-
-

设置公转

-
-

示例

```
gltfSource.setRevolution([1,2,3]);
```

参数:

名称	类型	描述
revolution	Array. <Number>	旋转弧度[x,y,z]

```
setAngle(angle)
```

描述:

-
-

设置水平方向角度

-
-

示例

```
labelMarker.setAngle(60);
```

参数:

名称	类型	描述
angle	Number	

```
getAngle()
```

描述:

-
-

获取水平角度

-
-

返回:

类型:
`number`

`getId()`

描述:

-
-

获取 ID

-
-

返回:

类型:
`*,string`

`setId(id)`

描述:

-
-

设置 ID

-
-

参数:

名称	类型	描述
id		

getzIndex()

描述:

-
-

获取层级

-
-

返回:

类型:
number,string

setzIndex(zIndex)

描述:

-
-

设置层级

-
-

参数:

名称	类型	描述
zIndex		

hide()

描述:

-
-

隐藏

-
-

show()

描述:

-
-

显示

-
-

setZooms(zooms)

描述:

-
-

设置比例尺范围

-
-

参数:

名称

类型

描述

zooms

getZooms()

描述:

-
-

获取比例尺范围

-
-

返回:

类型:

`Array.<Number>`

getExtData()

描述:

-
-

获取用户自定义属性

-
-

返回:

类型:

`*,null`

setExtData(extData)

描述:

-
-

设置用户自定义属性

-
-

参数:

名称	类型	描述
extData		

```
getStyle()
```

描述:

-
-

获取样式表达

-
-

```
on(type,listener)
```

描述:

-
-

添加事件

-
-

参数:

名称	类型	描述
type		
listener		

返回:

类型:
BaseEvent

off(type,listener)

描述:

-
-

删除事件

-
-

参数:

名称	类型	描述
----	----	----

type		
------	--	--

listener		
----------	--	--

返回:

类型:
BaseEvent

destroy()

描述:

-
-

销毁自己

-
-

属性

bubble

描述:

-
-

获取是否冒泡

-
-

draggable

描述:

-
-

获取是否可以拖拽

-
-

positionOffset

描述:

-
-

获取位置偏移

-
-

示例

```
gltfSource.positionOffset;
```

rotation

描述:

-
-

获取旋转

-
-

Event

事件

`show()`

描述:

-
-

显示事件

-
-

参数:

名称	类型	描述
e	Object	结果

`hide()`

描述:

-
-

隐藏事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
click()
```

描述:

-
-

单击事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
dblclick()
```

描述:

-
-

双击事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
rightclick()
```

描述:

-
-

右键点击事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
mousemove()
```

描述:

-
-

鼠标在上移动事件

-
-

参数:

名称	类型	描述
e	Object	结果

mouseover()

描述:

-
-

鼠标移入事件

-
-

参数:

名称	类型	描述
e	Object	结果

mouseout()

描述:

-
-

鼠标移出事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

mousedown()

描述:

-
-

鼠标按下事件

-
-

参数:

名称	类型	描述
e	Object	结果

mouseup()

描述:

-
-

鼠标弹起事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

dragstart()

描述:

-
-

拖拽开始事件

-
-

参数:

名称	类型	描述
e	Object	结果

dragging()

描述:

-
-

拖拽中事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

dragend()

描述:

-
-

拖拽结束事件

-
-

参数:

名称	类型	描述
e	Object	结果

touchstart()

描述:

-
-

触摸开始事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

```
touchmove()
```

描述:

-
-

触摸移动事件

-
-

参数:

名称	类型	描述
e	Object	结果

```
touchend()
```

描述:

-
-

触摸结束事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

OverlayGroup

```
new OverlayGroup ()
```

描述:

-
-

覆盖物组

-
-

示例

```
let overlayGroup = new KMap.OverlayGroup();  
map.add(overlayGroup);
```

函数

```
getIndex(overlay)
```

描述:

-
-

获取覆盖物下标

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
overlay	DomOverlay, Feature	

返回:

下标 overlayGroup.getIndex(marker);

类型:
Number

```
add(obj)
```

描述:

-
-

添加覆盖物到集合中

-
-

示例

```
overlayGroup.add(circle);
overlayGroup.add(labelMarker);
overlayGroup.add(polygon);
overlayGroup.add(polyline);
overlayGroup.add(marker);
overlayGroup.add(textMarker);
```

参数:

名称	类型	描述
obj	DomOverlay, Feature	

```
getAll()
```

描述:

-
-

返回当前集合中所有覆盖物

-
-

示例

```
overlayGroup.getAll();
```

返回:

类型:
`Array.<DomOverlay>,Array.<Feature>`

```
has(overlay)
```

描述:

-
-

判断传入的覆盖物是否在集合中

-
-

示例

```
overlayGroup.has(marker);
```

参数:

名称	类型	描述
overlay	DomOverlay, Feature	

返回:

类型:
`Boolean`

```
remove(obj)
```

描述:

-

-

从组中移除

-
-

示例

```
overlayGroup.remove(marker);  
overlayGroup.remove([marker,labelMarker]);
```

参数:

名称	类型	描述
----	----	----

obj	DomOverlay, Feature, Array.<DomOverlay>, Array.<Feature>	
-----	--	--

clear()

描述:

-
-

清空集合

-
-

示例

```
overlayGroup.clear();
```

show()

描述:

-
-

在地图上显示集合中覆盖物

-

-

示例

```
overlayGroup.show();
```

hide()

描述:

-
-

在地图上隐藏集合中覆盖物

-
-

示例

```
overlayGroup.hide();
```

setOptions(props)

描述:

-
-

修改覆盖物属性(包括线样式、颜色色等等)

-
-

示例

```
overlayGroup.setOptions({style:{fillColor: 'rgba(255,228,181, 0.5)',borderColor: 'rgba(0, 100, 0, 0.5)',strokeColor: 'rgba(221,160,221, 0.5)'}});
```

参数:

名称

类型

描述

props

Object

```
destroy()
```

描述:

-
-

销毁集合，但不销毁集合内覆盖物

-
-

示例

```
overlayGroup.destroy();
destroy = null;
```

工具

CircleEditor

```
new CircleEditor (map,circle,options)
```

描述:

-
-

编辑 KMap.Circle 对象，编辑圆半径大小、拖动圆心改变圆的位置。

-
-

示例

```
let circleeditor = new KMap.CircleEditor(map, circle,
{controlPointStyle:{fillColor:"#000FFF"}})
let circleeditor = new KMap.CircleEditor(map, circle)
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
<code>map</code>	<code>Map</code>	地图对象
<code>circle</code>	<code>Circle</code>	要编辑的对象
<code>options</code>	<code>optionsObject</code>	配置参数对象

optionsObject :

名称	类型	描述
<code>style</code>	<code>PolygonStyle</code>	圆样式表达
<code>controlPointStyle</code>	<code>PolygonStyle</code>	控制点样式
<code>controlPointRadius</code>	<code>Number</code>	控制点半径,单位: 米, 默认: 6
<code>midControlPointStyle</code>	<code>PolygonStyle</code>	中心点样式, <code>PolygonStyle</code> 格式样式
<code>midControlPointRadius</code>	<code>Number</code>	中心点半径,单位: 米, 默认: 6

optionsControlPointStyleObject :

名称	类型	描述
<code>fillColor</code>	<code>String</code>	填充颜色
<code>borderColor</code>	<code>String</code>	边线颜色 eq: #FF00007F, rgba(0,255,55,0.5)

名称	类型	描述
borderWidth	String	边线宽度,默认:3

optionsMidControlPointStyleObject :

名称	类型	描述
fillColor	String	填充颜色，默认： #48a8fc
borderColor	String	边线颜色，默认： #ffffff
borderWidth	String	边线宽度，默认： 1

函数

getTarget()

描述:

-
-

获取编辑对象

-
-

返回:

圆对象

类型:

Circle

setTarget(circle)

描述:

-
-

设置编辑对象

-
-

参数:

名称	类型	描述
circle	Circle	圆对象

返回:

类型:

boolean

open()

描述:

-
-

开启编辑功能

-
-

close()

描述:

-
-

关闭编辑功能

-
-

事件

start()

描述:

-
-

开始编辑事件

-
-

参数:

名称	类型	描述
circle	Circle	编辑对象

move()

描述:

-
-

移动覆盖物时触发此事件

-
-

参数:

名称	类型	描述
circle	Circle	编辑对象

```
adjust()
```

描述:

-
-

调整折线上某个点时触发此事件

-
-

参数:

名称	类型	描述
circle	Circle	编辑对象

```
removenode()
```

描述:

-
-

移除一个节点时触发此事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
circle	Circle	编辑对象

end()

描述:

-
-

移动覆盖物时触发此事件

-
-

参数:

名称	类型	描述
circle	Circle	编辑对象

CircleMarker

事件

change()

描述:

-
-

改变圆形标注时触发此事件

-
-

参数:

名称	类型	描述
circleMarker	CircleMarker	编辑对象

EllipseEditor

```
new EllipseEditor (map,ellipse,options)
```

描述:

-
-

编辑 KMap.Ellipse 对象，编辑圆半径大小、拖动圆心改变圆的位置，选择角度。

-
-

示例

```
let ellipseeditor = new KMap.EllipseEditor(map, ellipse, {controlPointStyle:{fillColor:"#000FFF"}})
let ellipseeditor = new KMap.EllipseEditor(map, ellipse)
```

参数:

名称	类型	描述
map	Map	地图对象
ellipse	Ellipse	要编辑的对象
options	optionsObject	配置参数对象

optionsObject :

名称	类型	描述
<code>style</code>	<code>PolygonStyle</code>	椭圆样式表达
<code>controlPointStyle</code>	<code>PolygonStyle</code>	控制点样式
<code>controlPointRadius</code>	<code>Number</code>	控制点半径,单位: 米, 默认: 6
<code>midControlPointStyle</code>	<code>PolygonStyle</code>	中心点样式, <code>PolygonStyle</code> 格式样式
<code>midControlPointRadius</code>	<code>Number</code>	中心点半径,单位: 米, 默认: 6

optionsControlPointStyleObject :

名称	类型	描述
<code>fillColor</code>	<code>String</code>	填充颜色
<code>borderColor</code>	<code>String</code>	边线颜色 eq:#FF00007F,rgba(0,255,55,0.5)
<code>borderWidth</code>	<code>String</code>	边线宽度,默认:3

optionsMidControlPointStyleObject :

名称	类型	描述
<code>fillColor</code>	<code>String</code>	填充颜色, 默认: #48a8fc
<code>borderColor</code>	<code>String</code>	边线颜色, 默认: #ffffff

名称	类型	描述
borderWidth	String	边线宽度，默认：1

函数

```
setTarget(ellipse)
```

描述:

-
-

设置编辑对象

-
-

参数:

名称	类型	描述
ellipse	Ellipse	圆对象

返回:

类型:

boolean

```
getTarget()
```

描述:

-
-

获取编辑对象

-

-

返回:

编辑对象

类型:

Circle

open()

描述:

-
-

开启编辑功能

-
-

close()

描述:

-
-

关闭编辑功能

-
-

事件

start()

描述:

-
-

开始编辑事件

-
-

参数:

名称	类型	描述
ellipse	Ellipse	编辑对象

```
start()
```

描述:

-
-

开始编辑事件

-
-

参数:

名称	类型	描述
ellipse	Ellipse	编辑对象

```
move()
```

描述:

-
-

移动覆盖物时触发此事件

-
-

参数:

名称	类型	描述
ellipse	Ellipse	编辑对象

```
adjust()
```

描述:

-
-

调整折线上某个点时触发此事件

-
-

参数:

名称	类型	描述
ellipse	Ellipse	编辑对象

```
end()
```

描述:

-
-

移动覆盖物时触发此事件

-
-

参数:

名称	类型	描述
ellipse	Ellipse	编辑对象

GeoJSON

事件

change()

描述:

-
-

改变时触发此事件

-
-

参数:

名称	类型	描述
geoJSON	GeoJSON	编辑对象

MassMarks

事件

change()

描述:

-

-

改变时触发此事件

-
-

参数:

名称	类型	描述
<code>massMarks</code>	<code>MassMarks</code>	编辑对象

PolygonEditor

```
new PolygonEditor (map,gon,options)
```

描述:

-
-

编辑 KMap.Polygon 对象。

-
-

示例

```
let circleeditor = new KMap.CircleEditor(map, circle, {controlPointStyle:{fillColor:"#000FFF"}})
let circleeditor = new KMap.CircleEditor(map, circle)
```

参数:

名称	类型	描述
<code>map</code>	<code>Map</code>	地图对象

名称	类型	描述
<code>circle</code>	<code>Circle</code>	要编辑的对象
<code>options</code>	<code>optionsObject</code>	配置参数对象

optionsObject :

名称	类型	描述
<code>style</code>	<code>PolygonStyle</code>	多边形样式表达
<code>controlPointStyle</code>	<code>PolygonStyle</code>	控制点样式
<code>controlPointRadius</code>	<code>Number</code>	控制点半径,单位: 米, 默认: 6
<code>midControlPointStyle</code>	<code>PolygonStyle</code>	中心点样式, <code>PolygonStyle</code> 格式样式
<code>midControlPointRadius</code>	<code>Number</code>	中心点半径,单位: 米, 默认: 6

optionsControlPointStyleObject :

名称	类型	描述
<code>fillColor</code>	<code>String</code>	填充颜色
<code>borderColor</code>	<code>String</code>	边线颜色 eq: #FF00007F, rgba(0, 255, 55, 0.5)
<code>borderWidth</code>	<code>String</code>	边线宽度, 默认: 3

optionsMidControlPointStyleObject :

名称	类型	描述
fillColor	String	填充颜色，默认：#48a8fc
borderColor	String	边线颜色，默认：#ffffff
borderWidth	String	边线宽度，默认：1

函数

```
setTarget(gon)
```

描述:

- -
- 设置编辑对象

-
-

参数:

名称	类型	描述
ellipse	Polygon	圆对象

返回:

类型:
boolean

```
getTarget()
```

描述:

-
-

获取编辑对象

-
-

返回:

编辑对象

类型:

Polygon

open()

描述:

-
-

开启编辑功能

-
-

close()

描述:

-
-

关闭编辑功能

-
-

setAdsorbPolygons(list)

描述:

-
-

设置要吸附的多边形

-
-

示例

```
let polygon = new KMap.Polygon({
  path: [
    new KMap.Point(114.02643481,22.54244415),
    new KMap.Point(114.03742479,22.53455732),
    new KMap.Point(114.02693351,22.52671132),
    new KMap.Point(114.01494302,22.53447281),
    new KMap.Point(114.0167148,22.54194104),
    new KMap.Point(114.02643481,22.54244415)
  ]
});
polygonEditor.setAdsorbPolygons(polygon);
```

参数:

名称	类型	描述
list	Polygon, Array.<Polygon>	多边形对象或多边形对象数组

addAdsorbPolygons(list)

描述:

-
-

添加要吸附的多边形

-
-

示例

```
let polygon = new KMap.Polygon({
  path: [
    new KMap.Point(114.02643481,22.54244415),
    new KMap.Point(114.03742479,22.53455732),
    new KMap.Point(114.02693351,22.52671132),
    new KMap.Point(114.01494302,22.53447281),
    new KMap.Point(114.0167148,22.54194104),
    new KMap.Point(114.02643481,22.54244415)
  ]
});
polygonEditor.addAdsorbPolygons(polygon);
```

参数:

名称	类型	描述
<code>list</code>	<code>Polygon, Array.<Polygon></code>	多边形对象或多边形对象数组

clearAdsorbPolygons()

描述:

-
-

清除吸附的多边形

-
-

示例

```
polygonEditor.clearAdsorbPolygons();
```

removeAdsorbPolygons(list)

描述:

-
-

移除吸附的多边形

-
-

示例

```
let polygon = new KMap.Polygon({
  path: [
    new KMap.Point(114.02643481,22.54244415),
    new KMap.Point(114.03742479,22.53455732),
    new KMap.Point(114.02693351,22.52671132),
    new KMap.Point(114.01494302,22.53447281),
    new KMap.Point(114.0167148,22.54194104),
    new KMap.Point(114.02643481,22.54244415)
  ]
});
polygonEditor.removeAdsorbPolygons(polygon);
```

参数:

名称	类型	描述
list	Polygon, Array.<Polygon>	多边形对象或多边形对象数组

事件

```
start()
```

描述:

-
-

开始编辑事件

-
-

参数:

名称	类型	描述
polygon	Polygon	编辑对象

start()

描述:

-
-

开始编辑事件

-
-

参数:

名称	类型	描述
polygon	Polygon	编辑对象

move()

描述:

-
-

移动覆盖物时触发此事件

-
-

参数:

名称	类型	描述
polygon	Polygon	编辑对象

adjust()

描述:

-
-

调整折线上某个点时触发此事件

-
-

参数:

名称	类型	描述
polygon	Polygon	编辑对象

end()

描述:

-
-

移动覆盖物时触发此事件

-
-

参数:

名称	类型	描述
polygon	Polygon	编辑对象

PolylineEditor

new PolylineEditor (map,line,options)

描述:

-
-

编辑 KMap.Polyline 对象，编辑折点位置，删除/添加折点。

-
-

示例

```
let lineeditor = new KMap.PolylineEditor(map, line, {controlPointStyle:{fillColor:"#000FFF"}})
let lineeditor = new KMap.PolylineEditor(map, line)
```

参数:

名称	类型	描述
map	Map	地图对象
line	Polyline	要编辑的对象
options	optionsObject	配置参数对象

optionsObject :

名称	类型	描述
style	PolylineStyle	线样式表达
controlPointStyle	PolygonStyle	控制点样式
controlPointRadius	Number	控制点半径,单位：米，默认： 6

名称	类型	描述
<code>midControlPointStyle</code>	<code>PolygonStyle</code>	中心点样式, <code>PolygonStyle</code> 格式样式
<code>midControlPointRadius</code>	<code>Number</code>	中心点半径, 单位: 米, 默认: 6

optionsControlPointStyleObject :

名称	类型	描述
<code>fillColor</code>	<code>String</code>	填充颜色
<code>borderColor</code>	<code>String</code>	边线颜色 eq: #FF00007F, rgba(0,255,55,0.5)
<code>borderWidth</code>	<code>String</code>	边线宽度, 默认: 3

optionsMidControlPointStyleObject :

名称	类型	描述
<code>fillColor</code>	<code>String</code>	填充颜色, 默认: #48a8fc
<code>borderColor</code>	<code>String</code>	边线颜色, 默认: #ffffff
<code>borderWidth</code>	<code>String</code>	边线宽度, 默认: 1

函数

setTarget(line)

描述:

-
-

设置编辑对象

-
-

参数:

名称	类型	描述
line	Polyline	编辑对象

返回:

类型:

boolean

```
getTarget()
```

描述:

-
-

获取编辑对象

-
-

返回:

编辑对象

类型:

Polyline

open()

描述:

-
-

开启编辑功能

-
-

close()

描述:

-
-

关闭编辑功能

-
-

事件

start()

描述:

-
-

开始编辑事件

-
-

参数:

名称	类型	描述
polyline	Polyline	编辑对象

start()

描述:

-
-

开始编辑事件

-
-

参数:

名称	类型	描述
polyline	Polyline	编辑对象

move()

描述:

-
-

移动覆盖物时触发此事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
polyline	Polyline	编辑对象

adjust()

描述:

-
-

调整折线上某个点时触发此事件

-
-

参数:

名称	类型	描述
polyline	Polyline	编辑对象

end()

描述:

-
-

移动覆盖物时触发此事件

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
polyline	Polyline	编辑对象

RectangleEditor

```
new RectangleEditor (map,rectangle,options)
```

描述:

-
-

编辑 KMap.Rectangle 对象。

-
-

示例

```
let rectangleEditor = new KMap.RectangleEditor(map, rectangle, {southWestPointStyle:{fillColor:"#000FFF"}})
let rectangleEditor = new KMap.RectangleEditor(map, rectangle)
```

参数:

名称	类型	描述
map	Map	地图对象
circle	Rectangle	要编辑的对象
options	optionsObject	配置参数对象

optionsObject :

名称	类型	描述
----	----	----

名称	类型	描述
<code>style</code>	<code>PolygonStyle</code>	矩形样式表达
<code>southWestPointStyle</code>	<code>PolygonStyle</code>	左下角控制点样式， <code>PolygonStyle</code> 格式样式
<code>southWestPointRadius</code>	<code>Number</code>	控制点半径,单位：米，默认：6
<code>northEastPointStyle</code>	<code>PolygonStyle</code>	右上角控制点样式， <code>PolygonStyle</code> 格式样式
<code>northEastPointRadius</code>	<code>Number</code>	中心点半径,单位：米，默认：6

optionsSouthWestPointStyleObject :

名称	类型	描述
<code>fillColor</code>	<code>String</code>	填充颜色 默认：#ffffff
<code>borderColor</code>	<code>String</code>	边线颜色 默认：#48a8fc
<code>borderWidth</code>	<code>String</code>	边线宽度,默认:2

optionsNorthEastPointStyleObject :

名称	类型	描述
<code>fillColor</code>	<code>String</code>	填充颜色 默认：#ffffff

名称	类型	描述
<code>borderColor</code>	<code>String</code>	边线颜色 默认: <code>#48a8fc</code>
<code>borderWidth</code>	<code>String</code>	边线宽度,默认:2

函数

```
setTarget(rectangle)
```

描述:

-
-
- 设置编辑对象
-
-

参数:

名称	类型	描述
<code>rectangle</code>	<code>Rectangle</code>	编辑对象

返回:

类型:
`boolean`

```
getTarget()
```

描述:

-
-

获取编辑对象

-
-

返回:

编辑对象

类型:
Rectangle

open()

描述:

-
-

开启编辑功能

-
-

close()

描述:

-
-

关闭编辑功能

-
-

事件

start()

描述:

-
-

开始编辑事件

-
-

参数:

名称	类型	描述
rectangle	Rectangle	编辑对象

start()

描述:

-
-

开始编辑事件

-
-

参数:

名称	类型	描述
rectangle	Rectangle	编辑对象

move()

描述:

-

-

移动覆盖物时触发此事件

-
-

参数:

名称	类型	描述
<code>rectangle</code>	<code>Rectangle</code>	编辑对象

`adjust()`

描述:

-
-

调整折线上某个点时触发此事件

-
-

参数:

名称	类型	描述
<code>rectangle</code>	<code>Rectangle</code>	编辑对象

`end()`

描述:

-
-

移动覆盖物时触发此事件

-
-

参数:

名称	类型	描述
rectangle	Rectangle	编辑对象

DrawTool

new DrawTool (props)

描述:

-
-

绘制工具

-
-

示例

```
let drawTool = new KMap.DrawTool();
let drawTool = new KMap.DrawTool({
  style: {
    marker: {
      tmp: () => {
        return {
          radius: 10,
          fillColor: "#43a5fa55",
          borderColor: '#43a5fa',//线条颜色
          borderWidth: '2',//轮廓线宽度
          borderStyle: "solid",
          borderDasharray: [15, 5],    //虚线段
        }
      },
      finish: () => {
        return {
          radius: 10,
          fillColor: "#ffc100",
          borderColor: '#034b06',//线条颜色
          borderWidth: 2,//轮廓线宽度
```

```
        borderStyle: "solid",
        borderDasharray: [15, 5],    //虚线段
    }
},
line: {
    tmp: {
        strokeColor: '#000000', //线条颜色
        strokeWidth: 2, //轮廓线宽度
        strokeStyle: "solid",
        strokeDasharray: [15, 5], //虚线段
    },
    finish: {
        strokeColor: '#000000', //线条颜色
        strokeWidth: 2, //轮廓线宽度
        strokeStyle: "solid",
        strokeDasharray: [15, 5], //虚线段
    }
},
fill: {
    tmp: {
        fillColor: "#43a5fa55",
        borderColor: '#43a5fa', //线条颜色
        borderWidth: '2', //轮廓线宽度
        borderStyle: "solid",
        borderDasharray: [15, 5],    //虚线段
    },
    finish: {
        fillColor: "#fe606155",
        borderColor: '#fe6061', //线条颜色
        borderWidth: 2, //轮廓线宽度
        borderStyle: "solid",
        borderDasharray: [15, 5],    //虚线段
    }
}
});
map.add(drawTool);
```

参数:

名称	类型	描述
props	propsObject	

propsObject :

名称	类型	描述
style	propsStyleObject	绘制表达

propsStyleObject :

名称	类型	描述
marker	propsStyleMarkerObject	绘制点表达
line	propsStyleLineObject	绘制线表达
fill	propsStyleFillObject	绘制面表达

propsStyleMarkerObject :

名称	类型	描述
tmp	propsStyleMarkerTmpObject, function	绘制中点表达
finish	propsStyleMarkerFinishObject, function	绘制完成点表达

propsStyleMarkerTmpObject :

名称	类型	描述
radius	Number	绘制中点半径, 像素
fillColor	String	绘制中点填充色
borderColor	String	绘制中点线条颜色
borderWidth	Number	绘制中点轮廓线宽度
borderStyle	String	绘制中点轮廓线样式

名称	类型	描述
<code>borderDasharray</code>	<code>String</code>	绘制中点虚线段

propsStyleMarkerFinishObject :

名称	类型	描述
<code>radius</code>	<code>Number</code>	绘制完成点半径, 像素
<code>fillColor</code>	<code>String</code>	绘制完成点填充色
<code>borderColor</code>	<code>String</code>	绘制完成点线条颜色
<code>borderWidth</code>	<code>Number</code>	绘制完成点轮廓线宽度
<code>borderStyle</code>	<code>String</code>	绘制完成点轮廓线样式
<code>borderDasharray</code>	<code>String</code>	绘制完成点虚线段

propsStyleLineObject :

名称	类型	描述
<code>tmp</code>	<code>propsStyleLineTmpObject, function</code>	绘制中线表达
<code>finish</code>	<code>propsStyleLineFinishObject, function</code>	绘制完成线表达

propsStyleLineTmpObject :

名称	类型	描述
----	----	----

名称	类型	描述
<code>strokeColor</code>	<code>String</code>	绘制中线条颜色
<code>strokeWidth</code>	<code>Number</code>	绘制中轮廓线宽度
<code>strokeStyle</code>	<code>String</code>	绘制中轮廓线样式
<code>strokeDasharray</code>	<code>String</code>	绘制中虚线段

propsStyleLineFinishObject :

名称	类型	描述
<code>strokeColor</code>	<code>String</code>	绘制完成线条颜色
<code>strokeWidth</code>	<code>Number</code>	绘制完成轮廓线宽度
<code>strokeStyle</code>	<code>String</code>	绘制完成轮廓线样式
<code>strokeDasharray</code>	<code>String</code>	绘制完成虚线段

propsStyleFillObject :

名称	类型	描述
<code>tmp</code>	<code>propsStyleFillTmpObject, function</code>	绘制中面表达
<code>finish</code>	<code>propsStyleFillFinishObject, function</code>	绘制完成线表达

propsStyleFillTmpObject :

名称	类型	描述
<code>fillColor</code>	<code>String</code>	绘制中面填充色
<code>strokeColor</code>	<code>String</code>	绘制中线条颜色
<code>strokeWidth</code>	<code>Number</code>	绘制中轮廓线宽度
<code>strokeStyle</code>	<code>String</code>	绘制中轮廓线样式
<code>strokeDasharray</code>	<code>String</code>	绘制中虚线段

propsStyleFillFinishObject :

名称	类型	描述
<code>fillColor</code>	<code>String</code>	绘制完成面填充色
<code>strokeColor</code>	<code>String</code>	绘制完成线条颜色
<code>strokeWidth</code>	<code>Number</code>	绘制完成轮廓线宽度
<code>strokeStyle</code>	<code>String</code>	绘制完成轮廓线样式
<code>strokeDasharray</code>	<code>String</code>	绘制完成虚线段

函数

`clear`(code)

描述:

-
-

依据编码，销毁绘制结果

-
-

示例

```
drawTool.clear('abcdesfs')
```

参数:

名称	类型	描述
code	String	绘制编码

```
clearAll()
```

描述:

-
-

清空所有结果

-
-

示例

```
drawTool.clearAll()
```

```
turnOn(t)
```

描述:

-
-

开启绘制 当传入对象时，完成后使用对象样式，清除时不会清除传入对象

-
-

示例

```
drawTool.turnOn('Polyline');
drawTool.turnOn(new KMap.CircleMarker({radius:5}));
```

参数:

名称	类型	描述
t	String, Polyline, Polygon, Circle, Rectangle, Ellipse, CircleMarker	绘制类型 Point, Polyline, Polygon, Circle, Rectangle, Ellipse

返回:

类型:
string

turnOff(isEmptyResult)

描述:

-
-

关闭绘制

-
-

示例

```
drawTool.turnOff(true);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
isEmptyResult	Boolean	是否清空所有绘制

destroy()

描述:

-
-

销毁

-
-

示例

```
drawTool.destroy();
drawTool = null;
```

事件

start()

描述:

-
-

开始绘制

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

addnode()

描述:

-
-

增加节点

-
-

参数:

名称	类型	描述
e	Object	结果

removenode()

描述:

-
-

移除节点

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

```
cancle()
```

描述:

-
-

取消绘制

-
-

参数:

名称	类型	描述
e	Object	结果

```
end()
```

描述:

-
-

绘制结束

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

```
remove()
```

描述:

-
-

移除绘制结果

-
-

参数:

名称	类型	描述
e	Object	结果

MeasureTool

```
new MeasureTool (props)
```

描述:

-
-

测量工具

-
-

示例

```

let measureTool = new KMap.MeasureTool({});
let measureTool = new KMap.MeasureTool({
  style: {
    marker: {
      start: function (type) {
        if (type === 'Circle') {
          return {
            'icon': measureCircleIcon,
            'offset': null,
            'anchor': 'center',
            'angle': 0
          }
        } else {
          return {
            'icon': measureIcon,
            'offset': null,
            'anchor': 'center',
            'angle': 0,
          }
        }
      },
      middle: {
        'icon': measureIcon, 'offset': null, 'anchor': 'center',
        'angle': 0, 'zooms': [2, 23]
      },
      end: {
        'icon': measureIcon, 'offset': null, 'anchor': 'center',
        'angle': 0, 'zooms': [2, 23]
      },
    },
    label: {
      start: function (type, val) {
        let text = '';
        if (type === 'Line') {
          text = '测距起点';
        } else if (type === Rectangle) {
          text = '矩形起点';
        } else if (type === Circle) {
          text = '圆心';
        } else if (type === Polygon) {
          text = '多边形起点';
        }
        return {
          'text': text, 'offset': [0, 0], 'direction': 'right'
        }
      },
      middle: function (type, val) {
        let text = '';
        if (type === Line) {
          text = val > 100 ? (Math.round(val / 10) /
100).toString() + "Km" : (Math.round(val * 100) / 100).toString() + "m";
        } else if (type === Rectangle) {
          text = '';
        } else if (type === Circle) {
          text = '';
        } else if (type === Polygon) {

```

```

        text = '';
    }
    return {
        'text': text, 'offset': [0, 0], 'direction': 'right'
    }
},
end: function (type, total, val) {
    let text = '';
    if (type === Line) {
        text = total > 100 ? "总长" + (Math.round(total / 10)
/ 100).toString() + "Km" : "共" + (Math.round(total * 100) /
100).toString() + "m";
    } else if (type === Rectangle || type === Circle || type
=== Polygon) {
        text = total > 10000 ? "面积" + (Math.round(total /
10000) / 100).toString() + "平方千米" : "面积" + (Math.round(total * 100)
/ 100).toString() + "平方米";
    } else {
        text = '';
    }
    return {
        'text': text, 'offset': [0, 0], 'direction': 'right'
    }
}
},
line: {
    tmp: {
        strokeColor: '#43a5fa',//线条颜色
        strokeWidth: 2,//轮廓线宽度
        strokeStyle: "solid",
        strokeDasharray: [15, 5],    //虚线段
    },
    finish: {
        strokeColor: '#fe6061',//线条颜色
        strokeWidth: 2,//轮廓线宽度
        strokeStyle: "solid",
        strokeDasharray: [15, 5],    //虚线段
    }
},
fill: {
    tmp: {
        fillColor: "#43a5fa55",
        borderColor: '#43a5fa',//线条颜色
        borderWidth: '2',//轮廓线宽度
        borderStyle: "solid",
        borderDasharray: [15, 5],    //虚线段
    },
    finish: function (type, val) {
        let fillColor = '';
        if (type === Rectangle) {
            fillColor = "rgba(255,132,0,0.33)";
        } else if (type === Circle) {
            fillColor = "rgba(67,255,0,0.33)";
        } else if (type === Polygon) {
            fillColor = "rgba(255,0,30,0.33)";
        }
    }
}

```

```

    }
    return {
      fillColor: fillColor,
      borderColor: '#fe6061',//线条颜色
      borderWidth: 2,//轮廓线宽度
      borderStyle: "solid",
      borderDasharray: [15, 5],    //虚线段
    }
  }
}
});
map.add(measureTool);

```

参数:

名称	类型	描述
<code>props</code>	<code>propsObject</code>	

propsObject :

名称	类型	描述
<code>style</code>	<code>propsStyleObject</code>	表达

propsStyleObject :

名称	类型	描述
<code>marker</code>	<code>propsStyleMarkerObject</code>	标记表达
<code>label</code>	<code>propsStyleLabelObject</code>	标注表达
<code>line</code>	<code>propsStyleLineObject</code>	线表达
<code>fill</code>	<code>propsStyleFillObject</code>	面表达

propsStyleMarkerObject :

名称	类型	描述
----	----	----

名称	类型	描述
<code>start</code>	<code>propsStyleMarkerStartObject, function</code>	起点标记表达
<code>middle</code>	<code>propsStyleMarkerMiddleObject, function</code>	中间点标记表达
<code>end</code>	<code>propsStyleMarkerEndObject, function</code>	结束点标记表达

propsStyleMarkerStartObject :

名称	类型	描述
<code>icon</code>	<code>Icon</code>	起点标记图标
<code>offset</code>	<code>Pixel</code>	起点标记偏移量
<code>anchor</code>	<code>String</code>	起点标记锚点
<code>angle</code>	<code>Number</code>	起点标记旋转角度
<code>zooms</code>	<code>Vector2</code>	起点标记缩放层级范围

propsStyleMarkerMiddleObject :

名称	类型	描述
<code>icon</code>	<code>Icon</code>	中间点标记图标
<code>offset</code>	<code>Pixel</code>	中间点标记偏移量

名称	类型	描述
<code>anchor</code>	<code>String</code>	中间点标记锚点
<code>angle</code>	<code>Number</code>	中间点标记旋转角度
<code>zooms</code>	<code>Vector2</code>	中间点标记缩放层级范围

propsStyleMarkerEndObject :

名称	类型	描述
<code>icon</code>	<code>Icon</code>	结束点标记图标
<code>offset</code>	<code>Pixel</code>	结束点标记偏移量
<code>anchor</code>	<code>String</code>	结束点标记锚点
<code>angle</code>	<code>Number</code>	结束点标记旋转角度
<code>zooms</code>	<code>Vector2</code>	结束点标记缩放层级范围

propsStyleLabelObject :

名称	类型	描述
<code>start</code>	<code>propsStyleLabelStartObject, function</code>	起点标注表达
<code>middle</code>	<code>propsStyleLabelMiddleObject, function</code>	中间点标注表达

名称	类型	描述
<code>middle</code>	<code>propsStyleLabelMiddleObject, function</code>	结束点标注表达

propsStyleLabelStartObject :

名称	类型	描述
<code>text</code>	<code>String</code>	起点标注文本
<code>offset</code>	<code>Vector2</code>	起点标注偏移量
<code>direction</code>	<code>String</code>	起点标注方位

propsStyleLabelMiddleObject :

名称	类型	描述
<code>text</code>	<code>String</code>	中间点标注文本
<code>offset</code>	<code>Vector2</code>	中间点标注偏移量
<code>direction</code>	<code>String</code>	中间点标注方位
<code>text</code>	<code>String</code>	结束点标注文本
<code>offset</code>	<code>Vector2</code>	结束点标注偏移量
<code>direction</code>	<code>String</code>	结束点标注方位

propsStyleLineObject :

名称	类型	描述
tmp	propsStyleLineTmpObject, function	测量中线表达
tmp	propsStyleLineTmpObject, function	测量结束线表达

propsStyleLineTmpObject :

名称	类型	描述
strokeColor	String	测量中线条颜色
strokeWidth	String	测量中轮廓线宽度
strokeStyle	String	测量中轮廓线样式
strokeDasharray	Vector2	测量中虚线段

propsStyleLineFinishObject :

名称	类型	描述
strokeColor	String	测量结束线条颜色
strokeWidth	String	测量结束轮廓线宽度
strokeStyle	String	测量结束轮廓线样式
strokeDasharray	Vector2	测量结束虚线段

propsStyleFillObject :

名称	类型	描述
tmp	propsStyleFillTmpObject, function	测量中面表达
finish	propsStyleFillFinishObject, function	测量结束面表达

propsStyleFillTmpObject :

名称	类型	描述
fillColor	String	测量中面填充色
borderColor	String	测量中线颜色
borderWidth	String	测量中轮廓线宽度
borderStyle	String	测量中轮廓线样式
borderDasharray	String	测量中虚线段

propsStyleFillFinishObject :

名称	类型	描述
fillColor	String	测量结束面填充色
borderColor	String	测量结束线颜色
borderWidth	String	测量结束轮廓线宽度

名称	类型	描述
<code>borderStyle</code>	<code>String</code>	测量结束轮廓线样式
<code>borderDasharray</code>	<code>String</code>	测量结束虚线段

函数

```
clear(code)
```

描述:

-
-
- 销毁依据编码，销毁测量结果
-
-

示例

```
measureTool.clear('abcdesf');
```

参数:

名称	类型	描述
<code>code</code>	<code>String</code>	测量编码

```
clearAll()
```

描述:

-
-
- 清空所有结果

-
-

示例

```
measureTool.clearAll();
```

turnOn(t)

描述:

-
-

开启测量，返回当前测量 ID

-
-

示例

```
measureTool.turnOn('Polyline');
```

参数:

名称	类型	描述
----	----	----

t	String	测量类型 Polyline, Polygon, Circle, Rectangle
---	--------	---

返回:

类型:

Boolean

turnOff(isEmptyResult)

描述:

-
-

关闭测量

-

-

示例

```
measureTool.turnOff(true);
```

参数:

名称	类型	描述
isEmptyResult	Boolean	是否清空所有测量

```
destroy()
```

描述:

-
-

销毁

-
-

示例

```
measureTool.destroy();  
measureTool = null;
```

事件

```
start()
```

描述:

-
-

开始测量

-
-

参数:

名称	类型	描述
e	Object	结果

```
addnode()
```

描述:

-
-

增加测量

-
-

参数:

名称	类型	描述
e	Object	结果

```
removenode()
```

描述:

-
-

移除测量

-
-

参数:

名称	类型	描述
e	Object	结果

```
cancle()
```

描述:

-
-

取消测量

-
-

参数:

名称	类型	描述
e	Object	结果

```
end()
```

描述:

-
-

测量结束

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
e	Object	结果

```
remove()
```

描述:

-
-

移除测量结果

-
-

参数:

名称	类型	描述
e	Object	结果

数据源

DataSource

```
new DataSource (options)
```

描述:

-
-

GeoJson 数据源

-

-

示例

```
let dataSource = new KMap.DataSource({
  url :
  'http://192.168.0.1/api/?layer=building&minx={left}&miny={bottom}&maxx=
  {right}&maxy={top}',
  coordinates : 'GCJ02',
  tileSize : 512,
});
```

参数:

名称	类型	描述
options	optionsObject	

optionsObject :

名称	类型	描述
id	String	ID
url	String	GeoJSON 数据地址
coordinates	String	数据坐标系，默认 gcj02
tileSize	Number	数据每次获取范围大小，单位像素，默认 1024

函数

setUrl(url)

描述:

-

-

设置 url 模板，范围为 {b}, {left}, {bottom}, {right}, {top}

-
-

示例

```
dataSource.setUrl('http://192.168.0.1/api/?layer=building&box={b}');
```

参数:

名称	类型	描述
url	string	url 地址

getUrl()

描述:

-
-

获取 url 地址

-
-

示例

```
dataSource.getUrl();
```

返回:

url 地址

类型:

string

setCoordinates(coordinates)

描述:

-
-

设置坐标系 (WGS84 | GCJ02)

-
-

示例

```
dataSource.setCoordinates('GCJ02');
```

参数:

名称	类型	描述
coordinates	string	坐标系

getCoordinates()

描述:

-
-

获取坐标系

-
-

示例

```
dataSource.getCoordinates();
```

返回:

坐标系

类型:

string

getTileSize()

描述:

-

-

获取瓦片大小

-
-

示例

```
dataSource.getTileSize();
```

返回:

瓦片大小，单位像素

类型:

`number`

setSize(tileSize)

描述:

-
-

设置瓦片大小

-
-

示例

```
dataSource.setTileSize(1024);
```

参数:

名称

类型

描述

`tileSize`

`number`

瓦片大小，单位像素

属性

id

描述:

-
-

设置 ID

-
-

示例

```
source.id = 'sz';
```

OGCSource

new OGCSource (*options*)

描述:

-
-

OGC 标准数据源

-
-

示例

```
let ogcSource = new KMap.OGCSource({
  service : OGCSERVICE.WMTS,
  url : 'http://{sub}.tianditu.gov.cn/img_w/wmts?tk=key',
  subdomains : ['p0', 'p1', 'p2', 'p3'],
  params : {
    layer: 'img',
    style: 'default',
    tilematrixset : 'w',
    format : 'image/png',
  },
  coordinates : 'wgs84'
});
```

参数:

名称	类型	描述
<code>options</code>	<code>optionsObject</code>	

optionsObject :

名称	类型	描述
<code>id</code>	<code>String</code>	ID
<code>service</code>	<code>OGCService</code>	数据类型，WMTS WMS
<code>url</code>	<code>String</code>	服务地址
<code>subdomains</code>	<code>Array.<String></code>	子域名数组，替换服务地址中的{sub}
<code>params</code>	<code>Object</code>	服务请求参数串，和 <code>queryString</code> 互斥
<code>coordinates</code>	<code>String</code>	数据坐标系，默认 gcj02
<code>queryString</code>	<code>String</code>	自定义请求参数，服务请为 <code>url+queryString</code> , 不设置则使用 <code>params</code> 参数

函数

`getQueryString()`

描述:

-
-

获取 QueryString 模板

-
-

示例

```
ogcSource.getQueryString();
```

返回:

参数模板

类型:
String

```
setQueryString(queryString)
```

描述:

-
-

设置 QueryString 模板 TileRow={y}&TileCol={x}&TileMatriX={z}&bbox={b}

-
-

示例

```
ogcSource.setQueryString('service=wms&request=GetMap&transparent=true&w  
idth=256&height=256&version=1.1.1&srs=EPSG:3857&styles=&format=image/pn  
g&layers=TOPO-WMS,OSM-Overlay-WMS&bbox={b}');
```

参数:

名称	类型	描述
queryString	String	参数模板

```
setUrl(url)
```

描述:

-
-

设置 url 格式化地址

-
-

示例

```
ogcSource.setUrl('https://ows.mundialis.de/services/service');
```

参数:

名称	类型	描述
<code>url</code>	<code>string</code>	url 地址

`setSubdomains(subdomains)`

描述:

-
-

设置子域名, 域名里带上 {sub} 可替换子域名

-
-

示例

```
ogcSource.setSubdomains(['t1','t2']);
```

参数:

名称	类型	描述
<code>subdomains</code>	<code>Array.<String></code>	子域名数组

`setCoordinates(coordinates)`

描述:

-
-

设置坐标系 (WGS84 | GCJ02)

-
-

示例

```
ogcSource.setCoordinates('WGS84');
```

参数:

名称	类型	描述
coordinates	String	坐标系

```
getCoordinates()
```

描述:

-
-

获取坐标系

-
-

示例

```
ogcSource.getCoordinates();
```

返回:

坐标系

类型:

String

属性

id

描述:

-
-

设置 ID

-
-

示例

```
source.id = 'sz';
```

通用库

SystemIcon

函数

getTrackIcon(type)

描述:

-
-

获取系统轨迹点图标表达(正北方向)

-
-

示例

```
let icon = KMap.SystemIcon.getTrackIcon();
```

参数:

名称	类型	描述
type	number	类型

typeObject :

名称	类型	描述
0	number	灰色
1	number	红色
2	number	橙色
3	number	黄色
4	number	绿色

返回:

图标对象

类型:

Icon

```
getOriginIcon()
```

描述:

-
-

获取系统起点 Icon

-

-

示例

```
let icon = KMap.SystemIcon.getOriginIcon();
```

返回:

图标对象

类型:

Icon

getDestinationIcon()

描述:

-
-

获取系统终点 Icon

-
-

示例

```
let icon = KMap.SystemIcon.getDestinationIcon();
```

返回:

图标对象

类型:

Icon

getViaIcon()

描述:

-
-

获取系统途经点 Icon

-
-

示例

```
let icon = KMap.SystemIcon.getViaIcon();
```

返回:

图标对象

类型:

Icon

getPoiMarkerIcon(*color*)

描述:

-
-

获取系统 PoiMarkerIcon

-
-

示例

```
let icon = KMap.SystemIcon.getPoiMarkerIcon();
```

参数:

名称	类型	描述
color	<i>string</i>	颜色，默认蓝色

colorObject :

名称	类型	描述
blue	<i>string</i>	蓝色图标

名称	类型	描述
red	string	红色图标

返回:

图标对象

类型:

Icon

getNumberPoiMarkerIcon(color,number)

描述:

-
-

获取系统数字 PoiMarkerIcon

-
-

示例

```
let icon = KMap.SystemIcon.getNumberPoiMarkerIcon('blue',1);
```

参数:

名称	类型	描述
color	string	颜色，默认蓝色
number	number	数字(1-10)，默认 1

colorObject :

名称	类型	描述
----	----	----

名称	类型	描述
blue	string	蓝色图标
red	string	红色图标

返回:

图标对象

类型:

Icon

getTransitIcon(*type*)

描述:

-
-

获取系统交通 Icon (默认公交图标)

-
-

示例

```
let icon = KMap.SystemIcon.getTransitIcon('metro');
```

参数:

名称	类型	描述
type	string	类型，默认公交

typeObject :

名称	类型	描述
----	----	----

名称	类型	描述
<code>bus</code>	<code>string</code>	公交
<code>metro</code>	<code>string</code>	地铁
<code>walk</code>	<code>string</code>	步行
<code>car</code>	<code>string</code>	汽车
<code>taxi</code>	<code>string</code>	出租车

返回:

图标对象

类型:

`Icon`

`getCircle(color)`

描述:

-
-

获取圆点图标(默认绿色)

-
-

示例

```
let icon = KMap.SystemIcon.getCircle('green');
```

参数:

名称	类型	描述
color	string	颜色，默认蓝色

colorObject :

名称	类型	描述
green	string	绿色图标
red	string	红色图标
blue	string	蓝色图标

返回:

图标对象

类型:
Icon

Browser

属性

ua

描述:

-
-

获取浏览器 userAgent

-
-

示例

Browser.ua

platform

描述:

-
-

获取平台类型

-
-

示例

Browser.platform

mobile

描述:

-
-

判断是否是移动端

-
-

示例

Browser.mobile

plat

描述:

-
-

获取判断平台类型

-

-

示例

Browser.plat

windows

描述:

-
-

是否 windows 设备

-
-

示例

Browser.windows

ios

描述:

-
-

是否 iOS 设备

-
-

示例

Browser.ios

iPad

描述:

-
-

是否 iPad

-
-

示例

Browser.iPad

iPhone

描述:

-
-

是否 iPhone

-
-

示例

Browser.iPhone

android

描述:

-
-

是否 安卓设备

-
-

示例

Browser.android

android23

描述:

-
-

是否安卓 4 以下系统

-
-

示例

Browser.android23

chrome

描述:

-
-

是否 Chrome 浏览器

-
-

示例

Browser.chrome

firefox

描述:

-
-

是否 firefox 浏览器

-
-

示例

Browser.firefox

safari

描述:

-

-

是否 safari 浏览器

-

-

示例

Browser.safari

opera

描述:

-

-

是否 opera 浏览器

-

-

示例

Browser.opera

opera3d

描述:

-

-

是否支持 Css3D 的 opera 浏览器

-

-

示例

Browser.opera3d

wechat

描述:

-
-

是否微信浏览器

-
-

示例

Browser.wechat

uc

描述:

-
-

是否 uc 浏览器

-
-

示例

Browser.uc

qq

描述:

-
-

是否 qq 浏览器

-
-

示例

Browser.qq

ie

描述:

-
-

是否 ie 浏览器

-
-

示例

Browser.ie

ie3d

描述:

-
-

是否支持 Css3D 的 ie 浏览器

-
-

示例

Browser.ie3d

ie6

描述:

-
-

是否 IE6 浏览器

-
-

示例

Browser.ie6

ie7

描述:

-
-

是否 IE7 浏览器

-
-

示例

Browser.ie7

ie8

描述:

-
-

是否 IE8 浏览器

-
-

示例

Browser.ie8

ie9

描述:

-
-

是否 IE9 浏览器

-
-

示例

Browser.ie9

ie10

描述:

-
-

是否 IE10 浏览器

-
-

示例

Browser.ie10

ie11

描述:

-
-

是否 IE11 浏览器

-
-

示例

Browser.ie11

isLocalStorage

描述:

-
-

是否支持 LocaStorage

-
-

示例

```
Browser.isLocalStorage
```

isGeolocation

描述:

-
-

是否支持 Geolocation

-
-

示例

```
Browser.isGeolocation
```

edge

描述:

-
-

是否 edge 浏览器

-
-

示例

```
Browser.edge
```

webkit

描述:

-
-

是否 webkit 内核

-

-

示例

Browser.webkit

webkit3d

描述:

-
-

是否支持 Css3D 的 Webkit 浏览器

-
-

示例

Browser.webkit3d

mobileWebkit

描述:

-
-

是否 webkit 内核

-
-

示例

Browser.mobileWebkit

mobileWebkit3d

描述:

-
-

是否支持 Css3D 的 Webkit 移动端浏览器

-
-

示例

Browser.mobileWebkit3d

gecko

描述:

-
-

gecko 内核

-
-

示例

Browser.gecko

gecko3d

描述:

-
-

是否支持 Css3D 的 gecko 浏览器

-
-

示例

Browser.gecko3d

any3d

描述:

-
-

是否支持 Css3D 的浏览器

-
-

示例

```
Browser.any3d
```

```
retina
```

描述:

-
-

是否高清屏幕，devicePixelRatio>1

-
-

示例

```
Browser.retina
```

```
touch
```

描述:

-
-

是否触屏

-
-

示例

```
Browser.touch
```

```
msPointer
```

描述:

-

-

是否 msPointer 设备

-
-

示例

Browser.msPointer

pointer

描述:

-
-

是否 pointer 设备

-
-

示例

Browser.pointer

isCanvas

描述:

-
-

是否支持 canvas

-
-

示例

Browser.isCanvas

isSvg

描述:

-
-

是否支持 svg

-
-

示例

Browser.isSvg

isWorker

描述:

-
-

是否支持 WebWorker

-
-

示例

Browser.isWorker

isWebsocket

描述:

-
-

是否支持 Websocket

-
-

示例

Browser.isWebsocket

isWebGL

描述:

-
-

是否支持 WebGL

-
-

示例

```
Browser.isWebGL
```

devicePixelRatio

描述:

-
-

设备像素比

-
-

示例

```
Browser.devicePixelRatio
```

Coord

函数

```
gcj02ToKld(lng,lat)
```

描述:

-
-

火星坐标系 (GCJ02) 转凯立德坐标

-

-

示例

```
let kld = Coord.gcj02ToKld(113.92660693,22.50458248);
```

参数:

名称	类型	描述
lng	Number	经度
lat	Number	纬度

返回:

结果

类型:

```
Array.<Number>
```

```
kldToGcj02(lng,lat)
```

描述:

-
-

凯立德坐标转火星坐标系 (GCJ02)

-
-

示例

```
let gcj = Coord.kldToGcj02(419364916, 143908009);
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
lng	Number	经度
lat	Number	纬度

返回:

结果

类型:
Array.<Number>

```
codeToGcj02(str)
```

描述:

-
-
- K 码转换为火星坐标系 (GCJ02)
-
-

示例

```
let gcj = Coord.codeToGcj02('8ju3fc41q');
```

参数:

名称	类型	描述
str	String	K 码

返回:

转换结果

类型：
Array.<Number>,undefined

bd09ToGcj02(*bd_lng*,*bd_lat*)

描述:

-
-

百度坐标系转火星坐标系 (GCJ02)

-
-

示例

```
let gcj = Coord.bd09ToGcj02(116.43014,39.990378);
```

参数:

名称	类型	描述
bd_lng	Number	经度
bd_lat	Number	纬度

返回:

类型：
Array.<Number>

Easing

函数

easeIn(*x*)

描述:

-

-

从 0 开始加速的缓动

-

-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

`easeOut(x)`

描述:

-

-

减速到 0 的缓动

-

-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

easeInAndOut(x)

描述:

-
-

前半段从 0 开始加速，后半段减速到 0 的缓动

-
-

参数:

名称	类型	描述
----	----	----

x	number	x 表示 0（动画开始）到 1（动画结束）范围内的值
---	--------	----------------------------

返回:

类型:

number

easeInSine(x)

描述:

-
-

从 0 开始加速的正弦曲线缓动

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
----	----	----

x	<code>number</code>	x 表示 0 （动画开始）到 1 （动画结束）范围内的值
----------	---------------------	---

返回:

类型:

`number`

easeOutSine(x)

描述:

-
-

减速到 0 的正弦曲线缓动

-
-

参数:

名称	类型	描述
----	----	----

x	<code>number</code>	x 表示 0 （动画开始）到 1 （动画结束）范围内的值
----------	---------------------	---

返回:

类型:

`number`

easeInOutSine(x)

描述:

-
-

半段从 0 开始加速，后半段减速到 0 的正弦曲线缓动

-
-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

`easeInQuad(x)`

描述:

-
-

从 0 开始加速的二次方的缓动

-
-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

`easeOutQuad(x)`

描述:

-
-

减速到 0 的二次方的缓动

-
-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

`easeInOutQuad(x)`

描述:

-
-

前半段从 0 开始加速，后半段减速到 0 的二次方缓动

-
-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

easeInCubic(x)

描述:

-
-

从 0 开始加速的三次方缓动

-
-

参数:

名称	类型	描述
----	----	----

x	number	x 表示 0（动画开始）到 1（动画结束）范围内的值
---	--------	----------------------------

返回:

类型:

number

easeOutCubic(x)

描述:

-
-

减速到 0 的缓动的三次方缓动

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
----	----	----

x	number	x 表示 0 （动画开始）到 1 （动画结束）范围内的值
----------	---------------	---

返回:

类型:

number

easeInOutCubic(x)

描述:

-
-

前半段从 0 开始加速，后半段减速到 0 的三次方缓动

-
-

参数:

名称	类型	描述
----	----	----

x	number	x 表示 0 （动画开始）到 1 （动画结束）范围内的值
----------	---------------	---

返回:

类型:

number

easeInQuart(x)

描述:

-
-

从 0 开始加速的四次方的缓动

-
-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

`easeOutQuart(x)`

描述:

-
-

减速到 0 的四次方的缓动

-
-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

`easeInOutQuart(x)`

描述:

-
-

前半段从 0 开始加速，后半段减速到 0 的四次方的缓动

-
-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

`easeInQuint(x)`

描述:

-
-

从 0 开始加速的五次方的缓动

-
-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

easeOutQuint(x)

描述:

-
-

减速到 0 的五次方的缓动

-
-

参数:

名称	类型	描述
----	----	----

x	number	x 表示 0（动画开始）到 1（动画结束）范围内的值
---	--------	----------------------------

返回:

类型:

number

easeInOutQuint(x)

描述:

-
-

前半段从 0 开始加速，后半段减速到 0 的五次方的缓动

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
----	----	----

x	number	x 表示 0 （动画开始）到 1 （动画结束）范围内的值
----------	---------------	---

返回:

类型:

number

easeInExpo(x)

描述:

-
-

从 0 开始加速的指数曲线的缓动

-
-

参数:

名称	类型	描述
----	----	----

x	number	x 表示 0 （动画开始）到 1 （动画结束）范围内的值
----------	---------------	---

返回:

类型:

number

easeOutExpo(x)

描述:

-
-

减速到 0 的指数曲线的缓动

-
-

参数:

名称	类型	描述
----	----	----

x	number	x 表示 0 （动画开始）到 1 （动画结束）范围内的值
----------	---------------	---

返回:

类型:

number

easeInOutExpo(x)

描述:

-
-

前半段从 0 开始加速，后半段减速到 0 的指数曲线的缓动

-
-

参数:

名称	类型	描述
----	----	----

x	number	x 表示 0 （动画开始）到 1 （动画结束）范围内的值
----------	---------------	---

返回:

类型:

number

easeInCirc(x)

描述:

-
-

从 0 开始加速的圆形曲线的缓动

-
-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

`easeOutCirc(x)`

描述:

-
-

减速到 0 的圆形曲线的缓动

-
-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

easeInOutCirc(x)

描述:

-
-

前半段从 0 开始加速，后半段减速到 0 的圆形曲线的缓动

-
-

参数:

名称	类型	描述
----	----	----

x	number	x 表示 0（动画开始）到 1（动画结束）范围内的值
---	--------	----------------------------

返回:

类型:

number

easeInBack(x)

描述:

-
-

从 0 开始加速的超过范围的三次方缓动

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
----	----	----

x	number	x 表示 0 （动画开始）到 1 （动画结束）范围内的值
----------	---------------	---

返回:

类型:

number

easeOutBack(x)

描述:

-
-

减速到 0 的超过范围的三次方缓动

-
-

参数:

名称	类型	描述
----	----	----

x	number	x 表示 0 （动画开始）到 1 （动画结束）范围内的值
----------	---------------	---

返回:

类型:

number

easeInOutBack(x)

描述:

-
-

前半段从 0 开始加速，后半段减速到 0 的超过范围的三次方缓动

-
-

参数:

名称	类型	描述
----	----	----

x	number	x 表示 0（动画开始）到 1（动画结束）范围内的值
---	--------	----------------------------

返回:

类型:

number

easeInElastic(x)

描述:

-
-

从 0 开始加速的指数衰减的指数衰减的正弦曲线缓动

-
-

参数:

名称	类型	描述
----	----	----

x	number	x 表示 0（动画开始）到 1（动画结束）范围内的值
---	--------	----------------------------

返回:

类型:

number

easeOutElastic(x)

描述:

-
-

减速到 0 的指数衰减的指数衰减的正弦曲线缓动

-
-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

`easeInOutElastic(x)`

描述:

-
-

前半段从 0 开始加速，后半段减速到 0 的指数衰减的正弦曲线缓动

-
-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

easeInBounce(x)

描述:

-
-

从 0 开始加速的指数衰减的反弹缓动

-
-

参数:

名称	类型	描述
----	----	----

x	number	x 表示 0（动画开始）到 1（动画结束）范围内的值
---	--------	----------------------------

返回:

类型:

number

easeOutBounce(x)

描述:

-
-

减速到 0 的指数衰减的反弹缓动

-
-

参数:

名称	类型	描述
----	----	----

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

`easeInOutBounce(x)`

描述:

-
-

前半段从 `0` 开始加速，后半段减速到 `0` 的指数衰减的反弹缓动

-
-

参数:

名称	类型	描述
----	----	----

<code>x</code>	<code>number</code>	<code>x</code> 表示 <code>0</code> （动画开始）到 <code>1</code> （动画结束）范围内的值
----------------	---------------------	---

返回:

类型:

`number`

Geometry

函数

`isClockwise(kPolygon)`

描述:

-
-

取一个环，无论环是顺时针还是逆时针，都返回真或假。

-
-

示例

```
Geometry.isClockwise([new KMap.Point(113.92443976,22.50926081),new KMap.Point(113.93122598,22.51094551),.....])
```

参数:

名称	类型	描述
kPolygon	Array.<Point>	多边形

返回:

类型:
undefined,boolean

```
makeClockwise(kPolygon)
```

描述:

-
-

将一个坐标变为顺时针

-
-

示例

```
Geometry.makeClockwise([new KMap.Point(113.92443976,22.50926081),new KMap.Point(113.93122598,22.51094551),.....])
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
kPolygon	Array.<Point>	多边形

返回:

类型:

Array.<Point>

```
makeAntiClockwise(kPolygon)
```

描述:

-
-

将一个坐标变为逆时针

-
-

示例

```
Geometry.makeAntiClockwise([new KMap.Point(113.92443976,22.50926081),new KMap.Point(113.93122598,22.51094551),.....])
```

参数:

名称	类型	描述
kPolygon	Array.<Point>	多边形

返回:

类型:

Array.<Point>

```
distance(kPoint1,kPoint2)
```

描述:

-
-

计算两个经纬度点之间的实际距离。单位：米

-
-

示例

```
Geometry.distance(new KMap.Point(113.92443976,22.50926081),new KMap.Point(113.93122598,22.51094551))
```

参数:

名称	类型	描述
kPoint1	Point	坐标点
kPoint2	Point	坐标点

返回:

类型:

Number

```
bearing(kPoint1,kPoint2)
```

描述:

-
-

获取两个点，找出它们之间的地理方位，即从正北算起的角度(0 度)

-
-

示例

```
Geometry.bearing(new KMap.Point(113.92443976,22.50926081),new KMap.Point(113.93122598,22.51094551))
```

参数:

名称	类型	描述
kPoint1	Point	坐标点
kPoint2	Point	坐标点

返回:

类型:
Number

```
pointToLineDistance(kPoint,kLine)
```

描述:

-
-

计算点到线最短距离。单位：米

-
-

示例

```
Geometry.pointToLineDistance(new KMap.Point(113.92443976,22.50926081),[new KMap.Point(113.93122598,22.51094551),new KMap.Point(113.93872733,22.50720224),.....])
```

参数:

名称	类型	描述
kPoint	Point	坐标点
kLine	Array.<Point>	线

返回:

类型:

Number

isPointInPolygon(kPoint,kPolygon)

描述:

-
-

判断点是否在线合围区域内

-
-

示例

```
Geometry.isPointInPolygon(new KMap.Point(113.92443976,22.50926081),[new  
KMap.Point(113.93122598,22.51094551),new  
KMap.Point(113.93872733,22.50720224),.....,new  
KMap.Point(113.93122598,22.51094551)])
```

参数:

名称	类型	描述
kPoint	Point	坐标点
kPolygon	Array.<Point>	多边形

返回:

类型:

Boolean

isPointOnPolygon(kPoint,kPolygon,tolerance)

描述:

-
-

点在多边形边线上

-
-

示例

```
Geometry.isPointOnPolygon(new KMap.Point(113.92443976,22.50926081),[new KMap.Point(113.93122598,22.51094551),new KMap.Point(113.93872733,22.50720224),.....,new KMap.Point(113.93122598,22.51094551)])
```

参数:

名称	类型	描述
kPoint	Point	坐标点
kPolygon	Array.<Point>	多边形

返回:

类型:
Boolean

isPointOnLine(kPoint,kLine,tolerance)

描述:

-
-

判断点是否在线上

-
-

示例

```
Geometry.isPointOnLine(new KMap.Point(113.92443976,22.50926081),[new KMap.Point(113.93122598,22.51094551),new KMap.Point(113.93872733,22.50720224),.....])
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
kPoint	Point	坐标点
kLine	Array.<Point>	边形
tolerance	Number	容忍度，单位米

返回:

类型:

Boolean

nearestPointOnLine(kPoint,kLine)

描述:

-
-

点到线上最近的点

-
-

示例

```
Geometry.nearestPointOnLine(new KMap.Point(113.92443976,22.50926081),[new KMap.Point(113.93122598,22.51094551),new KMap.Point(113.93872733,22.50720224),.....])
```

参数:

名称	类型	描述
kPoint	Point	坐标点

名称	类型	描述
kLine	Array.<Point>	线

返回:

类型:
Point

```
distanceOfLine(kLine)
```

描述:

-
-
- 计算线长度，单位米
-
-

示例

```
Geometry.distanceOfLine([new KMap.Point(113.93122598,22.51094551),new KMap.Point(113.93872733,22.50720224),.....])
```

参数:

名称	类型	描述
kLine	Array.<Point>	线

返回:

类型:
Number

```
lineIntersect(kLine1,kLine2)
```

描述:

-

-

获取两条线的交点

-

-

示例

```
Geometry.distanceOfLine([new KMap.Point(113.93122598,22.51094551),new KMap.Point(113.93872733,22.50720224),.....],[new KMap.Point(113.92960975,22.51172227),new KMap.Point(113.93872733,22.50720224),.....])
```

参数:

名称	类型	描述
kLine1	Array.<Point>	线
kLine2	Array.<Point>	线

返回:

类型:
Array.<Point>

```
lineOverlap(kLine1,kLine2,tolerance)
```

描述:

-

-

两个路径重叠区域

-

-

示例

```
Geometry.lineOverlap([new KMap.Point(113.93122598,22.51094551),new KMap.Point(113.93872733,22.50720224),.....],[new
```

```
KMap.Point(113.92960975,22.51172227),new  
KMap.Point(113.93872733,22.50720224),.....])
```

参数:

名称	类型	描述
kLine1	Array.<Point>	线
kLine2	Array.<Point>	线
tolerance	Number	容忍度，单位米

返回:

类型:
Array.<Array.<Point>>

```
linesSurround(kLines,isToSegments)
```

描述:

-
-

线合围成面，仅支持获取交点合围

-
-

示例

```
Geometry.linesSurround([new KMap.Point(113.93122598,22.51094551),new  
KMap.Point(113.93872733,22.50720224),.....],[new  
KMap.Point(113.92960975,22.51172227),new  
KMap.Point(113.93872733,22.50720224),.....])
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
kLines	Array.<Array.<Point>>	线集合

返回:

面集合

类型:

Array.<Array.<Point>>

polygonIntersect(kPolygon1,kPolygon2)

描述:

-
-

计算两个多边形相交面

-
-

示例

```
Geometry.polygonIntersect(  
  [new KMap.Point(113.93122598,22.51094551),new  
  KMap.Point(113.93872733,22.50720224),.....,new  
  KMap.Point(113.93122598,22.51094551)],  
  [new KMap.Point(113.92960975,22.51172227),new  
  KMap.Point(113.93872733,22.50720224),.....,new  
  KMap.Point(113.92960975,22.51172227)]  
)
```

参数:

名称	类型	描述
kPolygon1	Array.<Point>	面
kPolygon2	Array.<Point>	面

返回:

面集合

类型:

`Array.<Array.<Point>>`

```
mergePolygon(kPolygon1,kPolygon2,tolerance)
```

描述:

-
-

相交或共边面合并

-
-

示例

```
Geometry.mergePolygon(  
    [new KMap.Point(113.93122598,22.51094551),new  
    KMap.Point(113.93872733,22.50720224),.....,new  
    KMap.Point(113.93122598,22.51094551)],  
    [new KMap.Point(113.92960975,22.51172227),new  
    KMap.Point(113.93872733,22.50720224),.....,new  
    KMap.Point(113.92960975,22.51172227)]  
)
```

参数:

名称	类型	描述
kPolygon1	<code>Array.<Point></code>	面
kPolygon2	<code>Array.<Point></code>	面
tolerance	<code>Number</code>	容忍度，单位米

返回:

类型：
null,Array.<Point>

splitPolygonByLine(kPolygon,kLine)

描述:

-
-

用线分割多边形

-
-

示例

```
Geometry.splitPolygonByLine(  
    [new KMap.Point(113.93122598,22.51094551),new  
    KMap.Point(113.93872733,22.50720224),.....],  
    [new KMap.Point(113.92960975,22.51172227),new  
    KMap.Point(113.93872733,22.50720224),.....,new  
    KMap.Point(113.92960975,22.51172227)]  
)
```

参数:

名称	类型	描述
kPolygon	Array.<Point>	面
kLine	Point	面

返回:

面集合

类型：
Array.<Array.<Point>>

linePolygonIntersect(kLine,kPolygon)

描述:

-
-

线和多边形是否相交, 返回交点

-
-

示例

```
Geometry.linePolygonIntersect(  
    [new KMap.Point(113.93122598,22.51094551),new  
    KMap.Point(113.93872733,22.50720224),.....],  
    [new KMap.Point(113.92960975,22.51172227),new  
    KMap.Point(113.93872733,22.50720224),.....,new  
    KMap.Point(113.92960975,22.51172227)]  
)
```

参数:

名称	类型	描述
kLine	Array.<Point>	线
kPolygon	Array.<Point>	面

返回:

类型:
Array.<Point>

```
acreageOfPolygon(kPolygon)
```

描述:

-
-

获取多边形面积, 返回平方米

-
-

示例

```
Geometry.acreageOfPolygon([new KMap.Point(113.93122598,22.51094551),new KMap.Point(113.93872733,22.50720224),.....,new KMap.Point(113.93122598,22.51094551)]);
```

参数:

名称	类型	描述
kPolygon	Array.<Point>	面

返回:

类型:
Number

```
get2PointCenter(point1,point2)
```

描述:

-
-

获取两个点的中间点

-
-

参数:

名称	类型	描述
point1	Array.<Number>	点
point2	Array.<Number>	点

返回:

点

类型:
`Array.<Number>`

Util

函数

isDom(obj)

描述:

-
-

判断参数是否为 DOM 元素

-
-

示例

```
Util.isDom(el);
```

参数:

名称	类型	描述
el	Any	对象

返回:

类型:
`boolean`

extend(dest,sources)

描述:

-
-

将源对象中的所有属性复制到目标中

-
-

示例

```
Util.extend(obj1,obj2);
```

参数:

名称	类型	描述
dest	Object	对象
sources	Object	对象

返回:

类型:
*

isEmpty(obj)

描述:

-
-

判断一个对象是否为空

-
-

示例

```
Util.isEmpty(obj);
```

参数:

名称	类型	描述
obj	Object	对象

返回:

类型:

boolean

isEmptyObject(obj)

描述:

-
-

判断是否是一个空对象

-
-

示例

Util.isEmptyObject(obj);

参数:

名称	类型	描述
obj	Object	对象

返回:

类型:

boolean

clear(obj)

描述:

-
-

删除对象所有属性

-
-

示例

```
Util.clear(obj);
```

参数:

名称	类型	描述
obj	Object	对象

format(num,digits)

描述:

-
-

保留小数点后 digits 位

-
-

示例

```
Util.format(123.456,1);
```

参数:

名称	类型	描述
num	Number, String	数字
digits	Number	小数点位数

返回:

类型:
number

isArray(obj)

描述:

-
-

判断是否数组

-
-

示例

```
Util.isArray([1,23,4,5]);
```

参数:

名称	类型	描述
obj	Any	对象

```
isNumeric(obj)
```

描述:

-
-

是否是数字字符串

-
-

示例

```
Util.isNumeric("123456");
```

参数:

名称	类型	描述
obj	Any	对象

返回:

类型:

boolean

includes(arr,item)

描述:

-
-

判断数组是否包含某个元素

-
-

示例

```
Util.includes([1,2,3],1);
```

参数:

名称	类型	描述
arr	Array.<Any>	数组
item	Any	元素

返回:

类型:

boolean

indexOf(arr,item)

描述:

-
-

返回元素索引

-
-

示例

```
Util.indexOf([1,2,3],1);
```

参数:

名称	类型	描述
arr	Array.<Any>	数组
item	Any	元素

返回:

类型:
Number

```
deleteItemFromArray(arr,item)
```

描述:

-
-

从数组删除元素

-
-

示例

```
Util.deleteItemFromArray([1,2,3],1);
```

参数:

名称	类型	描述
arr	Array.<Any>	数组

名称	类型	描述
item	Any	元素

```
deleteItemFromArrayByIndex(arr,index)
```

描述:

-
-

按索引删除数组元素

-
-

示例

```
Util.deleteItemFromArrayByIndex([1,2,3],0);
```

参数:

名称	类型	描述
arr	Array	数组
index	Number	索引

```
isURL(str)
```

描述:

-
-

判断是否是 URL

-

-

示例

```
Util.isURL("http://open.careland.com.cn");
```

参数:

名称	类型	描述
str	String	Url

返回:

是/否

类型:

boolean

isHttps()

描述:

-
-

判断当前页面是否为 https 协议

-
-

示例

```
let flag = Util.isHttps();
```

返回:

是/否

类型:

boolean

getCurrentBaseUrl()

描述:

-
-

获取到页面当前绝对地址

-
-

示例

```
Util.getCurrentBaseUrl("https://open.careland.com.cn/docs/new/templates/function/map.html");
```

返回:

绝对地址 Url

类型:

string

parseURL(url)

描述:

-
-

解析 URL

-
-

示例

```
Util.parseURL("https://open.careland.com.cn/docs/new/templates/function/map.html?a=1");
```

参数:

名称	类型	描述
----	----	----

名称	类型	描述
url		

返回:

类型:
Object

```
pointArrayToString(points)
```

描述:

-
-
- 坐标数组格式化为 lon, lat;lon, lat
-
-

示例

```
Util.pointArrayToString([new KMap.Point(113.93122598,22.51094551),new KMap.Point(113.93872733,22.50720224)]);
```

参数:

名称	类型	描述
points	Array.<Point>	

返回:

类型:
String

```
stringToPointArray(str,split)
```

描述:

-
-
- 字符串坐标转为坐标串

-
-

示例

```
Util.stringToPointArray("113.93122598,22.51094551;113.93872733,22.50720224");
```

参数:

名称	类型	描述
str	String	lon,lat;lon,lat
split	String	分隔串，默认为;

返回:

类型:
Array.<Point>

```
stringToPoint(str,split)
```

描述:

-
-

字符坐标转为坐标

-
-

示例

```
Util.stringToPointArray("113.93122598,22.51094551");
```

参数:

名称	类型	描述
str	String	lon,lat

名称	类型	描述
<code>split</code>	<code>String</code>	默认,

返回:

类型:
`Point,null`

`degToRad(degrees)`

描述:

-
-

角度转弧度

-
-

参数:

名称	类型	描述
<code>degrees</code>		

返回:

`Util.degToRad(60);`

类型:
`Number`

`radToDeg(radians)`

描述:

-
-

弧度转角度

-
-

参数:

名称	类型	描述
radians		

返回:

Util.radToDeg(1.26565894);

类型:

Number

公共

BaseEvent

描述:

-
-

事件监听

-
-

Lang

描述:

-
-

语言选项

-
-

属性:

名称	类型	描述
<code>en</code>	<code>string</code>	英文
<code>cn</code>	<code>string</code>	中文

Theme

描述:

-
-

主题选项

-
-

属性:

名称	类型	描述
<code>bright</code>	<code>string</code>	亮色
<code>dark</code>	<code>string</code>	暗色

VehicleType

描述:

-
-

车型选项

-
-

属性:

名称	类型	描述
CAR	number	小车
RV	number	房车
MINIVAN_TRUCK	number	微型货车
LIGHT_TRUCK	number	轻型货车
MEDIUM_TRUCK	number	中型货车
HEAVY_TRUCK	number	重型货车

DrivingPolicy

描述:

-
-

驾车规划策略选项

-
-

属性:

名称	类型	描述
DEFAULT	number	系统推荐
PRIORITY_HIGHWAYS	number	高速优先
LEAST_FEE	number	最少收费
LEAST_TIME	number	最少时间
NO_HIGHWAYS	number	不走高速

TransitPolicy

描述:

-
-

公交规划策略选项

-
-

属性:

名称	类型	描述
DEFAULT	number	系统推荐
LEAST_TIME	number	最少时间

名称	类型	描述
LEAST_TRANSFER	number	最少换乘
LEAST_WALKING	number	最少步行
NO_METRO	number	不乘地铁
PRIORITY_METRO	number	地铁优先

MapViewMode

描述:

-
-

地图模式选项

-
-

属性:

名称	类型	描述
TWO	string	2D
THREE	string	3D

OGCService

描述:

-

-

OGC 服务模式选项

-
-

属性:

名称	类型	描述
<code>WMS</code>	<code>string</code>	<code>wms</code> 服务
<code>WMTS</code>	<code>string</code>	<code>wmts</code> 服务

MeasureType

描述:

-
-

测量类型选项

-
-

属性:

名称	类型	描述
<code>LINE</code>	<code>string</code>	线
<code>POLYGON</code>	<code>string</code>	面
<code>CIRCLE</code>	<code>string</code>	圆

名称	类型	描述
RECTANGLE	string	矩形

DrawType

描述:

-
-

绘制类型选项

-
-

属性:

名称	类型	描述
POINT	string	点
LINE	string	线
POLYGON	string	面
CIRCLE	string	圆
RECTANGLE	string	矩形
ELLIPSE	string	椭圆

Mvt

描述:

-
-

Mvt 解析

-
-

FeatureEdit

描述:

-
-

覆盖物编辑基类

-
-

PointStyle

描述:

-
-

矢量覆盖物点的样式，不是所有使用场景都支持所有属性

-
-

属性:

名称	类型	描述
<code>images</code>	<code>Array.<String></code>	纹理 url 地址数组，

名称	类型	描述
		eq:['url','url1'], 图片宽度必须是 2 的 N 次方
pointColor	String	点颜色 eq:#8af5bd
pointSize	Number	点大小（单位：像素）
textColor	String	文本颜色
textSize	Number	文本大小（单位：磅）
bkColor	String	背景颜色
textStyle	String	文本样式:背景透明、背景填充、描边
dispWidth	Number	显示宽度（一般用于贴地面的交通标志），单位为米
dispHeight	Number	显示宽度（一般用于贴地面的交通标志），单位为米
imageLeft	Number	在 image 中范围（从一张大图中扣一块显示），设置后不支持平铺， images 指向的图片宽度必须是 2 的 N 次方

名称	类型	描述
<code>imageTop</code>	Number	在 <code>image</code> 中范围（从一张大图中扣一块显示），设置后不支持平铺， <code>images</code> 指向的图片宽度必须是 2 的 N 次方
<code>imageRight</code>	Number	在 <code>image</code> 中范围（从一张大图中扣一块显示），设置后不支持平铺， <code>images</code> 指向的图片宽度必须是 2 的 N 次方
<code>imageBottom</code>	Number	在 <code>image</code> 中范围（从一张大图中扣一块显示），设置后不支持平铺， <code>images</code> 指向的图片宽度必须是 2 的 N 次方
<code>width</code>	Number	图片宽度
<code>height</code>	Number	图片高度
<code>posStyle</code>	Number	文本位置 0:中心(只显示图标)，1:中上，2:左中，3:中间（只显示文本），4:右中，5:中下，6:文本+背景图片
<code>bkImageStyle</code>	Number	<code>images</code> 作为文本的背景图片 0:根据文本拉伸，1:根据文本等比拉伸
<code>textLeft</code>	Number	文本显示（在背景图片中的）范围

名称	类型	描述
<code>textTop</code>	<code>Number</code>	文本显示（在背景图片中的）范围
<code>textRight</code>	<code>Number</code>	文本显示（在背景图片中的）范围
<code>textBottom</code>	<code>Number</code>	文本显示（在背景图片中的）范围

PolylineStyle

描述:

-
-

矢量覆盖物线的表达样式，不是所有使用场景都支持所有属性

-
-

属性:

名称	类型	描述
<code>strokeWidth</code>	<code>Number</code>	线宽 eq:8
<code>strokeColor</code>	<code>String</code>	线颜色 eq:#8af5bd
<code>strokeStyle</code>	<code>String</code>	线样式，实线:solid，虚线:dashed， 虚边线:borderDashed
<code>strokeDasharray</code>	<code>Array.<Number></code>	虚线段 eq:[15,5]

名称	类型	描述
<code>borderWidth</code>	<code>Number</code>	边线宽度
<code>borderColor</code>	<code>String</code>	边线颜色
<code>getCanvas</code>	<code>function</code>	画布回调函数，传入一个方法返回值为一个画布 <code>[document.createElement("canvas")]</code>
<code>images</code>	<code>Array.<String></code>	纹理 url 地址数组， <code>eq:['url','url1']</code>
<code>repeatWidth</code>	<code>Number</code>	纹理平铺宽度。 <code>-1</code> :拉伸， <code>0</code> :使用图片宽度平铺（默认），大于 <code>0</code> :指定平铺宽度
<code>repeatHeight</code>	<code>Number</code>	纹理平铺高度。 <code>-1</code> :拉伸（默认）， <code>0</code> :使用图片高度平铺，大于 <code>0</code> :指定平铺高度
<code>startCap</code>	<code>String</code>	开头线帽，默认值为' <code>butt</code> '无头，其他可选值: ' <code>round</code> '圆头, ' <code>triangle</code> '三角箭头, ' <code>square</code> '方头, ' <code>images</code> '图片",

名称	类型	描述
startImages	Array.<String>	开头线帽图片，eq:['url','url1']
endCap	String	结尾线帽，默认值为'butt'无头，其他可选值：'round'圆头，'triangle'三角箭头，'square'方头，'images'图片"，
endImages	Array.<String>	开头线帽图片，eq:['url','url1']
capScale	Number	线帽缩放（相对于线宽）

PolygonStyle

描述:

-
-

矢量覆盖物线的表达样式，不是所有使用场景都支持所有属性

-
-

属性:

名称	类型	描述
fillColor	String	填充 eq:#8af5bd
getCanvas	function	画布回调函数，传入一个方法返回值为一个画布

名称	类型	描述
		<code>[document.createElement("canvas")]</code>
<code>images</code>	<code>Array.<String></code>	纹理 <code>url</code> 地址数组， <code>eq:['url','url1']</code>
<code>imgIdx</code>	<code>Number</code>	非必须，多当前图片时，设置的显示序号
<code>repeatWidth</code>	<code>Number</code>	纹理平铺宽度。 <code>-1</code> :拉伸， <code>0</code> :使用图片宽度平铺（默认），大于 <code>0</code> :指定平铺宽度
<code>repeatHeight</code>	<code>Number</code>	纹理平铺高度。 <code>-1</code> :拉伸（默认）， <code>0</code> :使用图片高度平铺，大于 <code>0</code> :指定平铺高度
<code>timesOfRepS</code>	<code>Array.<String></code>	水平平铺次数
<code>timesOfRepT</code>	<code>Array.<String></code>	垂直平铺次数
<code>borderWidth</code>	<code>Number</code>	边线宽度
<code>borderColor</code>	<code>String</code>	边线颜色

名称	类型	描述
<code>borderImages</code>	<code>Array.<String></code>	边线纹理图片 <code>url</code>
<code>borderRepeatWidth</code>	<code>Number</code>	纹理平铺宽度。 <code>-1</code> :拉伸， <code>0</code> :使用图片宽度平铺（默认），大于 <code>0</code> :指定平铺宽度
<code>borderRepeatHeight</code>	<code>Number</code>	纹理平铺高度。 <code>-1</code> :拉伸（默认）， <code>0</code> :使用图片高度平铺，大于 <code>0</code> :指定平铺高度
<code>borderStyle</code>	<code>String</code>	边线样式，实线: <code>solid</code> ，虚线: <code>dashed</code> ， 虚边线: <code>borderDashed</code>
<code>borderDasharray</code>	<code>Number</code>	虚线段 <code>eq:[15,5]</code>
<code>offsetXStep</code>	<code>Number</code>	单张图的循环动画偏移步长， <code>[0]</code> x 方向偏移
<code>offsetYStep</code>	<code>Number</code>	单张图的循环动画偏移步长， <code>[1]</code> y 方向偏移
<code>interval</code>	<code>Number</code>	播放间隔，单位毫秒，可用于 <code>offsetStep/images</code>
<code>textureAngle</code>	<code>Number</code>	纹理角度（单位：弧度）

名称	类型	描述
isColor	Number	颜色叠加纹理，当纹理有效果时