

pyecharts可视化

In [1]:

```
import pyecharts.charts

print(dir(pyecharts.charts))
```

```
['BMap', 'Bar', 'Bar3D', 'Boxplot', 'Calendar', 'Candlestick', 'Custom', 'EffectScatter', 'Funnel', 'Gauge', 'Geo', 'Graph', 'GraphGL', 'Grid', 'HeatMap', 'Kline', 'Line', 'Line3D', 'Lines3D', 'Liquid', 'Map', 'Map3D', 'MapGlobe', 'Page', 'Parallel', 'PictorialBar', 'Pie', 'Polar', 'Radar', 'Sankey', 'Scatter', 'Scatter3D', 'Sunburst', 'Surface3D', 'Tab', 'ThemeRiver', 'Timeline', 'Tree', 'TreeMap', 'WordCloud', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__path__', '__spec__', 'base', 'basic_charts', 'chart', 'composite_charts', 'mixins', 'three_axis_charts']
```

pyecharts绘制图表流程

- 从pyecharts.charts模块中导入Bar()函数
- 导入数据
- 添加坐标轴
- 利用bar.render()在指定目录下生成render.html文件
- 打开html文件，得到可视化图表

In [23]:

```
#导入柱形图模块Bar
from pyecharts.charts import Bar

#设置x轴坐标数据
x_data = ["[90~100]", "[80~90)", "[70~80)", "[60~70)", "[0~60)"]
#设置y轴坐标数据
y_data = [1, 14, 14, 5, 8]

#绘制柱形图
bar = Bar()

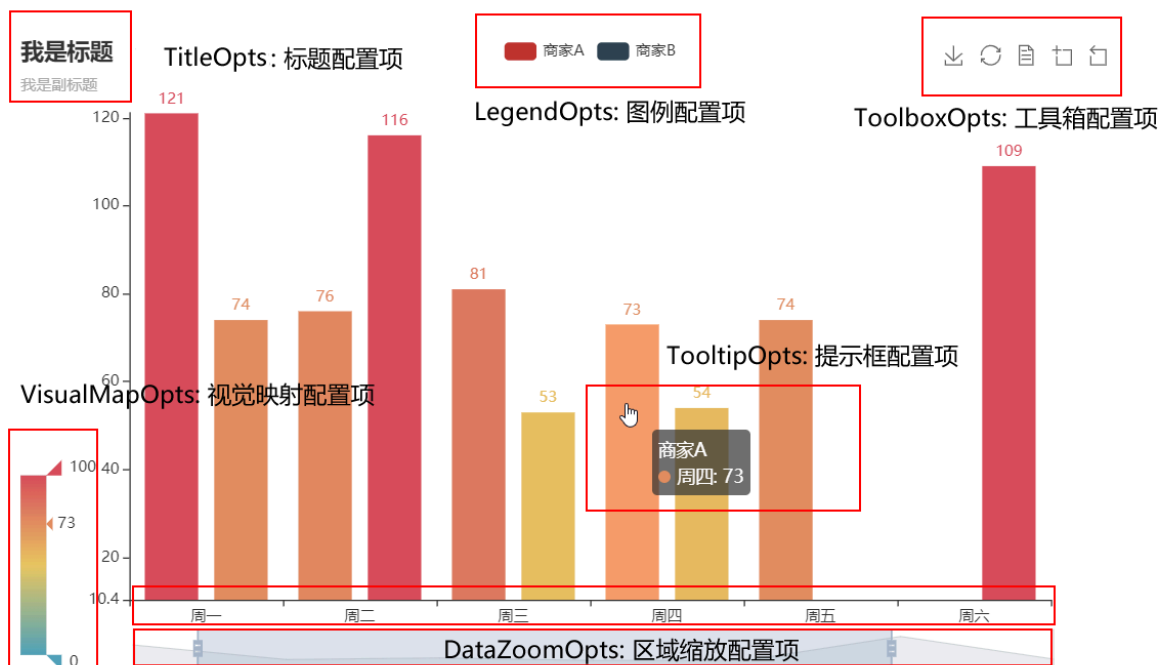
#添加x轴坐标
bar.add_xaxis(x_data)

#添加y轴坐标，第一个元素是类别，第二个是y轴坐标数据列表
bar.add_yaxis("成绩-人数", y_data)

# 利用bar.render()在当前目录下生成render.html文件
# 或传入路径参数在指定目录下生成HTML文件
bar.render("./task6-2.html")
```

Out[23]: 'E:\\\\Ricemoon\\\\Desktop\\\\408\\\\数据可视化\\\\代码\\\\task6-2.html'

图表的全局配置项



1.初始化配置项

参数	描述
width	画布宽度，要求字符串格式，如 width="500px"
height	画布高度，要求字符串格式，如 width="500px"
chart_id	图表ID，作为图表的唯一标识。有多个图表时用来区分不同的图表
page_title	网页标题，字符串格式
theme	图表主题。由ThemeType模块提供
bg_color	图表背景颜色，字符串格式

In [10]:

```
#导入柱形图模块Bar
from pyecharts.charts import Bar
# 设置配置项
from pyecharts import options as opts

#设置x轴坐标数据
x_data = ["[90~100]", "[80~90)", "[70~80)", "[60~70)", "[0~60)"]
#设置y轴坐标数据
y_data = [1, 14, 14, 5, 8]

#绘制柱形图
bar = Bar()
#设置初始化参数 要在画图前设置
bar = Bar(opts.InitOpts(width="1000px",height="800px"))

#添加x轴坐标
bar.add_xaxis(x_data)

#添加y轴坐标，第一个元素是类别，第二个是y轴坐标数据列表
bar.add_yaxis("成绩-人数",y_data)

# 利用bar.render()在当前目录下生成render.html文件
```

```
# 或传入路径参数在指定目录下生成HTML文件
bar.render("../task6-3.html")
```

Out[10]: 'E:\\Ricemoon\\Desktop\\408\\数据可视化\\代码\\task6-3.html'

2.标题配置项

- (1) 通过from pyecharts import options as opts导入options 模块;
- (2) 通过set_global_opts()设置opts.TitleOpts()中的参数。

```
In [34]: from pyecharts import options as opts
from pyecharts.charts import Bar

#设置x轴坐标数据
x_data = ["[90~100]", "[80~90)", "[70~80)", "[60~70)", "[0~60)"]

#设置y轴坐标数据
y_data = [1, 14, 14, 5, 8]

#绘制柱形图
bar = (
    Bar()
    .add_xaxis(x_data)
    .add_yaxis("成绩-人数", y_data)
    .set_global_opts(
        #标题配置项
        title_opts=opts.TitleOpts(
            title="成绩分布状况", pos_left='20%',
            subtitle='网络1班学生成绩', item_gap=20)
    )
)
bar.render("../task6-4.html")
```

Out[34]: 'E:\\Ricemoon\\Desktop\\408\\数据可视化\\代码\\task6-4.html'

3.图例配置项

- (1) 通过from pyecharts import options as opts导入options 模块;
- (2) 通过set_global_opts()方法设置opts.LegendOpts()中的参数

```
In [37]: from pyecharts import options as opts
from pyecharts.charts import Bar
#设置x轴坐标数据
x_data = ["[90~100]", "[80~90)", "[70~80)", "[60~70)", "[0~60)"]
#设置y轴坐标数据
y_data = [1, 14, 14, 5, 8]
#绘制柱形图
bar = (
    Bar()
    .add_xaxis(x_data)
    .add_yaxis("成绩-人数", y_data)
    .set_global_opts(
        #图例配置项
        legend_opts=
            opts.LegendOpts(pos_right=120,
                            orient='horizontal')
    )
)
bar.render("../task6-5.html")
```

```
Out[37]: 'E:\\Ricemoon\\Desktop\\408\\数据可视化\\代码\\task6-5.html'
```

4.工具箱配置项

- (1) 通过from pyecharts import options as opts导入options 模块;
- (2) 通过set_global_opts()方法设置opts.ToolboxOpts()中的参数。

```
In [38]: from pyecharts import options as opts
from pyecharts.charts import Bar
#设置x轴坐标数据
x_data = ["[90~100]", "[80~90)", "[70~80)", "[60~70)", "[0~60)"]
#设置y轴坐标数据
y_data = [1, 14, 14, 5, 8]
#绘制柱形图
bar = (
    Bar()
    .add_xaxis(x_data)
    .add_yaxis("成绩-人数", y_data)
    .set_global_opts(
        #工具箱配置
        toolbox_opts=opts.ToolboxOpts()
    )
)
bar.render("./task6-6.html")
```

```
Out[38]: 'E:\\Ricemoon\\Desktop\\408\\数据可视化\\代码\\task6-6.html'
```

5.视觉映射配置项

- (1) 通过from pyecharts import options as opts导入options 模块;
- (2) 通过set_global_opts()方法设置opts.VisualMapOpts()中的参数。

```
In [45]: from pyecharts import options as opts
from pyecharts.charts import Bar
#设置x轴坐标数据
x_data = ["[90~100]", "[80~90)", "[70~80)", "[60~70)", "[0~60)"]
#设置y轴坐标数据
y_data = [1, 14, 14, 5, 8]
#绘制柱形图
bar = (
    Bar()
    .add_xaxis(x_data)
    .add_yaxis("成绩-人数", y_data)
    .set_global_opts(
        #视觉映射配置项
        visualmap_opts=opts.VisualMapOpts(

            visualmap_opts=opts.VisualMapOpts(
                is_show=True,
                # 默认为color, size
                type_='color',
                min_=0, # 默认最小值为0
                max_=30, # 默认最大值为100
                range_opacity=0.7, # 图元（比如柱形图）与文字的透明度
                range_text=['max', 'min'], # 两端的文本
                range_color=['blue', 'green', 'red'], # 自定义过渡颜色

                # 对齐方式 : vertical(默认)、horizontal
```

```

        orient='vertical',
        # 图例位置: pos_left、pos_right、pos_bottom、pos_top
        is_pieewise=False, # 是否分段
        is_inverse=False, # 是否翻转(比如递增-递减)
    )

)

bar.render("./task6-7.html")

```

Out[45]: 'E:\\Ricemoon\\Desktop\\408\\数据可视化\\代码\\task6-7.html'

6.提示框配置项

- (1) 通过from pyecharts import options as opts导入options 模块;
- (2) 通过set_global_opts()方法设置opts.TooltipOpts()中的参数.

```

In [49]: from pyecharts import options as opts
from pyecharts.charts import Bar
#设置x轴坐标数据
x_data = ["[90~100]", "[80~90]", "[70~80]", "[60~70]", "[0~60]"]
#设置y轴坐标数据
y_data = [1, 14, 14, 5, 8]
#绘制柱形图
bar = (
    Bar()
    .add_xaxis(x_data)
    .add_yaxis("成绩-人数", y_data)
    .set_global_opts(
        #提示框配置项
        tooltip_opts=opts.TooltipOpts() # 显示提示框组件

        tooltip_opts=opts.TooltipOpts(
            is_show=True, # 默认为True

            # 触发类型
            # item: 数据项, 一般用于散点图、柱形图、饼图
            # axis: 坐标轴, 提示线, 主要用于条形图、折线图
            trigger='item',
            # 触发条件: mousemove,click,mousemove|click
            trigger_on='mousemove|click', # 默认移动或点击

            is_show_content=True, # 是否提示款浮层(内容)
            # 标签内容的格式
            # 字符串中模板变量:
            # {a}:系列名series_name
            # {b}:数据名
            # {c}:值
            formatter='{a}:{b}-{c}',
            background_color='green', # 提示框的背景颜色
            border_width=0.8, # 边框的宽度
        )

    )

bar.render("./task6-8.html")

```

Out[49]: 'E:\\Ricemoon\\Desktop\\408\\数据可视化\\代码\\task6-8.html'

7.区域缩放配置项


```
)  
bar.render("./task6-10.html")
```

Out[54]: 'E:\\Ricemoon\\Desktop\\408\\数据可视化\\代码\\task6-10.html'

9.原生图形元素组件的设置方法

- (1) 通过from pyecharts import options as opts导入options 模块;
- (2) 通过set_global_opts()方法设置opts.GraphicItem()中的参数。

```
In [62]: from pyecharts import options as opts  
from pyecharts.charts import Bar  
#设置x轴坐标数据  
x_data = ["[90~100]", "[80~90]", "[70~80]", "[60~70]", "[0~60)"]  
#设置y轴坐标数据  
y_data = [1, 14, 14, 5, 8]  
#绘制柱形图  
bar = (  
    Bar()  
    .add_xaxis(x_data)  
    .add_yaxis("成绩-人数", y_data)  
    .set_global_opts(  
  
        # 添加脚注  
        graphic_opts = opts.GraphicGroup(  
            graphic_item = opts.GraphicItem(left='2%', bottom='10%'),  
            children = [  
                opts.GraphicText(graphic_textstyle_opts =  
                    opts.GraphicTextStyleOpts(  
                        text='数据来源：学生成绩表',  
                        font="36px SimHei")  
                )  
            ]  
        )  
    )  
)  
bar.render("./task6-11.html")
```

Out[62]: 'E:\\Ricemoon\\Desktop\\408\\数据可视化\\代码\\task6-11.html'

10.文字样式配置项

- (1) 通过from pyecharts import options as opts导入options 模块;
- (2) 通过set_global_opts()方法设置opts.TextStyleOpts()中的参数。

```
In [63]: from pyecharts import options as opts  
from pyecharts.charts import Bar  
#设置x轴坐标数据  
x_data = ["[90~100]", "[80~90]", "[70~80]", "[60~70)", "[0~60)"]  
#设置y轴坐标数据  
y_data = [1, 14, 14, 5, 8]  
#绘制柱形图  
bar = (  
    Bar()  
    .add_xaxis(x_data)  
    .add_yaxis("成绩-人数", y_data)  
    .set_global_opts(  
        #提示框配置项  
        tooltip_opts=opts.TooltipOpts(#文字样式配置项  
            textstyle_opts=
```

```

        opts.TextStyleOpts(
            font_size=20,
            font_weight='bold',
            font_style='italic')
    )
)
bar.render("./task6-12.html")

```

Out[63]: 'E:\\Ricemoon\\Desktop\\408\\数据可视化\\代码\\task6-12.html'

11. 标签配置项

- (1) 通过from pyecharts import options as opts导入options 模块;
- (2) 通过set_series_opts()方法或者.add_yaxis()方法设置opts.LabelOpts()中的参数。

```

In [64]: from pyecharts import options as opts
from pyecharts.charts import Bar
#设置x轴坐标数据
x_data = ["[90~100]", "[80~90)", "[70~80)", "[60~70)", "[0~60)"]
#设置y轴坐标数据
y_data = [1, 14, 14, 5, 8]
#绘制柱形图
bar = (
    Bar()
    .add_xaxis(x_data)
    .add_yaxis("成绩-人数", y_data)
    .set_global_opts(
        #提示框配置项
        tooltip_opts=opts.TooltipOpts(), # 显示提示框组件
    )
    #系列配置项
    .set_series_opts(
        # 标签配置项
        label_opts=opts.LabelOpts(is_show=True,
                                   font_size=20,
                                   font_weight='bold',
                                   font_style='italic')
    )
)
bar.render("./task6-13.html")

```

Out[64]: 'E:\\Ricemoon\\Desktop\\408\\数据可视化\\代码\\task6-13.html'

12. 线样式配置项

- (1) 通过from pyecharts import options as opts导入options 模块;
- (2) 通过set_series_opts()方法或者.add_yaxis()中设置opts.LineStyleOpts()中的参数。

```

In [66]: from pyecharts import options as opts
from pyecharts.charts import Line
#设置x轴坐标数据
x_data = ["[90~100]", "[80~90)", "[70~80)", "[60~70)", "[0~60)"]
#设置y轴坐标数据
y_data = [1, 14, 14, 5, 8]
#绘制折线图
line = (
    Line()
    .add_xaxis(x_data)

```



```

        .add_yaxis("成绩-人数", y_data)
        #系列配置项
        .set_series_opts(
            #线样式配置项
            linestyle_opts=opts.LineStyleOpts(is_show=True,width=4,
                                                type_='dotted',opacity=0.5)
        )
    )
    line.render("./task6-14.html")

```

Out[66]: 'E:\\Ricemoon\\Desktop\\408\\数据可视化\\代码\\task6-14.html'

13.分割线配置项

- (1) 通过from pyecharts import options as opts导入options 模块;
- (2) 通过坐标轴配置项中的splitline_opts=opts.SplitLineOpts()设置分割线配置项中的参数。

```

In [67]: from pyecharts import options as opts
        from pyecharts.charts import Bar
        #设置x轴坐标数据
        x_data = ["[90~100]", "[80~90)", "[70~80)", "[60~70)", "[0~60)"]
        #设置y轴坐标数据
        y_data = [1, 14, 14, 5, 8]
        #绘制柱形图
        bar = (
            Bar()
            .add_xaxis(x_data)
            .add_yaxis("成绩-人数", y_data)
            .set_global_opts(#提示框配置项
                tooltip_opts=opts.TooltipOpts(), # 显示提示框组件
                # x坐标轴配置项
                xaxis_opts=opts.AxisOpts(name='成绩',
                                         name_location='middle',
                                         name_gap=30,
                                         name_rotate=45,
                                         #分割线配置项
                                         splitline_opts=opts.SplitLineOpts(
                                             is_show=True,
                                             linestyle_opts=
                                                opts.LineStyleOpts(width=4))
                ),
                # y坐标轴配置项
                yaxis_opts=opts.AxisOpts(name='人数', name_gap=30, name_rotate=45)
            )
        )
        bar.render("./task6-15.html")

```

Out[67]: 'E:\\Ricemoon\\Desktop\\408\\数据可视化\\代码\\task6-15.html'