```
from  pyecharts  import  options  as  opts
from  pyecharts.charts  import  Map
from  pyecharts.faker  import  Faker

c  =  (
    Map()
    .add("商家A",  [list(z)  for  z  in  zip(Faker.provinces,  Faker.values())],  "china")
    .set_global_opts(title_opts=opts.TitleOpts(title="Map-基本示例"))
    .render_notebook()
)

c
```

```
from  pyecharts  import  options  as  opts
from  pyecharts.charts  import  Map
from  pyecharts.faker  import  Faker

c  =  (
    Map()
    .add("商家A",  [list(z)  for  z  in  zip(Faker.provinces,  Faker.values())],  "china")
    .set_series_opts(label_opts=opts.LabelOpts(is_show=False))
    .set_global_opts(title_opts=opts.TitleOpts(title="Map-不显示Label"))
    .render_notebook()
)

c
```

```
from  pyecharts  import  options  as  opts
from  pyecharts.charts  import  Map
from  pyecharts.faker  import  Faker

c  =  (
        Map()
        .add("商家A",  [list(z)  for  z  in  zip(Faker.guangdong_city,  Faker.values())],  "广东")  #  [list(z)  for  z  in  zip(['沈阳市'],  Faker.v
        .set_global_opts(
                title_opts=opts.TitleOpts(title="Map-广东地图"),  visualmap_opts=opts.VisualMapOpts()
        )
                .render_notebook()
)


c
```

开始借助 AI 编写或生成代码。

```
from  pyecharts  import  options  as  opts
from  pyecharts.charts  import  Map
from  pyecharts.faker  import  Faker

c  =  (
        Map()
        .add("商家A",  [list(z)  for  z  in  zip(Faker.country,  Faker.values())],  "world")
        .set_series_opts(label_opts=opts.LabelOpts(is_show=False))
        .set_global_opts(
                title_opts=opts.TitleOpts(title="Map-世界地图"),
                visualmap_opts=opts.VisualMapOpts(max_=200),
        )
        .render_notebook()
)


c
```

```
#    组合图表
#  并行多图  Grid()
from  pyecharts  import  options  as  opts
from  pyecharts.charts  import  Bar,  Grid,  Line
from  pyecharts.faker  import  Faker

bar  =  (
        Bar()
        .add_xaxis(Faker.choose())
        .add_yaxis("商家A",  Faker.values())
        .add_yaxis("商家B",  Faker.values())
        .set_global_opts(title_opts=opts.TitleOpts(title="Grid-Bar"))
)
line  =  (
        Line()
        .add_xaxis(Faker.choose())
        .add_yaxis("商家A",  Faker.values())
        .add_yaxis("商家B",  Faker.values())
        .set_global_opts(
                title_opts=opts.TitleOpts(title="Grid-Line",  pos_top="48%"),
                legend_opts=opts.LegendOpts(pos_top="48%"),
        )
)

grid  =  (
        Grid()
        .add(bar,  grid_opts=opts.GridOpts(pos_bottom="60%"))    #  相对于底部高  60  %
        .add(line,  grid_opts=opts.GridOpts(pos_top="60%"))
        .render_notebook()
)

grid
```

```
from pyecharts import options as opts
from pyecharts.charts import Grid, Line, Scatter
from pyecharts.faker import Faker

scatter = (
        Scatter()
        .add_xaxis(Faker.choose())
        .add_yaxis("商家A", Faker.values())
        .add_yaxis("商家B", Faker.values())
        .set_global_opts(
                title_opts=opts.TitleOpts(title="Grid-Scatter"),
                legend_opts=opts.LegendOpts(pos_left="20%"),
        )
)
line = (
        Line()
        .add_xaxis(Faker.choose())
        .add_yaxis("商家A", Faker.values())
        .add_yaxis("商家B", Faker.values())
        .set_global_opts(
                title_opts=opts.TitleOpts(title="Grid-Line", pos_right="5%"),
                legend_opts=opts.LegendOpts(pos_right="20%"),
        )
)

grid = (
        Grid()
        .add(scatter, grid_opts=opts.GridOpts(pos_left="55%"))
        .add(line, grid_opts=opts.GridOpts(pos_right="55%"))
        .render_notebook()
)

grid
```

```
#  选项卡多图

from  pyecharts  import  options  as  opts
from  pyecharts.charts  import  Bar,  Grid,  Line,  Pie,  Tab
from  pyecharts.faker  import  Faker


def  bar_datazoom_slider():
        c  =  (
                Bar()
                .add_xaxis(Faker.days_attrs)
                .add_yaxis("商家A",  Faker.days_values)
                .set_global_opts(
                        title_opts=opts.TitleOpts(title="Bar-DataZoom（slider-水平）"),
                )
        )
        return  c


def  line_markpoint()  :
        c  =  (
                Line()
                .add_xaxis(Faker.choose())
                .add_yaxis(
                        "商家A",
                        Faker.values(),
                )
                .set_global_opts(title_opts=opts.TitleOpts(title="Line-MarkPoint"))
        )
        return  c


def  pie_rosetype():
        v  =  Faker.choose()
        c  =  (
                Pie()
                .add(
                        "",
                        [list(z)  for  z  in  zip(v,  Faker.values())],
                        radius=["30%",  "75%"],
                        center=["25%",  "50%"],
                        rosetype="radius",
                        label_opts=opts.LabelOpts(is_show=False),
                )
                .set_global_opts(title_opts=opts.TitleOpts(title="Pie-玫瑰图示例"))
        )
        return  c

tab  =  Tab()
tab.add(bar_datazoom_slider(),  "bar-example")
tab.add(line_markpoint(),  "line-example")
tab.add(pie_rosetype(),  "pie-example")
tab.render_notebook()
```