



河北师范大学软件学院  
Software College of Hebei Normal University

# 百度地图实战



Android教研室



# 自我介绍

丁盟

qq: 2622885094





## ● 覆盖物介绍

### ● 标注覆盖物

#### ● 几何图形覆盖物

#### ● 文字覆盖物

#### ● 地形图图层覆盖物

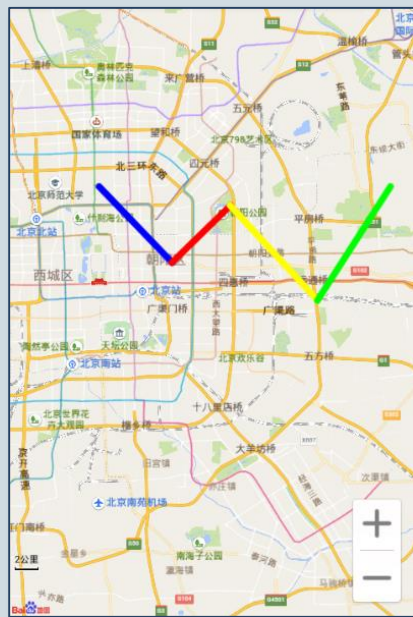
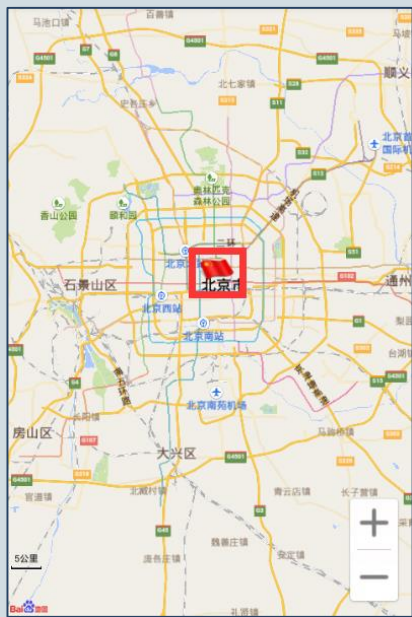
### ● 弹出窗覆盖物





## 覆盖物介绍

❖ 所有叠加或覆盖到地图的内容，我们统称为**地图覆盖物**。如文本、标注、几何图形、信息窗口等。





## 覆盖物介绍

- ❖ 在地图上添加覆盖物，一般需要以下几个步骤：
  1. 定义坐标点，有可能是一个，有可能是多个（比如：多边形覆盖物）。
  2. 构造**OverlayOptions**（地图覆盖物选型基类）。
  3. 在地图上添加覆盖物。
  4. 添加相应的监听事件。



## 覆盖物介绍 - 坐标点

### ❖ Class LatLng

- 地理坐标基本数据结构。

```
// 构造函数
```

```
public LatLng(double latitude,  
              double longitude)
```

```
latitude - 纬度
```

```
longitude - 经度
```



## 覆盖物介绍 - 地图上添加覆盖物

- ❖ 在地图上添加覆盖物通过 **BaiduMap** 类的 **addOverlay()** 方法。

```
// 向地图添加一个 Overlay
```

```
Overlay addOverlay(OverlayOptions options)
```

Parameters:

options - 将要添加的 overlay 选项

Returns:

被添加的 overlay 对象



## 覆盖物介绍 - 覆盖物选项类

- ❖ 想要在地图上添加覆盖物需要一个OverlayOptions类型的参数，**OverlayOptions**本身是一个**抽象类**，想要添加覆盖物则**必须实例化其子类**。

### OverlayOptions

ArcOptions	弧线构造选项类
CircleOptions	创建圆的选项类
DotOptions	创建圆点的选项类
GroundOverlayOptions	地形图图层选型类
MarkerOptions	标注覆盖物选型类
PolygonOptions	多边形选型类
PolylineOptions	折线选型类
TextOptions	文本选型类





## 覆盖物介绍 - 地图覆盖物类

- ❖ 通过BaiduMap类的addOverlay()方法可以获得一个Overlay对象，**Overlay**是一个**抽象类**，所有地图覆盖物的**基类**。

Overlay	Arc	弧线类
	Circle	圆类
	Dot	点类
	GroundOverlay	图层类
	Marker	标注类
	Polygon	多边形类
	Polyline	折线类
	Text	文本类



## 覆盖物介绍



### 标注覆盖物



### 几何图形覆盖物



### 文字覆盖物



### 地形图图层覆盖物



### 弹出窗覆盖物





## 标注覆盖物 - 标注覆盖物选项类

### ❖ Class MarkerOptions - 标注覆盖物选项类

MarkerOptions <b>alpha</b> (float alpha)	设置 marker 覆盖物图标透明度，取值为[0,1]，默认1.0，若超出范围则默认为1.0
MarkerOptions <b>draggable</b> (boolean draggable)	设置 marker 是否允许拖拽，默认不可拖拽
MarkerOptions <b>icon</b> (BitmapDescriptor icon)	设置 marker 覆盖物的图标
MarkerOptions <b>perspective</b> (boolean perspective)	设置是否开启 marker 覆盖物近大远小效果，默认开启
MarkerOptions <b>position</b> (LatLng position)	设置 marker 覆盖物的位置坐标
MarkerOptions <b>rotate</b> (float rotate)	设置 marker 覆盖物旋转角度，逆时针
MarkerOptions <b>title</b> (java.lang.String title)	设置 marker 覆盖物的标题
MarkerOptions <b>visible</b> (boolean visible)	设置 marker 覆盖物的可见性



## 标注覆盖物 - bitmap 描述信息工厂类

❖ Class **BitmapDescriptorFactory** - bitmap 描述信息工厂类

注意：在使用该类方法之前请确保已经调用了

**SDKInitializer.initialize(Context)** 函数以提供全局 Context 信息

static BitmapDescriptor <b>fromAsset</b> (java.lang.String assetName)	根据资源名称创建bitmap描述信息
static BitmapDescriptor <b>fromAssetWithDpi</b> (java.lang.String assetName)	根据资源名称和dpi创建bitmap描述信息，根据不同设备的dpi，对asset下图片等比例缩放
static BitmapDescriptor <b>fromBitmap</b> (Bitmap image)	根据 Bitmap 创建描述信息
static BitmapDescriptor <b>fromFile</b> (java.lang.String fileName)	根据应用程序私有文件夹里包含文件的文件名创建 bitmap 描述信息
static BitmapDescriptor <b>fromPath</b> (java.lang.String absolutePath)	根据文件绝对路径创建 bitmap 描述信息
static BitmapDescriptor <b>fromResource</b> (int resourceId)	根据资源 Id 创建 bitmap 描述信息
static BitmapDescriptor <b>fromView</b> (View view)	根据一个 View 创建 Bitmap 描述信息, 当 view 为 null 时返回 null



## 标注覆盖物

### ❖ 示例代码

```
// 定义Maker坐标点
LatLng point = new LatLng(39.915, 116.404);
// 构建Marker图标
BitmapDescriptor bitmap = BitmapDescriptorFactory
    .fromResource(R.drawable.flag);
// 构建MarkerOption, 用于在地图上添加Marker
OverlayOptions option = new MarkerOptions()
    .position(point)// 设置marker的位置
    .icon(bitmap); // 必须设置marker图标
//在地图上添加Marker, 并显示
Marker marker = (Marker)mBaiduMap.addOverlay(option);
```



## 标注覆盖物 - 设置监听方法

### ❖ 设置监听方法

- static interface BaiduMap.**OnMarkerClickListener**
  - 地图 Marker 覆盖物点击事件监听接口
- static interface BaiduMap.**OnMarkerDragListener**
  - 地图 Marker 覆盖物拖拽事件监听接口



## 标注覆盖物 - 设置监听方法

### ➤ 设置点击事件监听

```
// 调用BaiduMap对象的setOnMarkerClickListener方法设置marker点击事件的监听
mBaiduMap.setOnMarkerClickListener(new OnMarkerClickListener()
{
    public boolean onMarkerClick(Marker marker) {
        // 点击处理
    }
});
```



## 标注覆盖物 - 设置监听方法

### ➤ 设置拖拽事件监听

// 调用BaiduMap对象的setOnMarkerDragListener方法设置marker拖拽事件的监听

```
mBaiduMap.setOnMarkerDragListener(new OnMarkerDragListener() {  
    public void onMarkerDrag(Marker marker) {  
        // 拖拽中  
    }  
    public void onMarkerDragEnd(Marker marker) {  
        // 拖拽结束  
    }  
    public void onMarkerDragStart(Marker marker) {  
        // 开始拖拽  
    }  
});
```





## 标注覆盖物 - 设置监听方法

### ➤ 示例代码

```
mBaiduMap.setOnMarkerClickListener(new BaiduMap
    .OnMarkerClickListener() {
    @Override
    public boolean onMarkerClick(Marker marker) {
        final LatLng latLng = marker.getPosition();
        Toast.makeText(MainActivity.this,
                        latLng.toString(),
                        Toast.LENGTH_SHORT)
            .show();

        return false;
    }
});
```



## 运行效果





## 覆盖物介绍



### 标注覆盖物



#### 几何图形覆盖物



#### 文字覆盖物



#### 地形图图层覆盖物



#### 弹出窗覆盖物





## 几何图形覆盖物

- ❖ 地图SDK提供多种结合图形覆盖物，利用这些图形，可帮助构建更加丰富多彩的地图应用。目前提供的几何图形有：点（Dot）、折线（Polyline）、弧线（Arc）、圆（Circle）、多边形（Polygon）。

<b>DotOptions</b>	创建圆点的选项类
<b>PolylineOptions</b>	折线选型类
<b>ArcOptions</b>	弧线构造选项类
<b>CircleOptions</b>	创建圆的选项类
<b>PolygonOptions</b>	多边形选型类



# 几何图形覆盖物 - 圆点的选项类

## ❖ Class **DotOptions** - 创建圆点的选项类

DotOptions <b>center</b> (LatLng center)	设置圆点的圆心坐标
DotOptions <b>color</b> (int color)	设置圆点的颜色
DotOptions <b>extraInfo</b> (Bundle extraInfo)	设置圆点的额外信息
DotOptions <b>radius</b> (int radius)	设置圆点的半径，单位：像素, 默认为 5px
DotOptions <b>visible</b> (boolean visible)	设置圆点是否可见
DotOptions <b>zIndex</b> (int zIndex)	设置圆点的 zIndex



## 几何图形覆盖物 - 折线覆盖物选项类

### ❖ Class PolylineOptions - 创建折线覆盖物选项类

PolylineOptions <b>color</b> (int color)	设置折线颜色
PolylineOptions <b>colorsValues</b> (java.util.List<java.lang.Integer> colors)	设置折线每段的颜色值
PolylineOptions <b>dottedLine</b> (boolean dottedLine)	设置折线是否虚线
PolylineOptions <b>extraInfo</b> (Bundle extraInfo)	设置折线额外信息
PolylineOptions <b>points</b> (java.util.List<LatLng> points)	设置折线坐标点列表
PolylineOptions <b>visible</b> (boolean visible)	设置折线可见性
PolylineOptions <b>width</b> (int width)	设置折线线宽，默认为 5，单位：像素
PolylineOptions <b>zIndex</b> (int zIndex)	设置折线 zIndex



## 几何图形覆盖物 - 弧线覆盖物选项类

### ❖ Class **ArcOptions** - 创建弧线覆盖物选项类

ArcOptions <b>color</b> (int color)	设置弧线的颜色
ArcOptions <b>extraInfo</b> (Bundle extraInfo)	设置弧线的额外信息
ArcOptions <b>points</b> (LatLng start, LatLng middle, LatLng end)	设置弧线的起点、中点、终点坐标
ArcOptions <b>visible</b> (boolean visible)	设置弧线是否可见
ArcOptions <b>width</b> (int width)	设置弧线的线宽
ArcOptions <b>zIndex</b> (int zIndex)	设置弧线的 zIndex



# 几何图形覆盖物 - 圆覆盖物选项类

## ❖ Class **CircleOptions** - 创建圆覆盖物选项类

CircleOptions <b>center</b> (LatLng center)	设置圆心坐标
CircleOptions <b>radius</b> (int radius)	设置圆半径
CircleOptions <b>extraInfo</b> (Bundle extraInfo)	设置圆额外信息
CircleOptions <b>fillColor</b> (int color)	设置圆填充颜色
CircleOptions <b>stroke</b> (Stroke stroke)	设置圆边框信息
CircleOptions <b>visible</b> (boolean visible)	设置圆是否可见
CircleOptions <b>zIndex</b> (int zIndex)	设置圆 zIndex 信息





## 几何图形覆盖物 - 多边形覆盖物选项类

### ❖ Class PolygonOptions - 创建多边形覆盖物选项类

PolygonOptions <b>extraInfo</b> (Bundle extraInfo)	设置多边形额外信息
PolygonOptions <b>fillColor</b> (int color)	设置多边形填充颜色
PolygonOptions <b>points</b> (java.util.List<LatLng> points)	设置多边形坐标点列表
PolygonOptions <b>stroke</b> (Stroke stroke)	设置多边形边框信息
PolygonOptions <b>visible</b> (boolean visible)	设置多边形可见性
PolygonOptions <b>zIndex</b> (int zIndex)	设置多边形 zIndex



## 几何图形覆盖物 - 示例：多边形覆盖物

```
// 定义多边形的五个顶点
LatLng pt1 = new LatLng(40.115, 116.404);
LatLng pt2 = new LatLng(39.915, 116.104);
LatLng pt3 = new LatLng(39.715, 116.304);
LatLng pt4 = new LatLng(39.715, 116.504);
LatLng pt5 = new LatLng(39.915, 116.704);
List<LatLng> pts = new ArrayList<LatLng>();
pts.add(pt1); pts.add(pt2); pts.add(pt3);
pts.add(pt4); pts.add(pt5);
// 构建用户绘制多边形的Option对象
OverlayOptions polygonOption = new PolygonOptions()
    .points(pts).stroke(new Stroke(5, 0xAA00FF00))
    .fillColor(0xAAAAFFFF00);
// 在地图上添加多边形Option，用于显示
mBaiduMap.addOverlay(polygonOption);
```



## 运行效果



## 几何图形覆盖物 - 示例：折线覆盖物

```
// 构造折线点坐标
List<LatLng> points = new ArrayList<LatLng>();
points.add(new LatLng(39.965,116.404));
points.add(new LatLng(39.925,116.454));
points.add(new LatLng(39.955,116.494));
points.add(new LatLng(39.905,116.554));
points.add(new LatLng(39.965,116.604));
// 构建分段颜色索引数组
List<Integer> colors = new ArrayList<>();
colors.add(Integer.valueOf(Color.BLUE));
colors.add(Integer.valueOf(Color.RED));
colors.add(Integer.valueOf(Color.YELLOW));
colors.add(Integer.valueOf(Color.GREEN));
OverlayOptions polyline = new PolylineOptions().width(20)
        .colorsValues(colors).points(points);
// 添加在地图中
mBaiduMap.addOverlay(polyline);
```



# 几何图形覆盖物 - 示例：折线覆盖物



运行效果





## 覆盖物介绍



### 标注覆盖物



### 几何图形覆盖物



### 文字覆盖物



### 地形图图层覆盖物



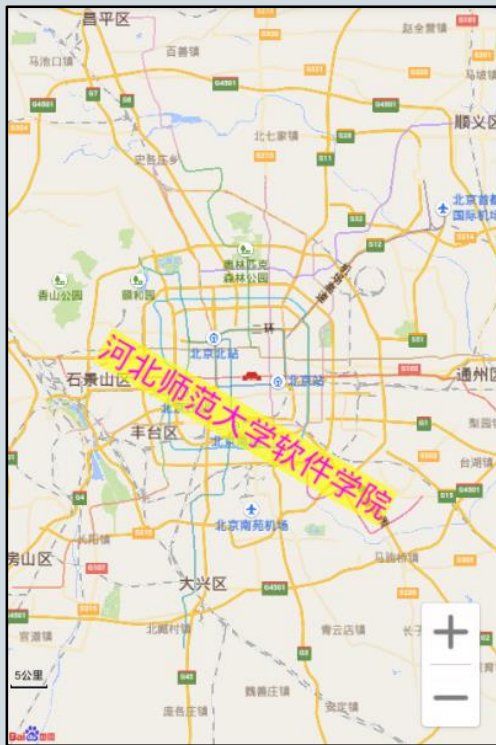
### 弹出窗覆盖物





## 文字覆盖物

- ❖ 文字，在地图中也是一种覆盖物，开发者可利用相关的接口，快速实现在地图上书写文字的需求。





# 文字覆盖物

## ❖ Class TextOptions - 创建文字覆盖物选项类

TextOptions <b>align</b> (int alignX, int alignY)	设置文字覆盖物对齐方式，默认居中对齐
TextOptions <b>bgColor</b> (int bgColor)	设置文字覆盖物背景颜色
TextOptions <b>extraInfo</b> (Bundle extraInfo)	设置文字覆盖物额外信息
TextOptions <b>fontColor</b> (int color)	设置文字覆盖物字体颜色，默认黑色
TextOptions <b>fontSize</b> (int size)	设置文字覆盖物字体大小
TextOptions <b>position</b> (LatLng position)	设置文字覆盖物地理坐标
TextOptions <b>rotate</b> (float rotate)	设置文字覆盖物旋转角度，逆时针
TextOptions <b>text</b> (java.lang.String text)	设置文字覆盖物的文字内容
TextOptions <b>typeface</b> (Typeface typeface)	设置文字覆盖物字体
TextOptions <b>visible</b> (boolean visible)	设置文字覆盖物可见性
TextOptions <b>zIndex</b> (int zIndex)	设置文字覆盖物 zIndex





## 文字覆盖物

### ➤ 示例代码

```
// 定义文字所显示的坐标点
LatLng llText = new LatLng(39.86923, 116.397428);
// 构建文字Option对象，用于在地图上添加文字
OverlayOptions textOption = new TextOptions()
    .bgColor(0xAAFFFF00)
    .fontSize(32)
    .fontColor(0xFFFF00FF)
    .text("河北师范大学软件学院")
    .rotate(-30)
    .position(llText);
// 在地图上添加该文字对象并显示
mBaiduMap.addOverlay(textOption);
```



## 覆盖物介绍



### 标注覆盖物



### 几何图形覆盖物



### 文字覆盖物



### 地形图图层覆盖物



### 弹出窗覆盖物





## 地形图图层覆盖物

- ❖ 地形图图层（GroundOverlay），又可叫做图片图层，即开发者可在地图的指定位置上添加图片。该图片可随地图的平移、缩放、旋转等操作做相应的变换。该图层是一种特殊的Overlay，它位于底图和底图标注层之间（即该图层不会遮挡地图标注信息）。





# 地形图图层覆盖物

## ❖ Class GroundOverlayOptions

### - 创建 ground 覆盖物的选项类

GroundOverlayOptions <b>image</b> (BitmapDescriptor image)	设置 Ground 覆盖物的图片信息
GroundOverlayOptions <b>positionFromBounds</b> (LatLngBounds bounds)	设置 Ground 覆盖物的位置信息， 设置西南与东北坐标范围
GroundOverlayOptions <b>transparency</b> (float transparency)	设置 Ground 覆盖物透明度
GroundOverlayOptions <b>visible</b> (boolean visible)	设置 Ground 覆盖物的可见性
GroundOverlayOptions <b>zIndex</b> (int zIndex)	设置 Ground 覆盖物的 zIndex
GroundOverlayOptions <b>extraInfo</b> (Bundle extraInfo)	设置 Ground 覆盖物的额外信息



## 地形图图层覆盖物

### ➤ 示例代码

```
// 定义Ground显示的图片
BitmapDescriptor bdGround = BitmapDescriptorFactory
    .fromResource(R.drawable.box);
// 定义Ground覆盖物选项
OverlayOptions ooGround = new GroundOverlayOptions()
    .positionFromBounds(bounds)
    .image(bdGround)
    .transparency(0.8f);

// 在地图中添加Ground覆盖物
mBaiduMap.addOverlay(ooGround);
```



## 覆盖物介绍



### 标注覆盖物



### 几何图形覆盖物



### 文字覆盖物



### 地形图图层覆盖物



### 弹出窗覆盖物





# 弹出窗覆盖物

- ❖ 在百度地图上可以添加一种可以弹出的覆盖物(**InfoWindow**)，弹出的窗口布局可以自定义，开发者可利用此接口，构建具有更强交互性的地图页面。





## 弹出窗覆盖物

### ❖ 构造方法一：

```
InfoWindow(View view, LatLng position, int yOffset)
```

`view` - InfoWindow 展示的 view  
`position` - InfoWindow 显示的地理位置  
`yOffset` - InfoWindow Y 轴偏移量

通过传入的 view 构造一个 InfoWindow, 此时只是利用该view生成一个Bitmap绘制在地图中, **只能做显示用, 没有响应事件。**





## 弹出窗覆盖物

### ❖ 构造方法二：

```
InfoWindow(BitmapDescriptor bd,  
            LatLng position,  
            int yOffset,  
            InfoWindow.OnInfoWindowClickListener listener)
```

bd - InfoWindow 展示的bitmap  
position - InfoWindow 显示的地理位置  
yOffset - InfoWindow Y 轴偏移量  
listener - InfoWindow 点击监听者

通过此构造方法，可以添加点击事件监听。



## 弹出窗覆盖物

### ➤ 点击事件监听接口

```
InfoWindow.OnInfoWindowClickListener infoWindowClickListener  
    = new InfoWindow.OnInfoWindowClickListener() {  
    public void onInfoWindowClick() {  
        // 点击处理  
    }  
};
```



## 弹出窗覆盖物

### ➤ 示例代码 - Part 1

```
// 创建infowindow展示的view  
Button btn = new Button(getApplicationContext());  
btn.setBackgroundResource(R.drawable.popup);  
btn.setText("弹出窗覆盖物");  
btn.setTextColor(Color.BLACK);  
BitmapDescriptor bitmapDescriptor = BitmapDescriptorFactory  
    .fromView(btn);
```



## 弹出窗覆盖物

### ➤ 示例代码 - Part 2

```
// InfoWindow点击事件
InfoWindow.OnInfoWindowClickListener
infoWindowClickListener
    = new InfoWindow.OnInfoWindowClickListener() {
    @Override
    public void onInfoWindowClick() {
        // 隐藏InfoWindow
        mBaiduMap.hideInfoWindow();
    }
};
```



## 弹出窗覆盖物

### ➤ 示例代码 - Part 3

```
// 创建Infowindow
LatLng point = new LatLng(39.915, 116.404);
InfoWindow infoWindow
    = new InfoWindow(bitmapDescriptor,
                    point,
                    -47,
                    infoWindowClickListener);

// 显示InfoWindow
mBaiduMap.showInfoWindow(infoWindow);
```



## 弹出窗覆盖物



## 运行效果





**Thank You!**

