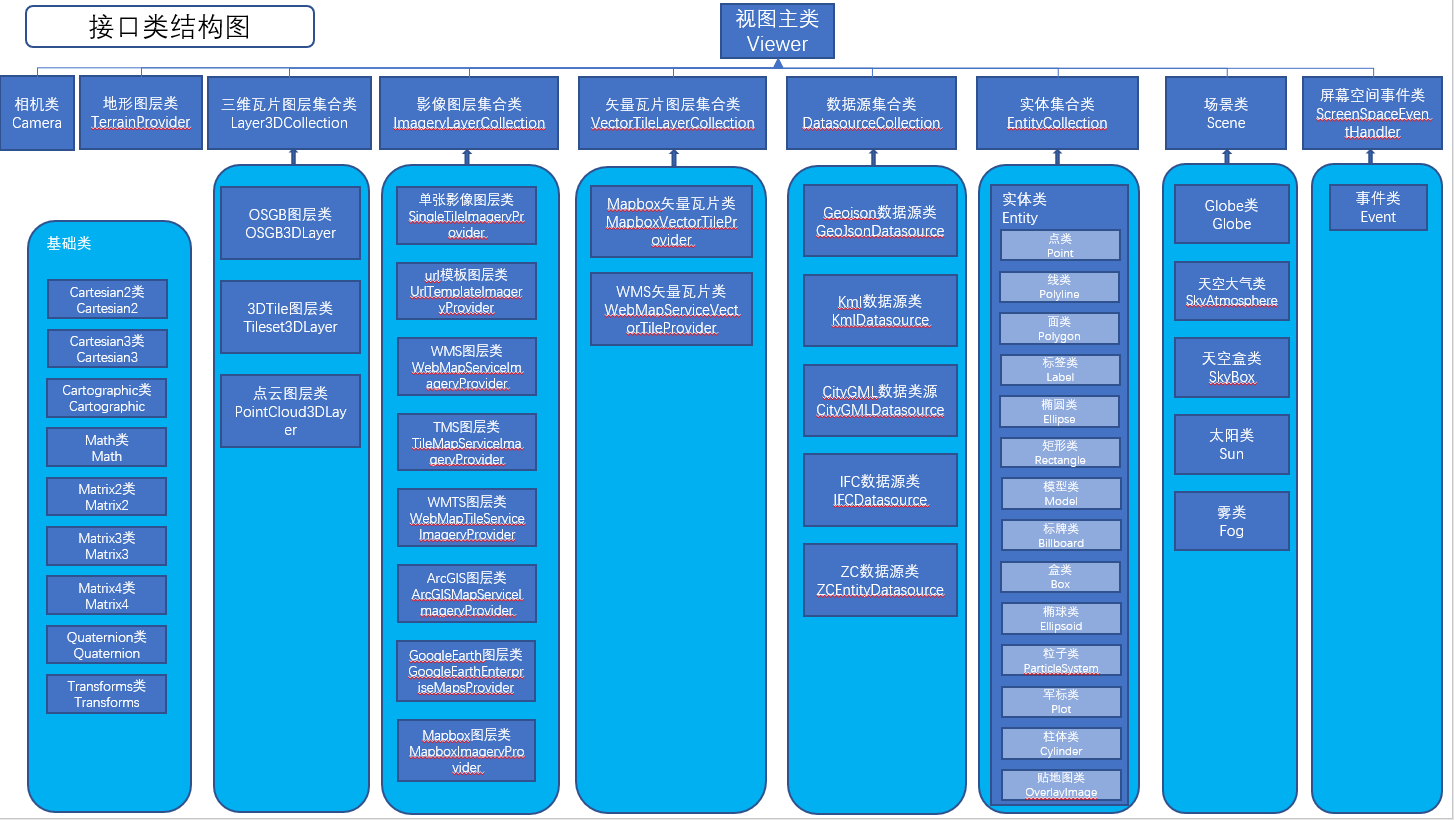
|  |  |
| --- | --- |
|  |  |
| PIE-Map-Web V0.1 | |
|  | |
| 接口文档 | |
|  | |
| 西安航天宏图信息技术有限公司  2023年02月 | |

**文档说明**

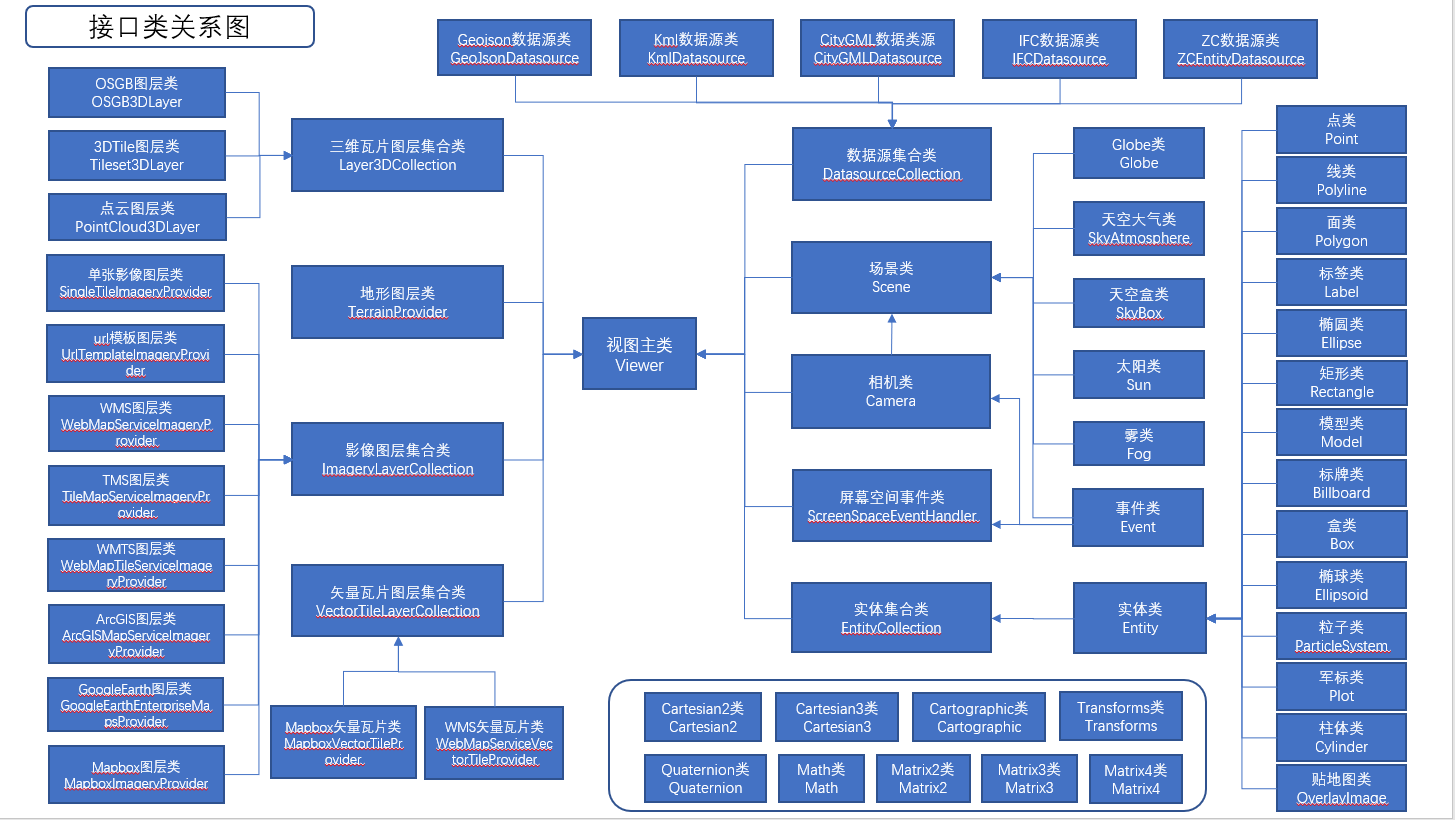
本文档包含所有的内容除说明以外， 版权均属航天宏图信息技术股份有限公司所有， 受《中华人民共和国著作权法》保护及相关法律法规和中国加入的所有知识产权方面的国际条约的保护。未经本公司书面许可，任何单位和个人不得以任何方式翻印和转载本文档的任何内容，否则视为侵权，航天宏图信息技术股份有限公司保留已发追究其法律责任的权利。本文档是PIE-Map Web JavaScriptAPI的用户使用参考手册，详细阐述了API提供用户的类和方法，以及可以实现的功能介绍。PIE-Map Web JavaScriptAPI让您可以将地图嵌入您自己的网页中。API提供了许多方法与地图进行交互，以及一系列向地图添加内容的服务，从而使您可以在自己的网站上创建稳定的地图应用程序。

**JavaScript API 整体概要设计**

**API接口基本结构图**



**API接口类基本关系图**



**目录**

[1 Core主类 6](#_Toc31422)

[1.1 Cartesian2类 6](#_Toc29738)

[1.2 Cartesian3类 14](#_Toc10825)

[1.3 Cartesian4类 25](#_Toc30517)

[1.4 Cartographic类 32](#_Toc32586)

[1.5 Color类 35](#_Toc9325)

[1.6 EllipsoidTerrainProvider类 41](#_Toc7949)

[1.7 Math类 42](#_Toc19076)

[1.8 Matrix3类 48](#_Toc10692)

[1.9 Matrix4类 57](#_Toc3078)

[1.10 ScreenSpaceEventType类 72](#_Toc13929)

[2 DataSources主类 73](#_Toc4153)

[2.1 DatasourceCollection类 73](#_Toc21139)

[2.2 Entity类 74](#_Toc19477)

[2.3 EntityCollection类 75](#_Toc19987)

[2.4 GeoJsonDatasource类 76](#_Toc11257)

[2.5 Model类 77](#_Toc6229)

[2.6 Point类 78](#_Toc27654)

[2.7 Polygon类 79](#_Toc26066)

[2.8 Polyline类 80](#_Toc7236)

[2.9 Label类 81](#_Toc17948)

[2.10 Rectangle类 82](#_Toc30461)

[2.11 Billboard类 83](#_Toc22347)

[2.12 Box类 84](#_Toc8953)

[2.13 Plot类 84](#_Toc27095)

[3 Scene主类 85](#_Toc2168)

[3.1 ArcGISMapServerImageryProvider类 85](#_Toc16408)

[3.2 Camera类 85](#_Toc22301)

[3.3 Fog类 87](#_Toc12046)

[3.4 Globe类 87](#_Toc3120)

[3.5 GoogleEarthEnterpriseMapsProvider类 88](#_Toc21019)

[3.6 ImageryLayerCollection类 88](#_Toc984)

[3.7 Layer3DCollection类 89](#_Toc20947)

[3.8 MapboxImageryProvider类 90](#_Toc13632)

[3.9 MapboxVectorTileProvider类 90](#_Toc32274)

[3.10 OSGB3DLayer类 91](#_Toc30912)

[3.11 Scene类 91](#_Toc15861)

[3.12 SceneMode类 93](#_Toc28808)

[3.13 SingleTileImageryProvider类 94](#_Toc11371)

[3.14 SkyAtmosphere类 94](#_Toc12123)

[3.15 SkyBox类 95](#_Toc11894)

[3.16 Sun类 95](#_Toc22100)

[3.17 TileMapServiceImageryProvider类 96](#_Toc30809)

[3.18 Tileset3Dlayer类 96](#_Toc27102)

[3.19 UrlTemplateImageryProvider类 97](#_Toc24895)

[3.20 VectorTileLayerCollection类 97](#_Toc24377)

[3.21 Viewer类 98](#_Toc8487)

[3.22 WebMapServiceImageryProvider类 99](#_Toc12091)

[3.23 WebMapTileServiceImageryProvider类 100](#_Toc13075)

# Core主类

## Cartesian2类

2D笛卡尔点。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Cartesian2 ( x , y ) | 2D笛卡尔点 | x组件，y分量 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| x | X组件 | Number |
| y | Y分量 | Number |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| clone ( result ) | 复制此Cartesian2实例。 | Result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| equals ( right ) | 将此笛卡尔坐标与提供的笛卡尔分量进行比较，然后返回如果相等，则为 true ，否则为 false 。 | Right：Cartesian2，右侧笛卡尔坐标。 | Boolean |
| equalsEpsilon ( right , relativeEpsilon , absoluteEpsilon ) | 将此笛卡尔坐标与提供的笛卡尔分量进行比较，然后返回 true 如果它们通过绝对或相对公差测试，否则为 false 。 | Right：Cartesian2，右侧笛卡尔坐标；relativeEpsilon：Number，用于相等性测试的相对epsilon公差；absoluteEpsilon：Number，用于相等性测试的绝对epsilon公差。 | Boolean |
| toString () | 创建一个以'（x，y）'格式表示此笛卡尔坐标的字符串。 |  | String |
| abs ( cartesian, result ) | 计算提供的笛卡尔坐标的绝对值。 | Cartesian：Cartesian2，要计算其绝对值的笛卡尔坐标。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| add ( left , right , result ) | 计算两个笛卡尔的按分量求和。 | left：Cartesian2，第一个笛卡尔式。  right：Cartesian2，第二个笛卡尔。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| angleBetween ( left , right ) | 返回所提供的笛卡尔坐标之间的角度（以弧度为单位）。 | left：Cartesian2，第一个笛卡尔式。  right：Cartesian2，第二个笛卡尔。 | Number |
| clone  (cartesian, result ) | 复制一个Cartesian2实例。 | cartesian：Cartesian2，要复制的笛卡尔坐标。  result：Cartesian2，可选，将结果存储到的对象。 | Cartesian2 |
| distance ( left , right ) | 计算两点之间的距离。 | left：Cartesian2，计算距离的第一点。  right：Cartesian2，计算距离的第二点。 | Number |
| distanceSquared ( left , right ) | 计算两点之间的平方距离。比较平方距离使用此功能比使用 Cartesian2＃distance 比较距离更为有效。 | left：Cartesian2，计算距离的第一点。  right：Cartesian2，计算距离的第二点。 | Number |
| divideByScalar (cartesian , scalar , result ) | 将提供的笛卡尔分量除以提供的标量。 | cartesian：Cartesian2，要划分的笛卡尔坐标。  scalar：Cartesian2，要除以的标量。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| divideComponents ( left , right , result ) | 计算两个笛卡尔的分量商。 | left：Cartesian2，第一个笛卡尔式。  right：Cartesian2，第二个笛卡尔。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| dot ( left , right ) | 计算两个笛卡尔的点（标量）乘积。 | left：Cartesian2，第一个笛卡尔式。  right：Cartesian2，第二个笛卡尔。 | Number |
| equals( left , right ) | 比较提供的笛卡尔分量，然后返回如果相等，则为 true ，否则为 false 。 | left：Cartesian2，第一个笛卡尔式。  right：Cartesian2，第二个笛卡尔。 | Boolean |
| equalsEpsilon ( left , right , relativeEpsilon , absoluteEpsilon ) | 比较提供的笛卡尔分量，然后返回 true 如果它们通过绝对或相对公差测试，否则为 false 。 | left：Cartesian2，第一个笛卡尔式。  right：Cartesian2，第二个笛卡尔。  relativeEpsilon：Number，用于相等性测试的相对epsilon公差。  absoluteEpsilon：Number，用于相等性测试的绝对epsilon公差。 | Boolean |
| FromArray (array, startingIndex , result ) | 从数组中的两个连续元素创建Cartesian2。 | array：Array.<Number>，其两个连续元素分别对应于x和y分量的数组。  startingIndex：Number，第一个元素数组的偏移量，它对应于x分量。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| fromCartesian3 (cartesian, result ) | 从现有的Cartesian3创建Cartesian2实例。这只是需要Cartesian3和下降z的x和y属性。 | cartesian：Cartesian3，Cartesian3实例，用于从中创建Cartesian2实例。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| fromCartesian4 (cartesian, result ) | 从现有的Cartesian4创建Cartesian2实例。这只是需要Cartesian4的x和y属性，并丢弃z和w。 | cartesian：Cartesian4，Cartesian4实例，用于从中创建Cartesian2实例。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| fromElements (x, y, result ) | 根据x和y坐标创建Cartesian2实例。 | x：Number，x坐标。  y：Number，y坐标。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| lerp (start, end, t, result) | 使用提供的笛卡尔坐标计算t处的线性插值或外推。 | start：Cartesian2，对应于t的值为0.0。  end：Cartesian2，对应于t的值为0.0。  t：Number，沿t的插值点。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| magnitude (cartesian) | 计算笛卡尔的大小（长度）。 | cartesain：Cartesian2，要计算其大小的笛卡尔实例。 | Number |
| magnitude Squared(cartesian) | 计算提供的笛卡尔平方强度。 | cartesain：Cartesian2，要计算其大小的笛卡尔实例。 | Number |
| maximumByComponent (first, second, result) | 比较两个笛卡尔，并计算一个笛卡尔，其中包含所提供笛卡尔的最大分量。 | first：Cartesian2，笛卡尔进行比较。  second：Cartesian2，笛卡尔进行比较。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| maximumComponent (cartesian) | 计算所提供的直角坐标的最大分量的值。 | cartesain：Cartesian2，要使用的笛卡尔坐标。 | Number |
| minimumByComponent (first, second, result) | 比较两个笛卡尔，并计算一个笛卡尔，其中包含所提供笛卡尔的最小分量。 | first：Cartesian2，笛卡尔进行比较。  second：Cartesian2，笛卡尔进行比较。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| minimumComponent (cartesian) | 计算所提供的直角坐标的最小分量的值。 | cartesain：Cartesian2，要使用的笛卡尔坐标。 | Number |
| mostOrthogonalAxis (cartesian, result) | 返回与提供的笛卡尔坐标最正交的轴。 | cartesain：Cartesian2，要在其上找到最正交轴的笛卡尔坐标。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| multiplyByScalar (cartesian, scalar, result) | 将提供的笛卡尔分量乘以提供的标量。 | cartesian：Cartesian2，第一个笛卡尔式。  scalar：Cartesian2，第二个笛卡尔式。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| multiplyComponents (left, right, result) | 计算两个笛卡尔的分量积。 | left：Cartesian2，第一个笛卡尔式。  right：Cartesian2，第二个笛卡尔式。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| negate (cartesian, result) | 取反提供的笛卡尔坐标。 | cartesian：Cartesian2，笛卡尔要否定。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| normalize (cartesian, result) | 计算提供的笛卡尔坐标系的标准化形式。 | cartesian：Cartesian2，要标准化的笛卡坐标。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| pack (value, array, startingIndex ) | 将提供的实例存储到提供的数组中。 | value：Cartesian2，要打包的值。  array：Array.<Number>，要打包的数组。  startingindex：Number，开始打包元素的数组索引。 | Array.<Number> |
| packArray (array, result ) | 将笛卡尔坐标数组展平为组件数组。 | array：Array.<Number>，要打包的数组。  result：Number，将结果存储到的数组。如果这是类型化的数组，则必须具有array.length \* 2个组件，否则将抛出 [DeveloperError](http://cesium.xin/cesium/cn/Documentation1.72/DeveloperError.html)。如果它是常规数组，则将其大小调整为具有（array.length \* 2）个元素。 | Array.<Number> |
| subtract (left, right, result) | 计算两个笛卡尔的分量差异。 | left：Cartesian2，第一个笛卡尔式。  right：Cartesian2，第二个笛卡尔式。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| unpack (array, startingIndex , result ) | 从压缩数组中检索实例。 | array：Array.<Number>，压缩数组。  startingIndex：Number，要解压缩的元素的起始索引。  result：Cartesian2，将结果存储到的对象。 | Cartesian2 |
| unpackArray (array, result ) | 将笛卡尔组件数组解压缩到笛卡尔数组中。 | array：Array.<Number>，要解包的组件数组。  result：Array.<Cartesian2>，将结果存储到的对象。 | Cartesian2 |

## Cartesian3类

3D笛卡尔点。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Cartesian3 ( x , y , z ) | 3D笛卡尔点 | x组件，y分量，z分量 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| x | X组件 | Number |
| y | Y分量 | Number |
| z | Z分量 | Number |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| clone ( result ) | 复制此Cartesian3实例。 | Result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| equals ( right ) | 将此笛卡尔坐标与提供的笛卡尔分量进行比较，然后返回如果相等，则为 true ，否则为 false 。 | Right：Cartesian3，右侧笛卡尔坐标。 | Boolean |
| equalsEpsilon ( right , relativeEpsilon , absoluteEpsilon ) | 将此笛卡尔坐标与提供的笛卡尔分量进行比较，然后返回 true 如果它们通过绝对或相对公差测试，否则为 false 。 | Right：Cartesian3，右侧笛卡尔坐标；relativeEpsilon：Number，用于相等性测试的相对epsilon公差；absoluteEpsilon：Number，用于相等性测试的绝对epsilon公差。 | Boolean |
| toString () | 创建一个以'（x，y）'格式表示此笛卡尔坐标的字符串。 |  | String |
| abs (cartesian, result) | 计算提供的笛卡尔坐标的绝对值。 | Cartesian：Cartesian3，要计算其绝对值的笛卡尔坐标。  result：Cartesian2，将结果存储到的对象。 | Cartesian3 |
| add ( left , right , result ) | 计算两个笛卡尔的按分量求和。 | left：Cartesian3，第一个笛卡尔式。  right：Cartesian3，第二个笛卡尔。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| angleBetween ( left , right ) | 返回所提供的笛卡尔坐标之间的角度（以弧度为单位）。 | left：Cartesian3，第一个笛卡尔式。  right：Cartesian3，第二个笛卡尔。 | Number |
| clone  (cartesian, result ) | 复制一个Cartesian3实例。 | cartesian：Cartesian3，要复制的笛卡尔坐标。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| cross ( left , right , result ) | 计算两个笛卡尔的叉（外）乘积。 | left：Cartesian3，第一个笛卡尔式。  right：Cartesian3，第二个笛卡尔。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| distance ( left , right ) | 计算两点之间的距离。 | left：Cartesian3，计算距离的第一点。  right：Cartesian3，计算距离的第二点。 | Number |
| distanceSquared ( left , right ) | 计算两点之间的平方距离。比较平方距离使用此功能比使用 Cartesian3＃distance 比较距离更为有效。 | left：Cartesian3，计算距离的第一点。  right：Cartesian3，计算距离的第二点。 | Number |
| divideByScalar (cartesian , scalar , result ) | 将提供的笛卡尔分量除以提供的标量。 | cartesian：Cartesian3，要划分的笛卡尔坐标。  scalar：Number，要除以的标量。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| divideComponents ( left , right , result ) | 计算两个笛卡尔的分量商。 | left：Cartesian3，第一个笛卡尔式。  right：Cartesian3，第二个笛卡尔。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| dot ( left , right ) | 计算两个笛卡尔的点（标量）乘积。 | left：Cartesian3，第一个笛卡尔式。  right：Cartesian3，第二个笛卡尔。 | Number |
| equals( left , right ) | 比较提供的笛卡尔分量，然后返回如果相等，则为 true ，否则为 false 。 | left：Cartesian3，第一个笛卡尔式。  right：Cartesian3，第二个笛卡尔。 | Boolean |
| equalsEpsilon ( left , right , relativeEpsilon , absoluteEpsilon ) | 比较提供的笛卡尔分量，然后返回 true 如果它们通过绝对或相对公差测试，否则为 false 。 | left：Cartesian3，第一个笛卡尔式。  right：Cartesian3，第二个笛卡尔。  relativeEpsilon：Number，用于相等性测试的相对epsilon公差。  absoluteEpsilon：Number，用于相等性测试的绝对epsilon公差。 | Boolean |
| fromArray (array, startingIndex , result ) | 从数组中的两个连续元素创建Cartesian3。 | array：Array.<Number>，其三个连续元素分别对应于x，y和z分量的数组。  startingIndex：Number，第一个元素数组的偏移量，它对应于x分量。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| fromCartesian4 (cartesian, result ) | 从现有的Cartesian4创建Cartesian3实例。这只是需要笛卡尔4和水滴w的的x，y和z属性。 | cartesian：Cartesian4，Cartesian4实例，用于从中创建Cartesian3实例。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| fromDegrees (longitude, latitude, height , ellipsoid , result ) | 从以度为单位的经度和纬度值返回Cartesian3位置。 | longitude：Number，经度，以度为单位。  latitude：Number，纬度，以度为单位。  height：Number，椭球上方的高度，以米为单位。  ellipsoid：Ellipsoid，位置所在的椭圆体。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| fromDegreesArray (coordinates, ellipsoid , result ) | 返回给定的经度和纬度值数组（以度为单位），该数组由Cartesian3位置组成。 | coordinates：Array.<Number>，经度和纬度值列表，值交替显示[经度，纬度，经度，纬度，...]  ellipsoid：Ellipsoid，位置所在的椭圆体。  result：Array.<Cartesian3>，一组Cartesian3对象用于存储结果。 | Array.< Cartesian3 > |
| fromDegreesArrayHeights (coordinates, ellipsoid , result ) | 返回给定的经度，纬度和高度值数组，其中经度和纬度以度为单位的Cartesian3位置数组。 | coordinates：Array.<Number>，经度和纬度值列表，值交替显示[经度，纬度，经度，纬度，...]  ellipsoid：Ellipsoid，位置所在的椭圆体。  result：Array.<Cartesian3>，一组Cartesian3对象用于存储结果。 | Array.< Cartesian3 > |
| fromElements (x, y, z, result ) | 根据x，y和z坐标创建Cartesia32实例。 | x：Number，x坐标。  y：Number，y坐标。  z：Number，z做标。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| fromRadians (longitude, latitude, height , ellipsoid , result ) | 从以弧度给出的经度和纬度值返回Cartesian3位置。 | longitude：Number，经度，以度为单位。  latitude：Number，纬度，以度为单位。  height：Number，椭球上方的高度，以米为单位。  ellipsoid：Ellipsoid，位置所在的椭圆体。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| fromRadiansArray (coordinates, ellipsoid , result ) | 返回给定的笛卡尔3位置数组，该数组给出以弧度给出的经度和纬度值的数组。 | coordinates：Array.<Number>，经度和纬度值列表，值交替显示[经度，纬度，经度，纬度，...]  ellipsoid：Ellipsoid，位置所在的椭圆体。  result：Array.<Cartesian3>，一组Cartesian3对象用于存储结果。 | Array.< Cartesian3 > |
| fromRadiansArrayHeights (coordinates, ellipsoid , result ) | 返回给定的经度，纬度和高度值数组，其中经度和纬度以弧度表示的Cartesian3位置数组。 | coordinates：Array.<Number>，经度和纬度值列表，值交替显示[经度，纬度，经度，纬度，...]  ellipsoid：Ellipsoid，位置所在的椭圆体。  result：Array.<Cartesian3>，一组Cartesian3对象用于存储结果。 | Array.< Cartesian3 > |
| fromSpherical (spherical, result ) | 将提供的Spherical转换为Cartesian3坐标。 | spherical：Spherical，要转换为笛卡尔3的球形。  result：Array.<Cartesian3>，一组Cartesian3对象用于存储结果。 | Cartesian3 |
| lerp (start, end, t, result) | 使用提供的笛卡尔坐标计算t处的线性插值或外推。 | start：Cartesian3，对应于t的值为0.0。  end：Cartesian3，对应于t的值为1.0。  t：Number，沿t的插值点。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| magnitude (cartesian) | 计算笛卡尔的大小（长度）。 | cartesain：Cartesian3，要计算其大小的笛卡尔实例。 | Number |
| magnitude Squared(cartesian) | 计算提供的笛卡尔平方强度。 | cartesain：Cartesian3，要计算其大小的笛卡尔实例。 | Number |
| maximumByComponent (first, second, result) | 比较两个笛卡尔，并计算一个笛卡尔，其中包含所提供笛卡尔的最大分量。 | first：Cartesian3，笛卡尔进行比较。  second：Cartesian3，笛卡尔进行比较。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| maximumComponent (cartesian) | 计算所提供的直角坐标的最大分量的值。 | cartesain：Cartesian3，要使用的笛卡尔坐标。 | Number |
| midpoint (left, right, result) | 计算左右笛卡尔坐标之间的中点。 | left：Cartesian3，第一个笛卡尔式。  right：Cartesian3，第二个笛卡尔式。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| minimumByComponent (first, second, result) | 比较两个笛卡尔，并计算一个笛卡尔，其中包含所提供笛卡尔的最小分量。 | first：Cartesian3，笛卡尔进行比较。  second：Cartesian3，笛卡尔进行比较。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| minimumComponent (cartesian) | 计算所提供的直角坐标的最小分量的值。 | cartesain：Cartesian3，要使用的笛卡尔坐标。 | Number |
| mostOrthogonalAxis (cartesian, result) | 返回与提供的笛卡尔坐标最正交的轴。 | cartesain：Cartesian3，要在其上找到最正交轴的笛卡尔坐标。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| multiplyByScalar (cartesian, scalar, result) | 将提供的笛卡尔分量乘以提供的标量。 | cartesian：Cartesian3，第一个笛卡尔式。  scalar：Number，第二个笛卡尔式。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| multiplyComponents (left, right, result) | 计算两个笛卡尔的分量积。 | left：Cartesian3，第一个笛卡尔式。  right：Cartesian3，第二个笛卡尔式。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| negate (cartesian, result) | 取反提供的笛卡尔坐标。 | cartesian：Cartesian3，笛卡尔要否定。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| normalize (cartesian, result) | 计算提供的笛卡尔坐标系的标准化形式。 | cartesian：Cartesian3，要标准化的笛卡坐标。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| pack (value, array, startingIndex ) | 将提供的实例存储到提供的数组中。 | value：Cartesian3，要打包的值。  array：Array.<Number>，要打包的数组。  startingindex：Number，开始打包元素的数组索引。 | Array.<Number> |
| packArray (array, result ) | 将笛卡尔3数组展平为组件数组。 | array：Array.<Number>，要打包的数组。  result：Number，将结果存储到的数组。如果这是类型化的数组，则必须具有array.length \* 3个组件，否则将抛出 [DeveloperError](http://cesium.xin/cesium/cn/Documentation1.72/DeveloperError.html)。如果它是常规数组，则将其大小调整为具有（array.length \* 3）个元素。 | Array.<Number> |
| projectVector (a, b, result) | 将向量a投射到向量b上。 | a：Cartesian3，需要投影的向量。  b：Cartesian3，需要投影的b向量。  result：Cartesian3，结果笛卡尔。 | Cartesian3 |
| subtract (left, right, result) | 计算两个笛卡尔的分量差异。 | left：Cartesian3，第一个笛卡尔式。  right：Cartesian3，第二个笛卡尔式。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| unpack (array, startingIndex , result ) | 从压缩数组中检索实例。 | array：Array.<Number>，压缩数组。  startingIndex：Number，要解压缩的元素的起始索引。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| unpackArray (array, result ) | 将笛卡尔组件数组解压缩到笛卡尔数组中。 | array：Array.<Number>，要解包的组件数组。  result：Array.<Cartesian3>，将结果存储到的对象。 | Cartesian3 |

## Cartesian4类

4D笛卡尔点。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Cartesian4 ( x , y , z , w ) | 4D笛卡尔点 | X组件，Y分量，z分量，w分量 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| x | X组件 | Number |
| y | Y分量 | Number |
| z | Z分量 | Number |
| w | W分量 | Number |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| clone ( result ) | 复制此Cartesian4实例。 | Result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| equals ( right ) | 将此笛卡尔坐标与提供的笛卡尔分量进行比较，然后返回如果相等，则为 true ，否则为 false 。 | Right：Cartesian4，右侧笛卡尔坐标。 | Boolean |
| equalsEpsilon (left, right , relativeEpsilon , absoluteEpsilon ) | 将此笛卡尔坐标与提供的笛卡尔分量进行比较，然后返回 true 如果它们通过绝对或相对公差测试，否则为 false 。 | Right：Cartesian4，右侧笛卡尔坐标；relativeEpsilon：Number，用于相等性测试的相对epsilon公差；absoluteEpsilon：Number，用于相等性测试的绝对epsilon公差。 | Boolean |
| toString () | 创建一个以'（x，y）'格式表示此笛卡尔坐标的字符串。 |  | String |
| abs (cartesian, result) | 计算提供的笛卡尔坐标的绝对值。 | Cartesian：Cartesian4，要计算其绝对值的笛卡尔坐标。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| add ( left , right , result ) | 计算两个笛卡尔的按分量求和。 | left：Cartesian4，第一个笛卡尔式。  right：Cartesian4，第二个笛卡尔。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| clone  (cartesian, result ) | 复制一个Cartesian4实例。 | cartesian：Cartesian4，要复制的笛卡尔坐标。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| distance ( left , right ) | 计算两点之间的距离。 | left：Cartesian4，计算距离的第一点。  right：Cartesian4，计算距离的第二点。 | Number |
| distanceSquared ( left , right ) | 计算两点之间的平方距离。比较平方距离使用此功能比使用 Cartesian4＃distance 比较距离更为有效。 | left：Cartesian4，计算距离的第一点。  right：Cartesian4，计算距离的第二点。 | Number |
| divideByScalar (cartesian , scalar , result ) | 将提供的笛卡尔分量除以提供的标量。 | cartesian：Cartesian4，要划分的笛卡尔坐标。  scalar：Number，要除以的标量。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| divideComponents ( left , right , result ) | 计算两个笛卡尔的分量商。 | left：Cartesian4，第一个笛卡尔式。  right：Cartesian4，第二个笛卡尔。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| dot ( left , right ) | 计算两个笛卡尔的点（标量）乘积。 | left：Cartesian4，第一个笛卡尔式。  right：Cartesian4，第二个笛卡尔。 | Number |
| equals( left , right ) | 比较提供的笛卡尔分量，然后返回如果相等，则为 true ，否则为 false 。 | left：Cartesian4，第一个笛卡尔式。  right：Cartesian4，第二个笛卡尔。 | Boolean |
| equalsEpsilon ( left , right , relativeEpsilon , absoluteEpsilon ) | 比较提供的笛卡尔分量，然后返回 true 如果它们通过绝对或相对公差测试，否则为 false 。 | left：Cartesian4，第一个笛卡尔式。  right：Cartesian4，第二个笛卡尔。  relativeEpsilon：Number，用于相等性测试的相对epsilon公差。  absoluteEpsilon：Number，用于相等性测试的绝对epsilon公差。 | Boolean |
| fromArray (array, startingIndex , result ) | 从数组中的两个连续元素创建Cartesian3。 | array：Array.<Number>，其三个连续元素分别对应于x，y，z和w分量的数组。  startingIndex：Number，第一个元素数组的偏移量，它对应于x分量。  result：Cartesian3，将结果存储到的对象。 | Cartesian4 |
| fromColor (color, result ) | 从 [Color](http://cesium.xin/cesium/cn/Documentation1.72/Color.html)创建Cartesian4实例。 红色 ， 绿色 ， 蓝色 ，和 alpha 分别映射到 x ， y ， z 和 w 。 | color：Color，源颜色。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| fromElements (x, y, z, w , result ) | 根据x，y，z和w坐标创建Cartesian4实例。 | x：Number，x坐标。  y：Number，y坐标。  z：Number，z做标。  w：Number，w坐标。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| lerp (start, end, t, result) | 使用提供的笛卡尔坐标计算t处的线性插值或外推。 | start：Cartesian4，对应于t的值为0.0。  end：Cartesian4，对应于t的值为1.0。  t：Number，沿t的插值点。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| magnitude (cartesian) | 计算笛卡尔的大小（长度）。 | cartesain：Cartesian4，要计算其大小的笛卡尔实例。 | Number |
| magnitude Squared(cartesian) | 计算提供的笛卡尔平方强度。 | cartesain：Cartesian4，要计算其大小的笛卡尔实例。 | Number |
| maximumByComponent (first, second, result) | 比较两个笛卡尔，并计算一个笛卡尔，其中包含所提供笛卡尔的最大分量。 | first：Cartesian4，笛卡尔进行比较。  second：Cartesian4，笛卡尔进行比较。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| maximumComponent (cartesian) | 计算所提供的直角坐标的最大分量的值。 | cartesain：Cartesian4，要使用的笛卡尔坐标。 | Number |
| minimumByComponent (first, second, result) | 比较两个笛卡尔，并计算一个笛卡尔，其中包含所提供笛卡尔的最小分量。 | first：Cartesian4，笛卡尔进行比较。  second：Cartesian4，笛卡尔进行比较。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| minimumComponent (cartesian) | 计算所提供的直角坐标的最小分量的值。 | cartesain：Cartesian4，要使用的笛卡尔坐标。 | Number |
| mostOrthogonalAxis (cartesian, result) | 返回与提供的笛卡尔坐标最正交的轴。 | cartesain：Cartesian4，要在其上找到最正交轴的笛卡尔坐标。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| multiplyByScalar (cartesian, scalar, result) | 将提供的笛卡尔分量乘以提供的标量。 | cartesian：Cartesian4，第一个笛卡尔式。  scalar：Number，第二个笛卡尔式。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| multiplyComponents (left, right, result) | 计算两个笛卡尔的分量积。 | left：Cartesian4，第一个笛卡尔式。  right：Cartesian4，第二个笛卡尔式。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| negate (cartesian, result) | 取反提供的笛卡尔坐标。 | cartesian：Cartesian4，笛卡尔要否定。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| normalize (cartesian, result) | 计算提供的笛卡尔坐标系的标准化形式。 | cartesian：Cartesian4，要标准化的笛卡坐标。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| pack (value, array, startingIndex ) | 将提供的实例存储到提供的数组中。 | value：Cartesian4，要打包的值。  array：Array.<Number>，要打包的数组。  startingindex：Number，开始打包元素的数组索引。 | Array.<Number> |
| packArray (array, result ) | 将笛卡尔坐标数组展平为组件数组。 | array：Array.<Cartesian4>，要打包的数组。  result：Number，将结果存储到的数组。如果这是类型化的数组，则必须具有array.length \* 4个组件，否则将抛出 [DeveloperError](http://cesium.xin/cesium/cn/Documentation1.72/DeveloperError.html)。如果它是常规数组，则将其大小调整为具有（array.length \* 4）个元素。 | Array.<Number> |
| packFloat (value, result ) | 将任意浮点值打包为可使用uint8表示的4个值。 | value：Number，浮点数。  result：Cartesian4，包含压缩浮点的笛卡尔4。 | Cartesian4 |
| subtract (left, right, result) | 计算两个笛卡尔的分量差异。 | left：Cartesian4，第一个笛卡尔式。  right：Cartesian4，第二个笛卡尔式。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| unpack (array, startingIndex , result ) | 从压缩数组中检索实例。 | array：Array.<Number>，压缩数组。  startingIndex：Number，要解压缩的元素的起始索引。  result：Cartesian3，将结果存储到的对象。 | Cartesian4 |
| unpackArray (array, result ) | 将笛卡尔组件数组解压缩到笛卡尔数组中。 | array：Array.<Number>，要解包的组件数组。  result：Array.<Cartesian4>，将结果存储到的对象。 | Array.<Cartesian4> |

## Cartographic类

由经度，纬度和高度定义的位置。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Cartographic ( longitude , latitude , height ) | 由经度，纬度和高度定义的位置 | 经度，以弧度为单位  维度，以弧度为单位  椭球上方的高度，以米为单位 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| Longitude | 经度，以弧度为单位。 | Number |
| Latitude | 纬度，以弧度为单位。 | Number |
| Height | 椭圆上方的高度（以米为单位） | Number |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| clone (cartographic, result ) | 复制制图实例。 | Cartographic：Cartographic，要复制的地图。  Result：Cartographic，将结果存储到的对象。 | Cartographic |
| equals ( left , right ) | 比较提供的制图成分并返回如果相等，则为 true ，否则为 false 。 | left：Cartographic，右侧笛卡尔坐标。  right：Cartographic，右侧笛卡尔坐标。 | Boolean |
| equalsEpsilon ( left , right , epsilon ) | 比较提供的制图成分并返回 true （如果它们在提供的epsilon内），否则为 false 。 | left：Cartographic，右侧笛卡尔坐标。  right：Cartographic，右侧笛卡尔坐标。  epsilon：Number，用于相等性测试的 epsilon。 | Boolean |
| fromCartesian (cartesian, ellipsoid , result ) | 从笛卡尔位置创建一个新的制图实例。中的值生成的对象将以弧度表示。 | cartesian：Cartesian3，要转换为制图表达的笛卡尔位置。。  ellipsoid：Ellipsoid，位置所在的椭圆体。  result：Cartographic，将结果存储到的对象。 | Cartographic |
| fromDegrees (longitude, latitude, height , result ) | 根据经度和纬度创建一个新的制图实例以度为单位指定。结果对象中的值将弧度。 | longitude：Number，经度，以度为单位。  latitude：Number，纬度，以度为单位。  height：Number，椭球上方的高度，以米为单位。  result：Cartographic，将结果存储到的对象。 | Cartographic |
| fromRadians (longitude, latitude, height , result ) | 根据经度和纬度创建一个新的制图实例以弧度指定。 | longitude：Number，经度，以度为单位。  latitude：Number，纬度，以度为单位。  height：Number，椭球上方的高度，以米为单位。  result：Cartographic，将结果存储到的对象。 | Cartographic |
| toCartesian (cartographic, ellipsoid , result ) | 根据制图输入创建一个新的Cartesian3实例。输入的值对象应以弧度为单位。 | cartographic：Cartographic，输入将转换为Cartesian3输出。  ellipsoid：Ellipsoid，位置所在的椭圆体。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| clone (result ) | 复制此实例。 | Result：Cartographic，将结果存储到的对象。 | Cartographic |
| equals ( right ) | 将提供的内容与此制图成分进行比较，然后返回如果相等，则为 true ，否则为 false 。 | right：Cartographic，第二个制图。 | Boolean |
| equalsEpsilon ( left , epsilon ) | 将提供的内容与此制图成分进行比较，然后返回 true （如果它们在提供的epsilon内），否则为 false 。 | right：Cartographic，第二个制图。  epsilon：Number，用于相等性测试的 epsilon。 | Boolean |

## Color类

使用红色，绿色，蓝色和Alpha值指定的颜色。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Color ( red , green , blue , alpha ) | 使用红色，绿色，蓝色和Alpha值指定的颜色，范围从 0 （无强度）到 1.0 （全强度）。 | 红色部分，绿色部分，蓝色部分，alpha组件 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| red | 红色部分 | Number |
| green | 绿色部分 | Number |
| blue | 蓝色部分 | Number |
| alpha | alpha组件 | Number |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| add (left, right, result) | 计算两种颜色的按分量求和。 | left：Color，第一种颜色。  right：Color，第二种颜色。  result：Color，将结果存储到的对象。 | Color |
| byteToFloat (number) | 将0到255范围内的'字节'颜色分量转换为'浮动'颜色分量，范围为0到1.0。 | number：Number，要转换的数字。 | Number |
| clone (color, result ) | 复制颜色。 | Color：Color，要复制的颜色。  result：Color，可选存储结果的对象，如果未定义，将创建一个新实例。 | Color |
| divide (left, right, result) | 计算两种颜色的分量商。 | left：Color，第一种颜色。  right：Color，第二种颜色。  result：Color，将结果存储到的对象。 | Color |
| divideByScalar (color, scalar, result) | 将提供的Color按分量除以提供的标量。 | color：Color，要划分颜色。  scalar：Number，要除以的标量。  result：Color，将结果存储到的对象。 | Color |
| equals (left, right) | 如果第一种颜色等于第二种颜色，则返回true。 | left：Color，第一种颜色。  right：Color，第二种颜色。 | Boolean |
| floatToByte (number) | 将0到1.0范围内的'浮动'颜色分量转换为'字节'颜色分量，范围为0到255。 | number：Number，要转换的数字。 | Number |
| fromAlpha (color, alpha, result ) | 创建具有相同红色，绿色和蓝色成分的新颜色指定颜色的颜色，但具有指定的alpha值。 | color：Color，底色。  alpha：Number，新的alpha组件。  result：Color，将结果存储到的对象。 | Color |
| fromBytes ( red , green , blue , alpha , result ) | 创建使用红色，绿色，蓝色和Alpha值指定的新颜色范围是0到255，内部将它们转换为0.0到1.0。 | red：Number，红色部分。  green：Number，绿色组件。  blue：Number，蓝色组件。  alpha：Number， Alpha组件。  result：Color，将结果存储到的对象。 | Color |
| fromCartesian4 (cartesian, result ) | 从 [Cartesian4](http://cesium.xin/cesium/cn/Documentation1.72/Cartesian4.html)创建Color实例。 x ， y ， z ，和 w 分别映射到 red ， green ， blue 和 alpha 。 | cartesian：Cartesian4，源笛卡尔。  result：Color，将结果存储到的对象。 | Color |
| fromCssColorString (color, result ) | 根据CSS颜色值创建一个Color实例。 | color：String，＃rgb，＃rgba，＃rrggbb，＃rrggbbaa，rgb（），rgba（），hsl（）或hsla（）格式的CSS颜色值。。  result：Color，存储结果的对象，如果未定义，将创建一个新实例。 | Color |
| fromHsl ( hue , saturation , lightness , alpha , result ) | 根据色相，饱和度和亮度创建一个Color实例。 | hue：Number，色相角0...1。  saturation：Number，可选，饱和度0...1。  lightness：Number，亮度值0...1。  alpha：Number， Alpha分量0...1。  result：Color，存储结果的对象，如果未定义，将创建一个新实例。 | Color |
| fromRgba (rgba, result ) | 使用字节序从单个数字无符号32位RGBA值创建新的颜色系统的。 | raba：Number，单个数字无符号32位RGBA值。  result：Color，存储结果的对象，如果未定义，将创建一个新实例。 | Color |
| mod (left, right, result) | 计算两种颜色的分量模数。 | left：Color，第一种颜色。  right：Color，第二种颜色。  result：Color，将结果存储到的对象。 | Color |
| multiply (left, right, result) | 计算两种颜色的分量乘积。 | left：Color，第一种颜色。  right：Color，第二种颜色。  result：Color，将结果存储到的对象。 | Color |
| multiplyByScalar (color, scalar, result) | 将提供的Color逐个乘以提供的标量。 | color：Color，要缩放的颜色。  scalar：Number，要与之相乘的标量。  result：Color，将结果存储到的对象。 | Color |
| pack (value, array, startingIndex ) | 将提供的实例存储到提供的数组中。 | value：Color，要打包的值。  array：Array.<Number>，要打包的数组。  startingIndex：Number，开始打包元素的数组索引。 | Array.<Number> |
| subtract (left, right, result) | 计算两种颜色的分量差异。 | left：Color，第一种颜色。  right：Color，第二种颜色。  result：Color，将结果存储到的对象。 | Color |
| unpack (array, startingIndex , result ) | 从压缩数组中检索实例。 | array：Array.<Number>，压缩数组。  startingIndex：Number，要解压缩的元素的起始索引。  result：Color，将结果存储到的对象。 | Color |
| brighten (magnitude, result) | 通过提供的幅度使该颜色变亮。 | magnitude：Number，一个正数，表示要增亮的数量。  result：Color，将结果存储到的对象。 | Color |
| clone ( result ) | 返回Color实例的副本。 | result：Color，存储结果的对象，如果未定义，将创建一个新实例。 | Color |
| darken (magnitude, result) | 使颜色变暗所提供的幅度 | Magnitude：Number，一个正数，表示要变暗的量。  result:Color将结果存储到的对象。 | Color |
| equals (other) | 如果此Color等于other，则返回true。 | other：Color，要比较是否相等的颜色。 | Boolean |
| equalsEpsilon (other, epsilon ) | 如果此Color在指定的epsilon中等于其他分量，则返回 true 。 | other：Color，要比较是否相等的颜色。  epsilon：Number，可选用于相等性测试的epsilon。 | Boolean |
| toBytes ( result ) | 将此颜色转换为红色，绿色，蓝色和Alpha值的数组范围是0到255。 | result：Array.<Number>，选存储结果的数组，如果未定义，将创建一个新实例。 | Array.<Number> |
| toCssColorString () | 创建一个包含该颜色的CSS颜色值的字符串。 |  | String |
| toCssHexString () | 创建一个包含此颜色的CSS十六进制字符串颜色值的字符串。 |  | String |
| toRgba () | 使用字节序将此颜色转换为单个数字无符号32位RGBA值系统的。 |  | Number |
| toString () | 创建一个以'（red，green，blue，alpha）'格式表示此Color的字符串 |  | String |
| withAlpha (alpha, result ) | 创建具有相同红色，绿色和蓝色成分的新颜色作为此颜色，但具有指定的alpha值。 | alpha：Number，新的alpha组件。  result：Color，可选将结果存储到的对象 | Color |

## EllipsoidTerrainProvider类

一个非常简单的TerrainProvider，通过细分椭圆形来生成几何表面。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| EllipsoidTerrainProvider ( options ) | 一个非常简单的 TerrainProvider ，通过细分椭圆形来生成几何表面。 | 具有tilingScheme、椭球属性的对象。  tilingScheme：tilingScheme，平铺方案指定椭圆形表面碎成瓷砖。如果未提供此参数，则 GeographicTilingScheme 使用。  椭球 椭球 椭球。如果指定了tilingScheme，忽略此参数，而是使用切片方案的椭球。如果两者都不参数，则使用WGS84椭球。 |

### 成员变量

无

### 方法

无

## Math类

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Math() | 数学函数 |  |

### 成员变量

无

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| acosClamped(value) | 计算Math.acos(value)，但首先将value限定在[-1.0,1.0]范围内，以便函数永远不会返回NaN | value：计算acos的值 | 如果值在[-1.0,1.0]范围内，则为该值的acos;如果值不在该范围内，则为-1.0或1.0，以较近者为准 |
| asinClamped(value) | 计算Math.asin(value)，但首先限制value在 [-1.0, 1.0] 范围内，以便函数永远不会返回 NaN。 | value：Number，计算asin的值。 | Number |
| chordLength(angle, radius) | 给定圆的半径和两点之间的角度，计算两点之间的弦长。 | angle：Number，两点之间的角度。  radius：Number，圆的半径。 | Number |
| clamp(value, min, max) | 将一个值限制在两个值之间。 | value：Number，要约束的值。  min：Number，最小值。  min：Number，最大值。 | Number |
| clampToLatitudeRange(angle) | 将纬度值（以弧度为单位）限制在 [ -Math.PI/2, Math.PI/2) 范围内的便利函数。用于在需要正确范围的对象中使用之前清理数据。 | angle：Number，以弧度为单位的经度值，以限制在[-Math.PI/2，Math.PI/2]范围内。 | Number |
| convertLongitudeRange(angle) | 将以弧度为单位的经度值转换为范围 [ -Math.PI, Math.PI)。 | angle：Number，将以弧度为单位的经度值准换范围[-Math.PI，Math.PI]。 | Number |
| cosh(value) | 返回数字的双曲余弦值。值的双曲余弦定义为 ( e x  + e -x )/2.0，其中e是欧拉数，大约为 2.71828183。  如果参数为 NaN，则结果为 NaN。  如果参数是无限的，那么结果是正无穷大。  如果参数为零，则结果为 1.0。 | value：Number，要返回其双曲线余弦值的数字。 | Number |
| equalsEpsilon(left, right, relativeEpsilon, absoluteEpsilon) | 使用绝对或相对公差测试确定两个值是否相等。这有助于避免在直接比较浮点值时由于舍入误差引起的问题。首先使用绝对公差测试比较这些值。如果失败，则执行相对耐受性测试。如果您不确定左右的大小，请使用此测试。 | left：Number，要比较的第一个值。  right：Number，要比较的第二个值。  relativeEpsilon：Number，相对公差测试left之间的最大包含增量。  absoluteEpsilon：Number，相对公差测试left之间的最大包含增量。 | Boolean |
| factorial(n) | 计算所提供数字的阶乘。 | n：Number，要计算其阶乘的数字。 | Number |
| fastApproximateAtan(x) | 计算范围[-1，1]内输入的Atan的快速近似值。 | x：Number，[-1,1]范围内的输入数字。 | Number |
| fastApproximateAtan2(x, y) | 计算任意输入标量的 Atan2(x, y) 的快速近似值。基于 nvidia 的 cg 参考实现的范围缩小数学 | x：Number，如果y为零，则输入数字不为零。  y：Number，如果x为零，则输入数字不为零。 | Number |
| fromSNorm(value, rangeMaximum) | 将 [0, rangeMaximum] 范围内的 SNORM 值转换为 [-1.0, 1.0] 范围内的标量。 | value：Number，[0, rangeMaximum] 范围内的 SNORM 值。  rangeMaximum：Number，可选，SNORM 范围内的最大值，默认为 255。 | Number |
| greaterThan(left, right, absoluteEpsilon) | 确定左值是否大于右值。如果两个值在 absoluteEpsilon彼此之内，则认为它们相等并且此函数返回 false。 | left：Number，要比较的第一个数字。  right：Number，要比较的第二个数字。  absoluteEpsilon：Number，用于比较的绝对epsilon。 | Boolean |
| greaterThanOrEquals(left, right, absoluteEpsilon) | 确定左侧值是否大于或等于右侧值。如果两个值在彼此的绝对Epsilon内，则认为它们相等，此函数返回true。 | left：Number，要比较的第一个数字。  right：Number，要比较的第二个数字。  absoluteEpsilon：Number，用于比较的绝对epsilon。 | Boolean |
| incrementWrap(n, maximumValue, minimumValue) | 增加数量和包装数量最小值,如果超过最大值。 | n：Number，要比较的第一个数字。  maximumValue：Number，要比较的第二个数字。  minimumValue：Number，用于比较的绝对epsilon。 | Number |
| isPowerOfTwo(n) | 确定正整数是否为2的幂。 | n：Number，要测试的正整数。 | Boolean |
| lerp(p, q, time) | 计算两个值的线性插值。 | p：Number，要插值的起始值。  q：Number，要插值的结束值。  time：Number，插值范围通常在[0.0，1.0]范围内。 | Number |
| lessThan(left, right, absoluteEpsilon) | 确定左值是否小于右值。如果两个值在彼此的绝对Epsilon内，则认为它们相等，此函数返回false。 | left：Number，要比较的第一个数字。  right：Number，要比较的第二个数字。  absoluteEpsilon：Number，用于相等性测试的绝对epsilon公差。 | Boolean |
| lessThanOrEquals(left, right, absoluteEpsilon) | 确定左侧值是否小于或等于右侧值。如果两个值在彼此的绝对Epsilon内，则认为它们相等，此函数返回true。 | left：Number，要比较的第一个数字。  right：Number，要比较的第二个数字。  absoluteEpsilon：Number，用于相等性测试的绝对epsilon公差。 | Boolean |
| logBase(number, base) | 查找数字的底对数。 | number：Number，数字。  base：Number，基数。 | Number |
| mod(m, n) | 股息的模运算。 | m：Number，股息。  n：Number，除数。 | Number |
| negativePiToPi(angle) | 生成范围为-Pi<=角度<=Pi的角度，该角度与提供的角度相等。 | angle：Number，以弧度为单位。 | Number |
| nextPowerOfTwo(n) | 计算大于或等于提供的正整数的两个整数的下一次幂。 | n：Number，要测试的正整数。 | Number |
| nextRandomNumber() | 使用Mersenne扭曲器生成范围为[0.0，1.0）的随机浮点数。 |  | Number |
| normalize(value, rangeMinimum, rangeMaximum) | 将范围为[rangeMinimum，rangeMaximum]的标量值转换为范围为[0.0，1.0]的标量。 | value：Number，范围[rangeMinimum，rangeMaximum]中的标量值。  rangeMinimum：Number，映射范围中的最小值。  rangeMaximum：Number，映射范围中的最大值。 | Number |
| randomBetween(min, max) | 生成两个数字之间的随机数。 | min：Number，最小值。  max：Number，最大值。 | Number |
| setRandomNumberSeed(seed) | 在cesiummath# nextRandomNumber中设置随机数生成器使用的种子。 | seed：Number，用作种子的整数。 |  |
| signNotZero(value) | 如果给定值为正或零，则返回1.0；如果为负，则返回-1.0。这类似于CeiumMath#符号，只是当输入值为0.0时返回1.0而不是0.0。 | value：Number，要返回符号的值。 | Number |
| toDegrees(radians) | 将弧度转换为度。 | radians：Number，要转换为弧度的角度。 | Number |
| toRadians(degrees) | 将角度转换为弧度 | degrees：Number，要转换的角度。 | Number |
| toSNorm(value, rangeMaximum) | 将范围[-1.0,1.0]中的标量值转换为范围[0,rangemmaximum]中的SNORM。 | value：Number，范围[-1.0，1.0]内的标量值。  rangeMaximum：Number，映射范围中的最大值，默认为255。 | Number |
| zeroToTwoPi(angle) | 生成一个范围为0 <= angle <= 2Pi的角，与所提供的角相等。 | angle：Number，以弧度为单位。 | Number |

## Matrix3类

一个3x3矩阵，可按列优先顺序数组进行索引。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Matrix3 ( column0Row0 , column1Row0 , column2Row0 , column0Row1 , column1Row1 , column2Row1 , column0Row2 , column1Row2 , column2Row2 ) | 一个3x3矩阵，可按列优先顺序数组进行索引。构造函数参数按行顺序排列，以提高代码的可读性。 | 行列值 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| length | 获取集合中的项目数。 | Number |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| clone ( result ) | 复制提供的Matrix3实例。 | Result：将结果存储到的对象。 | Matrix3 |
| equals ( right ) | 将此矩阵与提供的矩阵进行逐项比较，然后返回如果相等，则为 true ，否则为 false 。 | Right：Matrix3，右侧矩阵。 | Boolean |
| equalsEpsilon ( right , epsilon ) | 将此矩阵与提供的矩阵进行逐项比较，然后返回 true （如果它们在提供的epsilon内），否则为 false 。 | Right：Matrix3，右侧矩阵。  Epsilon：Number，可选用于相等性测试的epsilon。 | Boolean |
| toString () | 创建一个表示此Matrix的字符串，每一行为在单独的行上，格式为（（column0，column1）） |  | String |
| abs (matrix, result) | 计算一个矩阵，其中包含提供的矩阵元素的绝对（无符号）值。 | matrix：Matrix3，具有符号元素的矩阵。  result：Matrix3，将结果存储到的对象。 | Matrix3 |
| add (left, right, result) | 计算两个矩阵的总和。 | left：Matrix3，第一个矩阵。  right：Matrix3，第二个矩阵。  result：Matrix3，将结果存储到的对象。 | Matrix3 |
| clone (matrix, result ) | 复制一个Matrix3实例。 | matrix：Matrix3，要复制的矩阵。  result：Matrix3，将结果存储到的对象。 | Matrix3 |
| computeEigenDecomposition (matrix, result ) | 计算对称矩阵的特征向量和特征值。  返回对角矩阵和unit矩阵，使得: 矩阵= ary矩阵\*对角矩阵\*转置（unit矩阵）沿对角矩阵对角线的值是特征值。列unit矩阵的是对应的特征向量。 | matrix：Matrix3，分解为对角矩阵和unit矩阵的矩阵。应该是对称的。  result：Object，具有单一和对角线属性的对象，这些属性是将结果存储到其上的矩阵。 | Object |
| determinant (matrix) | 计算所提供矩阵的行列式。 | matrix：Matrix3，要使用的矩阵。 | Number |
| equals ( left , right ) | 按组件比较提供的矩阵并返回如果相等，则为 true ，否则为 false 。 | left：Matrix3，第一个矩阵。  right：Matrix3，第二个矩阵。 | Boolean |
| equalsEpsilon ( left , right , epsilon ) | 按组件比较提供的矩阵并返回 true （如果它们在提供的epsilon内），否则为 false 。 | left：Matrix3，第一个矩阵。  right：Matrix3，第二个矩阵。  right：Number，用于相等性测试的epsilon。 | Boolean |
| fromArray (array, startingIndex , result ) | 从数组中的9个连续元素创建Matrix3。 | array：Array.<Number>，其9个连续元素对应于矩阵位置的数组。假定列为主要顺序。。  startingIndex：Number，第一个元素的数组中的偏移量，它对应于矩阵中第一列第一行的位置。  result：Matrix3，将结果存储到的对象。 | Matrix3 |
| fromColumnMajorArray (values, result ) | 从列优先顺序数组创建Matrix2实例。 | values：Array.<Number>，列为主要顺序的数组。  result：Matrix3，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix3 |
| fromCrossProduct (vector, result ) | 计算表示Cartesian3向量的叉积等效矩阵的Matrix3实例。 | values：Cartesian3，叉积运算左侧的向量。  result：Matrix3，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix3 |
| fromHeadingPitchRoll (headingPitchRoll, result ) | 根据提供的headingPitchRoll计算3x3旋转矩阵。 | headingPitchRoll：HeadingPitchRoll，要使用的headingPitchRoll。  result：Matrix3，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix3 |
| fromQuaternion (quaternion, result ) | 根据提供的四元数计算3x3旋转矩阵。 | quaternion：Quaternion，要使用的四元数。  result：Matrix3，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix3 |
| fromRotationX (angle, result ) | 创建围绕X轴的旋转矩阵。 | angle：Number，旋转角度，以弧度为单位。正角是逆时针方向。  result：Matrix3，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix3 |
| fromRotationY (angle, result ) | 创建围绕Y轴的旋转矩阵。 | angle：Number，旋转角度，以弧度为单位。正角是逆时针方向。  result：Matrix3，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix3 |
| fromRotationZ (angle, result ) | 创建围绕Z轴的旋转矩阵。 | angle：Number，旋转角度，以弧度为单位。正角是逆时针方向。  result：Matrix3，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix3 |
| fromRowMajorArray (values, result ) | 从行优先顺序数组创建Matrix3实例。所得矩阵将按列优先顺序排列。 | values：Array.<Number>，行优先顺序的数组。  result：Matrix3，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix3 |
| fromScale (scale, result ) | 计算表示不均匀比例的Matrix3实例。 | scale：Cartesian3，x，y和z比例因子。  result：Matrix2，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix3 |
| fromUniformScale (scale, result ) | 计算表示统一比例尺的Matri3实例。 | scale：Number，统一比例因子。  result：Matrix3，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix3 |
| getColumn (matrix, index, result) | 在提供的索引处作为Cartesian3实例检索矩阵列的副本。 | matrix：matrix3，使用的矩阵。  index：Number，要检索的列的从零开始的索引。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| getElementIndex (row, column) | 计算提供的行和列处元素的数组索引。 | row：Number，该行从零开始的索引。  column：Number，该行从零开始的索引。 | Number |
| getMaximumScale (matrix) | 假设矩阵是仿射变换，则计算最大比例。最大比例是列向量的最大长度。 | matrix：Matrix3，矩阵。  result：Matrix3，将结果存储到的对象。 | Number |
| getRotation (matrix, result) | 假设矩阵是仿射变换，则提取旋转。 | matrix：Matrix3，矩阵。 | Matrix3 |
| getRow (matrix, index, result) | 以Cartesian3实例的形式在提供的索引处检索矩阵行的副本。 | matrix：Matrix3，使用的矩阵。  index：Number，要检索的行的从零开始的索引。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| getScale (matrix, result) | 假设矩阵是仿射变换，则提取非均匀比例。 | matrix：Matrix3，矩阵。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| multiply (left, right, result) | 计算两个矩阵的乘积。 | left：Matrix3，第一个矩阵。  right：Matrix3，第二个矩阵。  result：Matrix3，将结果存储到的对象。 | Matrix3 |
| multiplyByScalar (matrix, scalar, result) | 计算矩阵与标量的乘积。 | matrix：Matrix3，矩阵。  scalar：Number，要乘以的数字。  result：Matrix3，将结果存储到的对象。 | Matrix3 |
| multiplyByScale (matrix, scale, result) | 计算矩阵乘以（非均匀）标度的乘积，就好像该标度是一个标度矩阵一样。 | matrix：Matrix3，左侧的矩阵。  scalar：Cartesian3，比例尺在右侧。  result：Matrix3，将结果存储到的对象。 | Matrix3 |
| multiplyByVector (matrix, cartesian, result) | 计算矩阵与列向量的乘积。 | matrix：Matrix3，矩阵。  cartesian：Cartesian3，列。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| negate (matrix, result) | 创建所提供矩阵的求反副本。 | matrix：Matrix3，求反的矩阵。  result：Matrix3，将结果存储到的对象。 | Matrix3 |
| pack (value, array, startingIndex ) | 将提供的实例存储到提供的数组中。 | value：Matrix3，要打包的值。  array：Array.<Number>，要打包的数组。  startingIndex：Number，开始打包元素的数组索引。 | Array.<Number> |
| setColumn (matrix, index, cartesian, result) | 计算一个新矩阵，用提供的Cartesian3实例替换提供的矩阵中的指定列。 | matrix：Matrix3，使用的矩阵。  index：Number，要设置的列的从零开始的索引。  cartesian：Cartesian2，直角坐标系，其值将分配给指定的列。  result：Matrix3，将结果存储到的对象。 | Matrix3 |
| setRow (matrix, index, cartesian, result) | 计算一个新矩阵，用提供的Cartesian2实例替换提供的矩阵中的指定行。 | matrix：Matrix3，使用的矩阵。  index：Number，要设置的行的从零开始的索引。  cartesian：Cartesian3，直角坐标系，其值将分配给指定的行。  result：Matrix3，将结果存储到的对象。 | Matrix3 |
| subtract (left, right, result) | 计算两个矩阵的差。 | left：Matrix3，第一个矩阵。  right：Matrix3，第二个矩阵。  result：Matrix3，将结果存储到的对象。 | Matrix3 |
| toArray (matrix, result ) | 从提供的Matrix3实例创建一个数组。该数组将按列优先顺序排列。 | matrix,：Matrix3，要使用的矩阵。  result：Array.<Number>，将结果存储到的对象。 | Array.<Number> |
| transpose (matrix, result) | 计算提供的矩阵的转置。 | matrix,：Matrix3，要转置的矩阵。  result：Matrix3，将结果存储到的对象。 | Matrix3 |
| unpack (array, startingIndex , result ) | 从压缩数组中检索实例。 | array：Array.<Number>，压缩数组。  startingIndex：Number，要解压缩的元素的起始索引。  result：Matrix3，将结果存储到的对象。 | Matrix3 |

## Matrix4类

一个4x4矩阵，可索引为列主序数组。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Matrix4 ( column0Row0 , column1Row0 , column2Row0 , column3Row0 , column0Row1 , column1Row1 , column2Row1 , column3Row1 , column0Row2 , column1Row2 , column2Row2 , column3Row2 , column0Row3 , column1Row3 , column2Row3 , column3Row3 ) | 一个4x4矩阵，可索引为列主序数组。构造函数参数按行顺序排列，以提高代码的可读性。 | 行列值 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| length | 获取集合中的项目数。 | Number |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| clone ( result ) | 复制提供的Matrix4实例。 | Result：将结果存储到的对象。 | Matrix4 |
| equals ( right ) | 将此矩阵与提供的矩阵进行逐项比较，然后返回如果相等，则为 true ，否则为 false 。 | Right：Matrix4，右侧矩阵。 | Boolean |
| equalsEpsilon ( right , epsilon ) | 将此矩阵与提供的矩阵进行逐项比较，然后返回 true （如果它们在提供的epsilon内），否则为 false 。 | Right：Matrix4，右侧矩阵。  Epsilon：Number，可选用于相等性测试的epsilon。 | Boolean |
| toString () | 创建一个表示此Matrix的字符串，每一行为在单独的行上，格式为（（column0，column1）） |  | String |
| abs (matrix, result) | 计算一个矩阵，其中包含提供的矩阵元素的绝对（无符号）值。 | matrix：Matrix4，具有符号元素的矩阵。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| add (left, right, result) | 计算两个矩阵的总和。 | left：Matrix4，第一个矩阵。  right：Matrix4，第二个矩阵。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| clone (matrix, result ) | 复制一个Matrix4实例。 | matrix：Matrix4，要复制的矩阵。  result：Matrix4，`将结果存储到的对象。 | Matrix4 |
| computeInfinitePerspectiveOffCenter (left, right, bottom, top, near, result) | 计算表示无限偏心透视转换的Matrix4实例。 | left：Number，可以看到的相机左侧的米数。  right：Number，可以看到的相机右侧的米数。  bottom：Number，可以看到的相机下方的米数。  top：Number，可以看到的相机上方的米数。  near：Number，到附近飞机的距离，以米为单位。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| computeOrthographicOffCenter (left, right, bottom, top, near, far, result) | 计算表示正交变换矩阵的Matrix4实例。 | left：Number，可以看到的相机左侧的米数。  right：Number，可以看到的相机右侧的米数。  bottom：Number，可以看到的相机下方的米数。  top：Number，可以看到的相机上方的米数。  near：Number，到附近飞机的距离，以米为单位。  far：Number，到远平面的距离，以米为单位。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| computePerspectiveFieldOfView (fovY, aspectRatio, near, far, result) | 计算表示透视变换矩阵的Matrix4实例。 | fovY：Number，沿Y轴的弧度视场。  aspectRatio：Number，长宽比。  near：Number，到附近飞机的距离，以米为单位。  far：Number，到远平面的距离，以米为单位。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| computePerspectiveOffCenter (left, right, bottom, top, near, far, result) | 计算表示偏心透视转换的Matrix4实例。 | left：Number，可以看到的相机左侧的米数。  right：Number，可以看到的相机右侧的米数。  bottom：Number，可以看到的相机下方的米数。  top：Number，可以看到的相机上方的米数。  near：Number，到附近飞机的距离，以米为单位。  far：Number，到远平面的距离，以米为单位。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| computeView (position, direction, up, right, result) | 计算从世界空间转换为视图空间的Matrix4实例。 | position：Cartesian3，相机的位置。  direction：Cartesian3，前进的方向。  up：Cartesian3，向上的方向。  right：Cartesian3，正确的方向。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| computeViewportTransformation ( viewport , nearDepthRange , farDepthRange , result ) | 计算从规范化的设备坐标转换为窗口坐标的Matrix4实例。 | viewport：Object，视口的角。  nearDepthRange：Nnmber，窗口坐标中的近平面距离。  farDepthRange：Nnmber，窗口坐标中的远平面距离。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| equals ( left , right ) | 按组件比较提供的矩阵并返回如果相等，则为 true ，否则为 false 。 | left：Matrix4，第一个矩阵。  right：Matrix4，第二个矩阵。 | Boolean |
| equalsEpsilon ( left , right , epsilon ) | 按组件比较提供的矩阵并返回 true （如果它们在提供的epsilon内），否则为 false 。 | left：Matrix4，第一个矩阵。  right：Matrix4，第二个矩阵。  right：Number，用于相等性测试的epsilon。 | Boolean |
| fromArray (array, startingIndex , result ) | 从数组中的16个连续元素创建Matrix4。 | array：Array.<Number>，其16个连续元素对应于矩阵位置的数组。假定列为主要顺序。  startingIndex：Number，第一个元素的数组中的偏移量，它对应于矩阵中第一列第一行的位置。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| fromCamera (camera, result ) | 从Camera计算Matrix4实例。 | camera：Camera，要使用的相机。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| fromColumnMajorArray (values, result ) | 从列优先顺序数组创建Matrix4实例。 | values：Array.<Number>，列为主要顺序的数组。  result：Matrix4，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix4 |
| fromRotationTranslation (rotation, translation , result ) | 从Matrix3计算代表旋转的Matrix4实例和代表翻译的Cartesian3。 | rotation：Matrix3，矩阵的左上角代表旋转。  translation：Cartesian3，矩阵的右上角代表翻译。  result：Matrix4，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix4 |
| fromRowMajorArray (values, result ) | 从行优先顺序数组创建Matrix3实例。所得矩阵将按列优先顺序排列。 | values：Array.<Number>，行优先顺序的数组。  result：Matrix4，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix4 |
| fromScale (scale, result ) | 计算表示不均匀比例的Matrix4实例。 | scale：Cartesian3，x，y和z比例因子。  result：Matrix4，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix4 |
| fromTranslation (translation, result ) | 从笛卡尔3创建一个表示转换的Matrix4实例。 | translation：Cartesian3，矩阵的右上角代表翻译。  result：Matrix4，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix4 |
| fromTranslationQuaternionRotationScale (translation, rotation, scale, result ) | 根据平移，旋转和缩放（TRS）计算Matrix4实例用四元数表示旋转。 | translation：Cartesian3，翻译转换。  rotation：Quaternion，旋转变换。  scale：Cartesian3，非均匀尺度变换。  result：Matrix4，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix4 |
| fromTranslationRotationScale (translationRotationScale, result ) | 从 [TranslationRotationScale](http://cesium.xin/cesium/cn/Documentation1.72/TranslationRotationScale.html)实例创建Matrix4实例。 | translationRotationScale：TranslationRotationScale，实例。  result：Matrix4，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix4 |
| fromUniformScale (scale, result ) | 计算表示统一比例尺的Matrix4实例。 | scale：Number，统一比例因子。  result：Matrix4，存储结果的对象，如果未定义，将创建一个新实例。 | Matrix4 |
| getColumn (matrix, index, result) | 在提供的索引处作为Cartesian4实例检索矩阵列的副本。 | matrix：matrix4，使用的矩阵。  index：Number，要检索的列的从零开始的索引。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| getElementIndex (row, column) | 计算提供的行和列处元素的数组索引。 | row：Number，该行从零开始的索引。  column：Number，该行从零开始的索引。 | Number |
| getMatrix3 (matrix, result) | 假设矩阵是仿射变换矩阵，则获取所提供矩阵的左上3x3旋转矩阵。 | matrix：Matrix4，使用的矩阵。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| getMaximumScale (matrix) | 假设矩阵是仿射变换，则计算最大比例。最大比例是左上角列向量的最大长度3x3矩阵。 | matrix：Matrix3，矩阵。 | Number |
| getRow (matrix, index, result) | 以Cartesian4实例的形式在提供的索引处检索矩阵行的副本。 | matrix：Matrix4，使用的矩阵。  index：Number，要检索的行的从零开始的索引。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| getScale (matrix, result) | 假设矩阵是仿射变换，则提取非均匀比例。 | matrix：Matrix4，矩阵。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| getTranslation (matrix, result) | 假设矩阵是仿射变换矩阵，则获取提供的矩阵的平移部分。 | matrix：Matrix4，矩阵。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| inverse (matrix, result) | 使用Cramers规则计算所提供矩阵的逆。如果行列式为零，则矩阵无法求逆，并引发异常。如果矩阵是仿射变换矩阵，则效率更高使用 [Matrix4.inverseTransformation](http://cesium.xin/cesium/cn/Documentation1.72/Matrix4.html" \l ".inverseTransformation)将其反转。 | matrix：Matrix4，要求逆的矩阵。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| inverseTransformation (matrix, result) | 假设所提供的矩阵是仿射变换矩阵，其中左上3x3元素是旋转矩阵，第四个元素的前三个元素专栏是翻译。底行假定为[0，0，0，1]。矩阵未验证为正确形式。此方法比计算一般4x4的逆速度更快使用 [Matrix4.inverse](http://cesium.xin/cesium/cn/Documentation1.72/Matrix4.html" \l ".inverse)的矩阵。 | matrix：Matrix4，要求逆的矩阵。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| multiply (left, right, result) | 计算两个矩阵的乘积。 | left：Matrix4，第一个矩阵。  right：Matrix4，第二个矩阵。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| multiplyByMatrix3 (matrix, rotation, result) | 乘以一个转换矩阵（底行为 [0.0，0.0，0.0，1.0] ）由3x3旋转矩阵组成。这是一个优化对于 Matrix4.multiply（m，Matrix4.fromRotationTranslation（rotation），m）; 而言，分配和运算量较少。 | matrix：Matrix4，左侧的矩阵。  rotation：Matrix4，右侧的3\*3旋转矩阵。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| multiplyByPoint (matrix, cartesian, result) | 计算矩阵与 [Cartesian3](http://cesium.xin/cesium/cn/Documentation1.72/Cartesian3.html)的乘积。这等效于调用 [Matrix4.multiplyByVector](http://cesium.xin/cesium/cn/Documentation1.72/Matrix4.html" \l ".multiplyByVector)且 [Cartesian4](http://cesium.xin/cesium/cn/Documentation1.72/Cartesian4.html)的 w 组件为1，但返回 [Cartesian3](http://cesium.xin/cesium/cn/Documentation1.72/Cartesian3.html)而不是 [Cartesian4](http://cesium.xin/cesium/cn/Documentation1.72/Cartesian4.html)。 | matrix：Matrix4，矩阵。  cartesian：Cartesian3，重点。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| multiplyByPointAsVector (matrix, cartesian, result) | 计算矩阵与 [Cartesian3](http://cesium.xin/cesium/cn/Documentation1.72/Cartesian3.html)的乘积。这等效于调用 [Matrix4.multiplyByVector](http://cesium.xin/cesium/cn/Documentation1.72/Matrix4.html" \l ".multiplyByVector)且 [Cartesian4](http://cesium.xin/cesium/cn/Documentation1.72/Cartesian4.html)的 w 分量为零。 | matrix：Matrix4，矩阵。  cartesian：Cartesian3，重点。  result：Cartesian3，将结果存储到的对象。 | Cartesian3 |
| multiplyByScalar (matrix, scalar, result) | 计算矩阵与标量的乘积。 | matrix：Matrix4，矩阵。  scalar：Number，要乘以的数字。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| multiplyByScale (matrix, scale, result) | 乘以仿射变换矩阵（底行为 [0.0，0.0，0.0，1.0] ）通过隐式非均匀比例矩阵。这是一个优化对于 Matrix4.multiply（m，Matrix4.fromUniformScale（scale），m）; ，其中 m 必须是仿射矩阵。此函数执行较少的分配和算术运算。 | matrix：Matrix4，左侧的矩阵。  scalar：Cartesian3，比例尺在右侧。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| multiplyByTranslation (matrix, translation, result) | 乘以一个转换矩阵（底行为 [0.0，0.0，0.0，1.0] ）由 [Cartesian3](http://cesium.xin/cesium/cn/Documentation1.72/Cartesian3.html)定义的隐式转换矩阵。这是一个优化对于 Matrix4.multiply（m，Matrix4.fromTranslation（position），m）; 而言，分配和运算量较少。 | matrix：Matrix4，左侧的矩阵。  translation：Cartesian3，比例尺在右侧。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| multiplyByUniformScale (matrix, scale, result) | 乘以仿射变换矩阵（底行为 [0.0，0.0，0.0，1.0] ）通过隐式均匀比例矩阵。这是一个优化对于 Matrix4.multiply（m，Matrix4.fromUniformScale（scale），m）; ，其中 m 必须是仿射矩阵。此函数执行较少的分配和算术运算。 | matrix：Matrix4，左侧的仿射矩阵。  scale：Number，右侧的统一比例尺。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| multiplyByVector (matrix, cartesian, result) | 计算矩阵与列向量的乘积。 | matrix：Matrix4，矩阵。  cartesian：Cartesian4，向量。  result：Cartesian4，将结果存储到的对象。 | Cartesian4 |
| multiplyTransformation (left, right, result) | 假设两个矩阵的乘积为2，则计算两个矩阵的乘积仿射变换矩阵，其中左上3x3元素是旋转矩阵，第四个元素的前三个元素专栏是翻译。底行假定为[0，0，0，1]。矩阵未验证为正确形式。这种方法比计算一般4x4产品的速度更快使用 [Matrix4.multiply](http://cesium.xin/cesium/cn/Documentation1.72/Matrix4.html" \l ".multiply)的矩阵。 | left：Matrix4，第一个矩阵。  right：Matrix4，第二个矩阵。  result：Cartesian4，将结果存储到的对象。 | Matrix4 |
| negate (matrix, result) | 创建所提供矩阵的求反副本。 | matrix：Matrix4，求反的矩阵。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| pack (value, array, startingIndex ) | 将提供的实例存储到提供的数组中。 | value：Matrix4，要打包的值。  array：Array.<Number>，要打包的数组。  startingIndex：Number，开始打包元素的数组索引。 | Array.<Number> |
| setColumn (matrix, index, cartesian, result) | 计算一个新矩阵，用提供的Cartesian4实例替换提供的矩阵中的指定列。 | matrix：Matrix4，使用的矩阵。  index：Number，要设置的列的从零开始的索引。  cartesian：Cartesian4，直角坐标系，其值将分配给指定的列。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| setRow (matrix, index, cartesian, result) | 计算一个新矩阵，用提供的Cartesian4实例替换提供的矩阵中的指定行。 | matrix：Matrix4，使用的矩阵。  index：Number，要设置的行的从零开始的索引。  cartesian：Cartesian4，直角坐标系，其值将分配给指定的行。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| setScale (matrix, scale, result) | 计算一个新矩阵，用提供的比例尺替换比例尺。假设矩阵是仿射变换 | matrix：Matrix4，使用的矩阵。  scale：Cartesian3，该比例尺代替了所提供矩阵的比例尺。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| setTranslation (matrix, translation, result) | 计算一个新矩阵，该矩阵替换提供的最右边一列中的转换具有提供的翻译的矩阵。假设矩阵是仿射变换。 | matrix：Matrix4，使用的矩阵。  translation：Cartesian3，该比例尺代替了所提供矩阵的比例尺。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| subtract (left, right, result) | 计算两个矩阵的差。 | left：Matrix4，第一个矩阵。  right：Matrix4，第二个矩阵。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| toArray (matrix, result ) | 从提供的Matrix4实例创建一个数组。该数组将按列优先顺序排列。 | matrix,：Matrix4，要使用的矩阵。  result：Array.<Number>，将结果存储到的对象。 | Array.<Number> |
| transpose (matrix, result) | 计算提供的矩阵的转置。 | matrix,：Matrix4，要转置的矩阵。  result：Matrix4，将结果存储到的对象。 | Matrix4 |
| unpack (array, startingIndex , result ) | 从压缩数组中检索实例。 | array：Array.<Number>，压缩数组。  startingIndex：Number，要解压缩的元素的起始索引。  result：Matrix4，将结果存储到的对象。 | Matrix4 |

## ScreenSpaceEventType类

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| LEFT\_DOWN | 表示鼠标左键按下事件 | Number |
| LEFT\_UP | 代表鼠标左键按下事件 | Number |
| LEFT\_CLICK | 表示鼠标左键单击事件 | Number |
| LEFT\_DOUBLE\_CLICK | 表示鼠标左键双击事件 | Number |
| RIGHT\_DOWN | 表示鼠标右键按下事件 | Number |
| RIGHT\_UP | 代表鼠标右键单击事件 | Number |
| RIGHT\_CLICK | 代表鼠标右键单击事件 | Number |
| MIDDLE\_DOWN | 表示鼠标中键按下事件 | Number |
| MIDDLE\_UP | 代表鼠标中键按下事件 | Number |
| MIDDLE\_CLICK | 代表鼠标中键事件 | Number |
| MOUSE\_MOVE | 表示鼠标移动事件 | Number |
| WHEEL | 代表鼠标滚轮事件 | Number |
| PINCH\_START | 表示触摸屏上的两指事件的开始 | Number |
| PINCH\_END | 表示触摸屏上两指事件的结束 | Number |
| PINCH\_MOVE | 表示触摸表面上两个手指事件的变化 | Number |

### 构造函数

无

### 方法

无

# DataSources主类

## DatasourceCollection类

实例的集合。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| DataSourceCollection () | 实例的集合 |  |

### 成员变量

无

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| add (dataSource) | 将数据源添加到集合中。 | Datasource：数据源或对数据源的承诺以添加到集合中。 通过承诺时，实际上不会添加数据源 直到承诺成功解决为止。 | 将数据源添加到集合后便会解决的Promise |
| remove (dataSource, destroy ) | 从此集合中删除数据源（如果存在） | dataSource：要删除的数据源；  destroy：除删除数据源外是否还要销毁数据源 | 如果数据源在集合中并已删除，则为true， 如果数据源不在集合中，则返回false |
| removeAll ( destroy ) | 从此集合中删除所有数据源 | destroy：除删除数据源外是否还要销毁数据源 |  |

## Entity类

实体实例将多种形式的可视化聚集到单个高级对象中。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Entity ( options ) | 实体实例将多种形式的可视化聚集到单个高级对象中。可以手动创建它们并将其添加到 Viewer＃entities 或由数据源，例如 CzmlDataSource 和 GeoJsonDataSource 。 | Options：描述初始化选项的对象 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| id | 获取与此对象关联的唯一ID。 | String |
| show | 获取或设置是否应显示此实体。设为true时，仅当父实体的show属性也为true时，才显示该实体。 | Boolean |
| name | 获取或设置对象的名称 | String/undefined |
| parent | 获取或设置父对象 | Entity/undefined |
| point | 获取或设置点图形 | Point/undefined |
| polyline | 获取或设置折线 | Polyline/undefined |
| polygon | 获取或设置多边形 | Polygon/undefined |
| label | 获取包含实体位置的二维标签 | Label/undefined |
| ellipsoid | 获取笛卡尔坐标系中由等式定义的二次曲面 | Ellipsoid/undefined |
| rectangle | 获取指定为经度和纬度坐标的二维区域 | Rectangle/undefined |
| billboard | 获取在3D场景中定位的视口对齐图像 | Billboard/undefined |
| box | 获取一个盒子 | Box/undefined |
| ellipse | 获取由中心点，半长轴和半短轴定义的椭圆 | Ellipse/undefined |
| plot | 获取军标 | Plot/undefined |
| cylinder | 获取圆柱体，截头圆锥体或由长度，顶部半径和底部半径定义的圆锥体 | Cylinder/undefined |
| overlayImage | 获取贴地图片 | OverlayImage/undefined |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| addProperty (propertyName) | 向此对象添加属性。添加属性后，就可以用 Entity＃definitionChanged 观察并合成与 CompositeEntityCollection | propertyName：要添加的属性的名称。 |  |
| merge(source) | 将此对象上每个未分配的属性分配给该值提供的源对象具有相同属性 | source：要合并到该对象中的对象 |  |

## EntityCollection类

Entity实例的可观察集合。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| EntityCollection ( owner ) | 每个实体都有唯一ID的 Entity 实例的可观察集合。 | Owner：创建此集合的数据源（或复合实体集合）。 |

### 成员变量

无

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| add (entity) | 将实体添加到集合中。 | entity：要添加的实体 | 添加的实体 |
| remove (entity) | 从集合中删除实体 | entity：要删除的实体 | 如果已删除该项目，则为true；如果该集合中不存在该项目，则为false |
| removeAll () | 从集合中删除所有实体。 |  |  |

## GeoJsonDatasource类

GeoJson数据源。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| GeoJsonDataSource ( name ) | DataSource 可以同时处理两个 GeoJSON 和 TopoJSON 数据。如果 simplestyle-spec 属性也将使用存在 | Name：string，该数据源的名称。如果未定义，则名称将取自 GeoJSON文件的名称。 |

### 成员变量

无

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| Load(data, options) | 异步加载提供的GeoJSON或TopoJSON数据，替换任何现有数据。 | Data：要加载的网址，GeoJSON对象或TopoJSON对象。  Options：具有sourceUrI或描述等属性的对象。  sourceUrl：覆盖用于解析相对链接的网址；  markerSise：为每个点创建的地图图钉的默认大小（以像素为单位）；  markerSymbol：为每个点创建的地图图钉的默认符号；  markerColor：为每个点创建的地图图钉的默认颜色；  strokeWidth：折线和多边形轮廓的默认宽度；  clampToGround：如果我们希望将要素固定在地面上，则为true | 一个承诺，将在加载GeoJson时解决 |

## Model类

基于glTF的3D模型。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Model ( options ) | 基于glTF的3D模型 | options：描述初始化选项的对象 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| scale | 获取或设置数字属性，指定统一的线性比例对于此模型。大于1.0的值会增加模型的大小，而小于1.0的值会减少它 | Property/undefined |
| show | 获取或设置指定模型可见性的布尔属性 | Property/undefined |
| uri | 获取或设置字符串Property，该字符串指定glTF资产的URI | Property/undefined |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| clone ( result ) | 复制此实例 | result：将结果存储到的对象 | 修改后的结果参数或一个新实例（如果未提供） |
| merge(source) | 将此对象上每个未分配的属性分配给该值提供的源对象具有相同属性 | source:要合并到该对象中的对象 |  |

## Point类

描述位于包含Entity的位置的图形点。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Point ( options ) | 描述位于包含 Entity 的位置的图形点 | options：描述初始化选项的对象 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| color | 获取或设置指定该点的 Color 的属性 | Property/undefined |
| distanceDisplayCondition | 获取或设置 DistanceDisplayCondition 属性 | Property/undefined |
| heightReference | 获取或设置指定 HeightReference 的属性 | Property/undefined |
| pixelSize | 获取或设置数字属性，以像素为单位指定大小 | Property/undefined |
| show | 获取或设置指定点的可见性的布尔属性 | Property/undefined |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| clone ( result ) | 复制此实例 | result：将结果存储到的对象 | 修改后的结果参数或一个新实例（如果未提供） |
| merge (source) | 将此对象上每个未分配的属性分配给该值提供的源对象具有相同属性 | source：要合并到该对象中的对象 |  |

## Polygon类

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Polygon ( options ) | 描述由构成外部形状和任何嵌套孔的线性环的层次结构定义的多边形 | options：描述初始化选项的对象 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| fill | 获取或设置布尔属性，该属性指定多边形是否被提供的材质填充 | Property/undefined |
| height | 获取或设置数字属性，该属性指定多边形的恒定高度 | Property/undefined |
| show | 获取或设置指定多边形可见性的布尔属性 | Property/undefined |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| clone ( result ) | 复制此实例 | result：将结果存储到的对象 | 修改后的结果参数或一个新实例（如果未提供） |
| merge (source) | 将此对象上每个未分配的属性分配给该值提供的源对象具有相同属性 | source：要合并到该对象中的对象 |  |

## Polyline类

描述折线。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Polyline ( options ) | 描述折线。前两个位置定义线段，并且每个其他位置都从前一个位置定义了一个线段 | options：描述初始化选项的对象 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| arcType | 获取或设置 ArcType 属性，该属性指定线段应该是大弧线，菱形线还是线性连接 | Property/undefined |
| classificationType | 获取或设置 ClassificationType 属性，该属性指定此折线在地面上时是否对地形，3D瓷砖或两者进行分类 | Property/undefined |
| clampToGround | 获取或设置布尔属性，该布尔属性指定是否折线应固定在地面上 | Property/undefined |
| positions | 获取或设置指定 Cartesian3 数组的属性定义线条的位置 | Property/undefined |
| show | 获取或设置指定折线可见性的布尔属性 | Property/undefined |
| width | 获取或设置数字属性，以像素为单位指定宽度 | Property/undefined |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| clone ( result ) | 复制此实例 | result：将结果存储到的对象 | 修改后的结果参数或一个新实例（如果未提供） |
| merge (source) | 将此对象上每个未分配的属性分配给该值提供的源对象具有相同属性 | source：要合并到该对象中的对象 |  |

## Label类

描述位于包含实体位置的二位标签。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Label (options) | 描述位于包含实体位置的二维标签 | options：描述初始化选项的对象 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| backgroundColor | 获取或设置指定背景颜色的属性 | Property/undefined |
| backgroundPadding | 获取或设置Cartesian2属性，以像素为单位指定标签的水平和垂直背景填充 | Property/undefined |
| eyeOffset | 获取或设置Cartesian3属性，该属性指定标签的眼坐标偏移量 | Property/undefined |
| fillcolor | 获取或设置指定填充颜色的属性 | Property/undefined |
| style | 获取或设置指定Label的属性 | Property/undefined |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| clone ( result ) | 复制此实例 | result：将结果存储到的对象 | 修改后的结果参数或一个新实例（如果未提供） |
| merge (source) | 将此对象上每个未分配的属性分配给该值提供的源对象具有相同属性 | source：要合并到该对象中的对象 |  |

## Rectangle类

指定经度和纬度坐标的二维区域。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Rectangle ( west , south , east , north ) | 指定为经度和纬度坐标的二维区域 | west：最西的经度，以弧度表示，在[-Pi，Pi]范围内；  south：最南的经度，以弧度表示，在[-Pi/2，Pi/2]范围内；  east：最东的经度，以弧度表示，在[-Pi，Pi]范围内；  north：最北的经度，以弧度表示，在[-Pi/2，Pi/2]范围内； |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| height | 获取以弧度为单位的矩形的高度 | Number |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| clone ( result ) | 复制此矩形 | result：将结果存储到的对象 | 修改后的结果参数或新的Rectangle实例（如果未提供） |

## Billboard类

在3D场景中定位的视口对齐图像。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Billboard () | 在3D场景中定位的视口对齐图像 |  |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| color | 获取或设置与广告牌纹理相乘的颜色 | Color |
| height | 获取或设置广告牌的高度。如果未定义，将使用图像高度 | Number |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| equals (other) | 确定此广告牌是否等于另一个广告牌 | other：用于平等性比较的广告牌 | 广告牌相等为true，其余都为false |
| setImage (id, image) | 设置要用于此广告牌的图像 | id：图片的ID。这可以是任何唯一标识图像的字符串；  image：要加载的图像。 |  |
| setImageSubRegion (id, subRegion) | 将具有给定id的图像的子区域用作此广告牌的图像，从左下角开始以像素为单位进行测量 | id：要使用的图片的id；  subregion：图片的子区域 |  |

## Box类

描述一个盒子。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Box(options) | 描述一个盒子 | options：描述初始化选项的对象 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| dimensions | 获取或设置 Cartesian3 属性属性，该属性指定框的长度，宽度和高度 | Property/undefined |
| show | 获取或设置指定框的可见性的布尔属性 | Property/undefined |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| clone ( result ) | 复制此实例 | result： 将结果存储到的对象 | 修改后的结果参数或一个新实例（如果未提供） |
| merge (source) | 将此对象上每个未分配的属性分配给该值提供的源对象具有相同属性 | source：要合并到该对象中的对象 |  |

## Plot类

J标。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Plot() | J标 |  |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| linewidth | 线宽 | Number |
| lineColor | 线颜色 | Color |
| OutLineColor | 衬线颜色 | Color |
| OutLineWidth | 衬线宽度 | Number |
| FillMode | 填充模式 | Number |
| Size | 符号大小 | Number |

### 方法

无

# Scene主类

## ArcGISMapServerImageryProvider类

提供由ArcGIS MapServer托管的平铺图像。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| ArcGisMapServerImageryProvider (options) | 提供由ArcGIS MapServer托管的平铺图像 | options:描述初始化选项的对象 |

### 成员变量

无

### 方法

无

## Camera类

相机由位置、方向和视锥台定义。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Camera (scene) | 相机由位置，方向和视锥台定义 | scene：现场 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| heading | 获取摄像机的弧度方向 | Number |
| Pitch | 获取相机的弧度 | Number |
| roll | 获取相机的弧度角 | Number |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| zoomIn(amount) | 沿相机的视图矢量缩放 | amount：移动量 |  |
| zoomOut(amount) | 沿相反方向缩放相机的视角向量 | amount：移动量 |  |
| move(direction, amount) | 沿方向转换摄像机的位置 | direction:移动的方向  amount：移动的量（以米为单位） |  |
| flyHome(duration) | 将相机飞到主视图 | duration：飞行持续时间（以秒为单位） |  |
| flyTo(options) | 将相机从当前位置移动到新位置 | options  destination：摄像机在WGS84（世界）坐标中的最终位置或从上向下视图中可见的矩形  orientation：包含方向和向上属性或航向，俯仰和横滚属性的对象  duration：飞行持续时间（以秒为单位） |  |
| lookAt(target, offset) | 使用目标和偏移量设置摄像机的位置和方向 | target：世界坐标中的目标位置  offset：在局部东北朝上的参考框中，距目标的偏移量为中心 |  |

## Fog类

将大气与远离摄像机的几何体融合在一起，以获取地平线视图。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Fog () | 将大气与远离摄像机的几何体融合在一起，以获取地平线视图 |  |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| enabled | 如果启用雾，则为 true ，否则为 false | Boolean |

### 方法

无

## Globe类

场景中渲染的地球，包括其地形和图像图层。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Globe(ellipsoid) | 确定地球的尺寸和形状 | Ellipsoid 笛卡尔坐标系中由等式定义的二次曲面  主要用于表示行星体的形状 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| show | 确定是否显示地球 | Boolean |
| ellipsoid | 获取描述此地球形状的椭圆体 | Ellipsoid |

### 方法

无

## GoogleEarthEnterpriseMapsProvider类

使用谷歌地球图像API提供平铺图像。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| GoogleEarthEnterpriseMapsProvider(options) | 使用谷歌地球图像API提供平铺图像 | options：对象，描述初始化选项 |

### 成员变量

无

### 方法

无

## ImageryLayerCollection类

图像图层的有序集合。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| ImageryLayerCollection () | 图像图层的有序集合 |  |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| length | 获取此集合中的层数 | Number |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| contains (layer) | 检查集合是否包含给定的图层 | layer：要检查的层 | 如果集合包含图层，则为true，否则为false |
| add (layer, index ) | 在集合中添加一个图层 | layer：要添加的层；  index：在其上添加图层的索引。如果省略，该层将 添加到所有现有图层之上 |  |
| addImageryProvider (imageryProvider, index ) | 使用给定的ImageryProvider创建一个新层，并将其添加到集合中 | imageryProvider：图像提供者为其创建新图层；  index：在其上添加图层的索引。如果省略，该层将 添加在所有现有图层之上 | 新创建的图层 |
| remove (layer, destroy ) | 从该集合中删除一个图层（如果存在） | layer：要删除的图层；  destroy：除了删除图层外，是否还要破坏这些图层 | 如果该图层在集合中并已被删除，则为true， 如果图层不在集合中，则返回false |

## Layer3DCollection类

三维图层的有序集合。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Layer3DCollection (url) | 三维图层的有序集合 | url：三维图层地址 |

### 成员变量

无

### 方法

无

## MapboxImageryProvider类

提供由Mapbox托管的平铺图像。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| MapboxImageryProvider (options) | 提供由Mapbox托管的平铺图像 | options：描述初始化选项的对象 |

### 成员变量

无

### 方法

无

## MapboxVectorTileProvider类

Mapbox矢量瓦片图层。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| MapboxVectorTileProvider (options) | 提供由Mapbox托管的平铺图像 | options：描述初始化选项的对象 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| Url | 获取或设置指向要在credit中显示的谷歌Earth徽标的URL | String |

### 方法

无

## OSGB3DLayer类

OSGB图层的有序集合。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| OSGB3DLayer (url) | OSGB图层的有序集合 | url：OSGB数据地址 |

### 成员变量

无

### 方法

无

## Scene类

虚拟场景中所有3D图形对象和状态的容器。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Scene(options) | 虚拟场景中所有3D图形对象和状态的容器 | canvas：用于为其创建场景的HTMLcanvas元素；  contextOptions：上下文和WebGL创建属性；  creditContainer：将在其中显示积分的HTML元素  creditViewport：在其中显示功劳弹出窗口的HTML元素。如果未指定，则视口将被添加为画布的同级；  mapProjection：在2D和  Columbus View模式下使用的地图投影；  orderIndependentTranslucency：如果为true并且配置支持它，则使用顺序无关的半透明性；  scene3DOnly：如果为true，则可以优化3D模式的内存使用和性能，但会禁用使用2D或Columbus View的功能；  terrainExaggeration：用于放大地形的标量；  shadows：确定阴影是否由光源投射；  mapMode2D：确定2D地图是可旋转的还是可以在水平方向无限滚动；  requestRenderMode：如果为true，则仅根据场景中的更改确定是否需要渲染帧；  maximumRenderTimeChange：如果requestRenderMode为true，则此值定义在请求渲染之前允许的最大仿真时间更改。 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| skyAtmosphere | 遍布全球的天空气氛 | SkyAtmosphere |
| skyBox | 天空盒，用于绘制星星 | SkyBox |
| sun | 太阳 | Sun |
| fog | 将大气与远离摄像机的几何体融合在一起，以获取地平线视图。 | Fog |
| preUpdate | 获取在更新或渲染场景之前将引发的事件 | Event |
| postUpdate | 获取在场景更新之后和场景渲染之前立即引发的事件 | Event |
| preRender | 获取在场景更新之后以及场景渲染之前立即引发的事件 | Event |
| postRender | 获取将在渲染场景后立即引发的事件 | Event |
| mode | 获取或设置场景的当前模式 | SceneMode |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| pick (windowPosition, width , height ) | 返回具有' primitive'属性的对象，该对象包含场景中的第一个（顶部）基本体在特定的窗口坐标处；如果位置不存在，则为undefined。 | windowPosition：2D笛卡尔点  width：拾取矩形的宽度  height：拾取矩形的高度 | 包含选取的图元的对象：Object |
| sampleHeight (position ) | 返回给定制图位置处场景几何的高度；如果没有，则返回 undefined 场景几何要从其采样高度 | position：由经度，纬度和高度定义的位置 | 高度：Number |

## SceneMode类

### 构造函数

无

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| SCENE\_2D | 2D模式。可以使用正交投影从上至下查看该地图 |  |
| SCENE\_3D | 3D模式。地球的传统3D透视图 |  |

### 方法

无

## SingleTileImageryProvider类

提供单个顶层图像图块。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| SingleTileImageryProvider (options) | 提供单个顶层图像图块 | options：描述初始化选项的对象 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| Url | 获取或设置指向要在credit中显示的谷歌Earth徽标的URL | String |

### 方法

无

## SkyAtmosphere类

提供椭圆体四肢周围绘制的气氛。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| SkyAtmosphere ( ellipsoid ) | 提供椭圆体四肢周围绘制的气氛 | ellipsoid：周围的椭圆形 |

### 成员变量

无

### 方法

无

## SkyBox类

场景周围的天空盒可绘制星星。使用'真赤道平均春分（TEME）'轴定义天空盒。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| SkyBox (options) | 场景周围的天空盒可绘制星星。使用'真赤道平均春分（TEME）'轴定义天空盒 | options  资源：六个多维数据集映射面中每个面的源URL或 Image 对象  show：确定是否将显示此原语 |

### 成员变量

无

### 方法

无

## Sun类

绘制一个阳光广告牌。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Sun () | 绘制一个阳光广告牌 |  |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| glowFactor | 获取或设置一个数字，该数字控制太阳的镜头眩光的显示方式成为'明亮'。零仅显示没有任何耀斑的太阳盘。使用较大的值可以使太阳周围的耀斑更明显 | Number |
| show | 确定是否可以显示太阳 | Boolean |

### 方法

无

## TileMapServiceImageryProvider类

图像提供者，提供生成的平铺图像。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| TileMapServiceImageryProvider (options) | 图像提供者，提供生成的平铺图像 | options：描述初始化选项的对象 |

### 成员变量

无

### 方法

无

## Tileset3Dlayer类

三维瓦片图层的有序集合。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Tileset3Dlayer(url) | 三维瓦片图层的有序集合 | url：三维瓦片数据地址 |

### 成员变量

无

### 方法

无

## UrlTemplateImageryProvider类

通过使用指定的URL模板请求图块来提供图像。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| UrlTemplateImageryProvider (options) | 通过使用指定的URL模板请求图块来提供图像 | options：描述初始化选项的对象 |

### 成员变量

无

### 方法

无

## VectorTileLayerCollection类

矢量瓦片图层的有序集合。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| VectorTileLayerCollection () | 矢量瓦片图层的有序集合 |  |

### 成员变量

无

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| add (layer, index ) | 在集合中添加一个图层 | layer：要添加的层；  index：在其上添加图层的索引。如果省略，该层将 添加到所有现有图层之上 |  |
| addImageryProvider (imageryProvider, index ) | 使用给定的ImageryProvider创建一个新层，并将其添加到集合中 | imageryProvider：图像提供者为其创建新图层；  index：在其上添加图层的索引。如果省略，该层将 添加在所有现有图层之上 | 新创建的图层 |

## Viewer类

用于构建应用程序的基本窗口小部件。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| Viewer (container, options ) | 用于构建应用程序的基本窗口小部件 | container：将包含窗口小部件的DOM或ID  options：描述初始化选项的对象 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| canvas | 获取画布 | HTMLCanvasElement |
| container | 获取父容器 | Element |
| shadows | 确定阴影是否由光源投射 | Boolean |
| terrainProvider | 确定地形是投射光源还是投射阴影 | TerrainProvider |

### 方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 说明 | 参数 | 返回值 |
| flyTo(target, options) | 将相机移至提供的一个或多个实体或数据源 | target: 实体，实体阵列，实体集合，数据源，Cesium3DTileset，点云或要查看的图像层;  options  持续时间：飞行持续时间（秒），  maximumHeight：飞行高峰时的最大高度，  offset：在局部东北朝上的参考框中，距目标的偏移量为中心 | 如果飞行成功则解析为true的承诺，如果当前未在场景中可视化目标或取消飞行，则为false的Promise |
| resize() | 调整窗口小部件的大小以匹配容器的大小 |  |  |

## WebMapServiceImageryProvider类

提供由Web地图服务（WMS）服务器托管的平铺图像。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| WebMapServiceImageryProvider(options) | 提供由Web地图服务(WMS)服务器托管的平铺图像 | options：描述初始化选项的对象 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| Url | 获取或设置指向要在credit中显示的谷歌Earth徽标的URL | String |
| channel | 获取当前正在使用的图像通道（id） | Number |
| path | 获取谷歌Earth服务器上数据的url路径 | String |

### 方法

无

## WebMapTileServiceImageryProvider类

WMS服务矢量瓦片。

### 构造函数

|  |  |  |
| --- | --- | --- |
| 构造函数 | 说明 | 参数说明 |
| WebMapTileServiceImageryProvider (options) | WMS服务矢量瓦片 | options：描述初始化选项的对象 |

### 成员变量

|  |  |  |
| --- | --- | --- |
| 成员变量 | 说明 | 类型 |
| Url | 获取或设置指向要在credit中显示的谷歌Earth徽标的URL | String |

### 方法

无