



Basemap是Matplotlib包中一个非常有名的拓展工具，可以将Basemap理解为有地图背景的坐标系，因此它也可以快速地绘制矢量地图图形。Basemap支持Matplotlib语法，因此常与Matplotlib一起导入并大量用于科学领域绘图。

```
导入包 (matplotlib 1.0.7)
from mpl_toolkits.basemap import Basemap
import matplotlib.pyplot as plt
```

第一步 创建地图区域

`m = Basemap()`

该函数中包含以下常用可选参数：

- `llcrnrlon` | 地图左下角经度.
- `llcrnrlat` | 地图左下角纬度.
- `urcrnrlon` | 地图右上角经度.
- `urcrnrlat` | 地图右上角纬度.
- `width` | 映射单位下的地图宽度.
- `height` | 映射单位下的地图高度.
- `lon_0` | 中心点经度.
- `lat_0` | 中心点纬度.
- `llcrnrx` | 左下角x的映射单位坐标.
- `llcrnry` | 左下角y的映射单位坐标.
- `urcrnrx` | 右上角x的映射单位坐标.
- `urcrnry` | 右上角y的映射单位坐标.
- `resolution` | 指定边界数据集的分辨率，其中c、l、i、h、f分布表示极低、低、中、高、极高.
- `rsphere` | 定义球体投影的半径，默认为6370997米.
- `area_thresh` | 不绘制面积(km^2)小于指定大小的海岸线和湖泊.
- `projection` | 映射方式，默认为‘cyl’.

如果要查看Basemap支持的所有映射方式，可用类变量字典`supported_projections`；如果要查看每个映射方式所接受的参数，可使用类变量字典`projection_params`。比较常用的投射方式有：cea、aeqd、moll、lcc、eqdc、cyl、ortho、robin等。

第二步 添加地图细节

以下是添加地图细节常用的几个函数：

- `m.drawcoastlines()` | 添加海岸线.
- `m.drawcountries()` | 添加国界.
- `m.drawlsmask()`：对陆地、海洋和湖泊进行颜色填充.
- `m.drawstates()` | 添加美国州界.
- `m.drawrivers()` | 添加主要河流.
- `m.drawparallels()` | 添加纬线.
- `m.drawmeridians()` | 添加经线.
- `m.drawmapboundary()` | 添加地图边界.
- `m.fillcontinents()` | 对内陆填充颜色（此函数有时着色可能会不精确，推荐使用`m.drawlsmask()`函数）.
- `m.bluemarble()`：使用NASA卫星图像作为地图背景.
- `m.etopo()`：使用鲜艳浮雕样式.
- `m.shadedrelief()`：使用暗淡浮雕样式.
- `m.warpimage()`：使用自定义图像作为地图背景，默认为NASA卫星图像作为地图背景.

第三步 使用shapefile文件(可选)

如果需要添加非内置地图细节，可通过加载第三方shapefile（地理空间矢量图）文件的形式来添加更多地图细节。例如可以通过加载全国行政区划的shapefile文件来显示省界。添加shapefile时需指定文件夹而非单个文件。

`m.readshapefile()` | 添加shapefile文件.

第四步 添加数据

数据映射

向地图中添加数据的第一步是要将数据映射到地图的坐标比例中，可使用以下方法：

`lon1, lat1 = m(lon, lat)`

添加数据

- `m.contour()` | 绘制等高线图.
- `m.contourf()` | 带有填充效果的等高线图.
- `m.imshow()` | 在地图上显示自定义图像.
- `m.pcolor()` | 绘制伪彩色图.
- `m.pcolormesh()` | 快速绘制伪彩色图.
- `m.plot()` | 绘制折线图.
- `m.scatter()` | 绘制散点图.
- `m.streamplot()` | 绘制流线图.
- `m.quiver()` | 添加带有箭头的向量(u, v).
- `m.barbs()` | 绘制风杆图.
- `m.drawgreatcircle()` | 绘制大圆圈（用来绘制流向图）.

第五步 其它小工具

`m.nightshade()` | 向地图中添加日夜图，需配合datetime模块使用，例如：

```
from datetime import datetime
t = datetime(2018, 2, 9, 18, 00)
```

`m.nightshade(t, delta=0.15, color='k', alpha=0.6)`

`m.drawmapscale()` | 添加比例尺.

必要参数包括：

- `lon/lat` | 比例尺中心点的坐标.
- `lon0/lat0` | 计算比例尺参考点的坐标.
- `length` | 比例尺在地图上对应的公里数.

可选参数包括：

- `barstyle` | 比例尺的风格，可以是‘simpl-’或‘fancy’.
- `units` | 距离单位，默认为km.
- `fontsize` | 更改字体大小.
- `fontcolor` | 更改字体颜色.
- `yoffset` | 控制比例尺的高度.
- `fillcolor1/fillcolor2` | 比例尺的风格为‘fancy’时设置比例尺的交替颜色.
- `format` | 设置比例尺上的数字格式.

`m.tissot()` | 添加天梭指标球（用来观察地图映射时产生的形变）.

`m.colorbar()` | 向地图添加色彩条.

`plt.annotate()` | 使用plt的方法添加注释.

`plt.txt()` | 使用plt的方法添加文字.

其余plt方法也可使用，例如`plt.title()`、`plt.xlabel()`、`plt.savefig()`、`plt.show()`等.

绘图举例

```
plt.figure(figsize=(10, 10))
m = Basemap(projection='ortho', lat_0=45, lon_0=116)
m.drawcoastlines()
m.shadedrelief()
m.drawcountries(color='red')
m.drawparallels(np.arange(-80, 81, 20))
m.drawmeridians(np.arange(-180, 181, 20))
plt.xlabel("'Orthographic' Projection", family='fantasy', fontsize=30, labelpad=20)
plt.show()
```



'Orthographic' Projection

