

图表 API

`Base` 类是所有图表的基类，包括组合图表，`Base` 类 API 如下

func pyecharts.Base.add_js_funcs

```
# 新增 js 代码，js 代码会被渲染进 HTML 中执行
def add_js_funcs(*fns):
```

func pyecharts.Base.set_colors

```
# 设置全局 Label 颜色
def set_colors(colors: Sequence[str]):
```

func pyecharts.Base.get_options

```
# 获取全局 options
def get_options() -> dict:
```

func pyecharts.Base.dump_options

```
# 获取全局 options, JSON 格式 (JsCode 生成的函数不带引号)
```

```
def dump_options() -> str:
```

```
# 获取全局 options, JSON 格式 (JsCode 生成的函数带引号, 在前后端分离传输数据时使用)
```

```
def dump_options_with_quotes() -> str:
```

func pyecharts.Base.render

```
# 渲染图表到 HTML 文件
```

```
def render(
```

```
    # 生成图片路径
```

```
    path: str = "render.html",
```

```
    # 模板路径
```

```
    template_name: str = "simple_chart.html",
```

```
    # jinja2.Environment 类实例, 可以配置各类环境参数
```

```
    env: Optional[Environment] = None,
```

```
) -> str
```

func pyecharts.Base.render_notebook

```
# 将图形渲染到 notebook  
def render_notebook()
```

func pyecharts.Base.load_javascript

```
# 加载 js 资源，在 notebook 环境为 JupyterLab 时需要用到，仅在第一次渲染图前使用加载即可。  
def load_javascript()
```