

系列配置项

ItemStyleOpts：图元样式配置项

class pyecharts.options.ItemStyleOpts

```
class ItemStyleOpts(  
    # 图形的颜色。  
    # 颜色可以使用 RGB 表示，比如 'rgb(128, 128, 128)'，如果想要加上 alpha 通道表示不透明度，  
    # 可以使用 RGBA，比如 'rgba(128, 128, 128, 0.5)'，也可以使用十六进制格式，比如 '#ccc'。  
    # 除了纯色之外颜色也支持渐变色和纹理填充  
    #  
    # 线性渐变，前四个参数分别是 x0, y0, x2, y2，范围从 0 - 1，相当于在图形包围盒中的百分比，  
    # 如果 globalCoord 为 `true`，则该四个值是绝对的像素位置  
    # color: {  
    #     type: 'linear',  
    #     x: 0,  
    #     y: 0,  
    #     x2: 0,  
    #     y2: 1,  
    #     colorStops: [{  
    #         offset: 0, color: 'red' // 0% 处的颜色  
    #     }, {  
    #         offset: 1, color: 'blue' // 100% 处的颜色  
    #     }],  
    #     global: false // 缺省为 false  
    # }
```

```
#
# 径向渐变，前三个参数分别是圆心 x, y 和半径，取值同线性渐变
# color: {
#     type: 'radial',
#     x: 0.5,
#     y: 0.5,
#     r: 0.5,
#     colorStops: [{
#         offset: 0, color: 'red' // 0% 处的颜色
#     }, {
#         offset: 1, color: 'blue' // 100% 处的颜色
#     }],
#     global: false // 缺省为 false
# }
#
# 纹理填充
# color: {
#     image: imageDom, // 支持为 HTMLImageElement, HTMLCanvasElement, 不支持路径字符串
#     repeat: 'repeat' // 是否平铺, 可以是 'repeat-x', 'repeat-y', 'no-repeat'
# }
color: Optional[str] = None,

# 阴线 图形的颜色。
color0: Optional[str] = None,

# 图形的描边颜色。支持的颜色格式同 color, 不支持回调函数。
border_color: Optional[str] = None,

# 阴线 图形的描边颜色。
```

```
border_color0: Optional[str] = None,

# 描边宽度，默认不描边。
border_width: Optional[Numeric] = None,

# 支持 'dashed', 'dotted'。
border_type: Optional[str] = None,

# 图形透明度。支持从 0 到 1 的数字，为 0 时不绘制该图形。
opacity: Optional[Numeric] = None,

# 区域的颜色。
area_color: Optional[str] = None,
)
```

TextStyleOpts：文字样式配置项

class pyecharts.options.TextStyleOpts

```
class TextStyleOpts(
    # 文字颜色。
    color: Optional[str] = None,

    # 文字字体的风格
    # 可选: 'normal', 'italic', 'oblique'
    font_style: Optional[str] = None,
```

主标题文字字体的粗细，可选：

'normal', 'bold', 'bolder', 'lighter'

font_weight: Optional[str] = None,

文字的字体系列

还可以是 'serif' , 'monospace', 'Arial', 'Courier New', 'Microsoft YaHei', ...

font_family: Optional[str] = None,

文字的字体系大小

font_size: Optional[Numeric] = None,

文字水平对齐方式，默认自动

align: Optional[str] = None,

文字垂直对齐方式，默认自动

vertical_align: Optional[str] = None,

行高

line_height: Optional[str] = None,

文字块背景色。可以是直接的颜色值，例如：'#123234', 'red', 'rgba(0,23,11,0.3)'

background_color: Optional[str] = None,

文字块边框颜色

border_color: Optional[str] = None,

文字块边框宽度

border_width: Optional[Numeric] = None,

```
# 文字块的圆角
border_radius: Union[Numeric, Sequence, None] = None,

# 文字块的内边距
# 例如 padding: [3, 4, 5, 6]: 表示 [上, 右, 下, 左] 的边距
# 例如 padding: 4: 表示 padding: [4, 4, 4, 4]
# 例如 padding: [3, 4]: 表示 padding: [3, 4, 3, 4]
padding: Union[Numeric, Sequence, None] = None,

# 文字块的背景阴影颜色
shadow_color: Optional[str] = None,

# 文字块的背景阴影长度
shadow_blur: Optional[Numeric] = None,

# 文字块的宽度
width: Optional[str] = None,

# 文字块的高度
height: Optional[str] = None,

# 在 rich 里面，可以自定义富文本样式。利用富文本样式，可以在标签中做出非常丰富的效果
# 具体配置可以参考一下 https://www.echartsjs.com/tutorial.html#%E5%AF%8C%E6%96%87%E6%9C%AC%E6%A0%87%E7%AD%BE
rich: Optional[dict] = None,

)
```

LabelOpts: 标签配置项

class pyecharts.options.LabelOpts

```
class LabelOpts(  
    # 是否显示标签。  
    is_show: bool = True,  
  
    # 标签的位置。可选  
    # 'top', 'left', 'right', 'bottom', 'inside', 'insideLeft', 'insideRight'  
    # 'insideTop', 'insideBottom', 'insideTopLeft', 'insideBottomLeft'  
    # 'insideTopRight', 'insideBottomRight'  
    position: Union[str, Sequence] = "top",  
  
    # 文字的颜色。  
    # 如果设置为 'auto', 则为视觉映射得到的颜色, 如系列色。  
    color: Optional[str] = None,  
  
    # 距离图形元素的距离。当 position 为字符描述值 (如 'top'、'insideRight') 时候有效。  
    distance: Union[Numeric, Sequence, None] = None,  
  
    # 文字的字体大小  
    font_size: Numeric = 12,  
  
    # 文字字体的风格, 可选:  
    # 'normal', 'italic', 'oblique'  
    font_style: Optional[str] = None,  
  
    # 文字字体的粗细, 可选:  
    # 'normal', 'bold', 'bolder', 'lighter'
```

```
font_weight: Optional[str] = None,
```

```
# 文字的字体系列
```

```
# 还可以是 'serif' , 'monospace', 'Arial', 'Courier New', 'Microsoft YaHei', ...
```

```
font_family: Optional[str] = None,
```

```
# 标签旋转。从 -90 度到 90 度。正值是逆时针。
```

```
rotate: Optional[Numeric] = None,
```

```
# 刻度标签与轴线之间的距离。
```

```
margin: Optional[Numeric] = 8,
```

```
# 坐标轴刻度标签的显示间隔，在类目轴中有效。
```

```
# 默认会采用标签不重叠的策略间隔显示标签。
```

```
# 可以设置成 0 强制显示所有标签。
```

```
# 如果设置为 1，表示『隔一个标签显示一个标签』，如果值为 2，表示隔两个标签显示一个标签，以此类推。
```

```
# 可以用数值表示间隔的数据，也可以通过回调函数控制。回调函数格式如下：
```

```
# (index:number, value: string) => boolean
```

```
# 第一个参数是类目的 index，第二个值是类目名称，如果跳过则返回 false。
```

```
interval: Union[Numeric, str, None]= None,
```

```
# 文字水平对齐方式，默认自动。可选：
```

```
# 'left', 'center', 'right'
```

```
horizontal_align: Optional[str] = None,
```

```
# 文字垂直对齐方式，默认自动。可选：
```

```
# 'top', 'middle', 'bottom'
```

```
vertical_align: Optional[str] = None,
```

标签内容格式器，支持字符串模板和回调函数两种形式，字符串模板与回调函数返回的字符串均支持用 \n 换行。

模板变量有 {a}, {b}, {c}, {d}, {e}，分别表示系列名，数据名，数据值等。

在 trigger 为 'axis' 的时候，会有多个系列的数据，此时可以通过 {a0}, {a1}, {a2} 这种后面加索引的方式表示系列的索引。

不同图表类型下的 {a}, {b}, {c}, {d} 含义不一样。其中变量{a}, {b}, {c}, {d}在不同图表类型下代表数据含义为：

折线（区域）图、柱状（条形）图、K线图：{a}（系列名称），{b}（类目值），{c}（数值），{d}（无）

散点图（气泡）图：{a}（系列名称），{b}（数据名称），{c}（数值数组），{d}（无）

地图：{a}（系列名称），{b}（区域名称），{c}（合并数值），{d}（无）

饼图、仪表盘、漏斗图：{a}（系列名称），{b}（数据项名称），{c}（数值），{d}（百分比）

示例：formatter: '{b}: {@score}'

#

回调函数，回调函数格式：

(params: Object|Array) => string

参数 params 是 formatter 需要的单个数据集。格式如下：

{

componentType: 'series',

// 系列类型

seriesType: string,

// 系列在传入的 option.series 中的 index

seriesIndex: number,

// 系列名称

seriesName: string,

// 数据名，类目名

name: string,

// 数据在传入的 data 数组中的 index

dataIndex: number,

// 传入的原始数据项

data: Object,

// 传入的数据值


```
#     value: number|Array,  
#     // 数据图形的颜色  
#     color: string,  
# }  
formatter: Optional[str] = None,  
  
# 在 rich 里面，可以自定义富文本样式。利用富文本样式，可以在标签中做出非常丰富的效果  
# 具体配置可以参考一下 https://www.echartsjs.com/tutorial.html#%E5%AF%8C%E6%96%87%E6%9C%AC%E6%A0%87%E7%AD%BE  
rich: Optional[dict] = None,  
  
)
```

LineStyleOpts：线样式配置项

class pyecharts.options.LineStyleOpts

```
class LineStyleOpts(  
    # 是否显示  
    is_show: bool = True,  
  
    # 线宽。  
    width: Numeric = 1,  
  
    # 图形透明度。支持从 0 到 1 的数字，为 0 时不绘制该图形。  
    opacity: Numeric = 1,  
  
    # 线的弯曲度，0 表示完全不弯曲  
    curve: Numeric = 0,
```

```
# 线的类型。可选：
# 'solid', 'dashed', 'dotted'
type_: str = "solid",

# 线的颜色。
# 颜色可以使用 RGB 表示，比如 'rgb(128, 128, 128)'，如果想要加上 alpha 通道表示不透明度，
# 可以使用 RGBA，比如 'rgba(128, 128, 128, 0.5)'，也可以使用十六进制格式，比如 '#ccc'。
# 除了纯色之外颜色也支持渐变色和纹理填充
#
# 线性渐变，前四个参数分别是 x0, y0, x2, y2，范围从 0 - 1，相当于在图形包围盒中的百分比，
# 如果 globalCoord 为 `true`，则该四个值是绝对的像素位置
# color: {
#     type: 'linear',
#     x: 0,
#     y: 0,
#     x2: 0,
#     y2: 1,
#     colorStops: [{
#         offset: 0, color: 'red' // 0% 处的颜色
#     }, {
#         offset: 1, color: 'blue' // 100% 处的颜色
#     }],
#     global: false // 缺省为 false
# }
#
# 径向渐变，前三个参数分别是圆心 x, y 和半径，取值同线性渐变
# color: {
#     type: 'radial',
```

```

#     x: 0.5,
#     y: 0.5,
#     r: 0.5,
#     colorStops: [{
#         offset: 0, color: 'red' // 0% 处的颜色
#     }, {
#         offset: 1, color: 'blue' // 100% 处的颜色
#     }],
#     global: false // 缺省为 false
# }
#
# 纹理填充
# color: {
#     image: imageDom, // 支持为 HTMLImageElement, HTMLCanvasElement, 不支持路径字符串
#     repeat: 'repeat' // 是否平铺, 可以是 'repeat-x', 'repeat-y', 'no-repeat'
# }
color: Union[str, Sequence, None] = None,
)

```

Lines3DEffectOpts: 3D线样式配置项

class pyecharts.options.Lines3DEffectOpts

```

class Line3DEffectOpts(
    # 是否显示尾迹特效, 默认不显示。
    is_show: bool = True,

```

```
# 尾迹特效的周期。
period: Numeric = 4,

# 轨迹特效的移动动画是否是固定速度，单位按三维空间的尺寸
# 设置为非 null 的值后会忽略 period 配置项。
constant_speed: Optional[Numeric] = None,

# 尾迹的宽度。
trail_width: Numeric = 4,

# 尾迹的长度，范围从 0 到 1，为线条长度的百分比。
trail_length: Numeric = 0.1,

# 尾迹的颜色，默认跟线条颜色相同。
trail_color: Optional[str] = None,

# 尾迹的不透明度，默认跟线条不透明度相同。
trail_opacity: Optional[Numeric] = None,
)
```

SplitLineOpts：分割线配置项

class pyecharts.options.SplitLineOpts

```
class SplitLineOpts(
    # 是否显示分割线
    is_show: bool = False,
```

```
# 线风格配置项，参考 `series_options.SplitLineOpts`  
linestyle_opts: LineStyleOpts = LineStyleOpts()  
)
```

MarkPointItem: 标记点数据项

class pyecharts.options.MarkPointItem

```
class MarkPointItem(  
    # 标注名称。  
    name: Optional[str] = None,  
  
    # 特殊的标注类型，用于标注最大值最小值等。可选：  
    # 'min' 最大值。  
    # 'max' 最大值。  
    # 'average' 平均值。  
    type_: Optional[str] = None,  
  
    # 在使用 type 时有效，用于指定在哪个维度上指定最大值最小值，可以是  
    # 0 (xAxis, radiusAxis),  
    # 1 (yAxis, angleAxis)，默认使用第一个数值轴所在的维度。  
    value_index: Optional[Numeric] = None,  
  
    # 在使用 type 时有效，用于指定在哪个维度上指定最大值最小值。这可以是维度的直接名称，  
    # 例如折线图时可以是 x、angle 等、candlestick 图时可以是 open、close 等维度名称。  
    value_dim: Optional[str] = None,
```

```
# 标注的坐标。坐标格式视系列的坐标系而定，可以是直角坐标系上的 x, y,
# 也可以是极坐标系上的 radius, angle。例如 [121, 2323]、['aa', 998]。
coord: Optional[Sequence] = None,

# 相对容器的屏幕 x 坐标，单位像素。
x: Optional[Numeric] = None,

# 相对容器的屏幕 y 坐标，单位像素。
y: Optional[Numeric] = None,

# 标注值，可以不设。
value: Optional[Numeric] = None,

# 标记的图形。
# ECharts 提供的标记类型包括 'circle', 'rect', 'roundRect', 'triangle',
# 'diamond', 'pin', 'arrow', 'none'
# 可以通过 'image://url' 设置为图片，其中 URL 为图片的链接，或者 dataURI。
symbol: Optional[str] = None,

# 标记的大小，可以设置成诸如 10 这样单一的数字，也可以用数组分开表示宽和高，
# 例如 [20, 10] 表示标记宽为 20，高为 10。
symbol_size: Union[Numeric, Sequence] = None,

# 标记点样式配置项，参考 `series_options.ItemStyleOpts`
itemstyle_opts: Union[ItemStyleOpts, dict, None] = None,

)
```

MarkPointOpts: 标记点配置项

class pyecharts.options.MarkPointOpts

```
class MarkPointOpts(  
    # 标记点数据, 参考 `series_options.MarkPointItem`  
    data: Sequence[Union[MarkPointItem, dict]] = None,  
  
    # 标记的图形。  
    # ECharts 提供的标记类型包括 'circle', 'rect', 'roundRect', 'triangle',  
    # 'diamond', 'pin', 'arrow', 'none'  
    # 可以通过 'image://url' 设置为图片, 其中 URL 为图片的链接, 或者 dataURI。  
    symbol: Optional[str] = None,  
  
    # 标记的大小, 可以设置成诸如 10 这样单一的数字, 也可以用数组分开表示宽和高,  
    # 例如 [20, 10] 表示标记宽为 20, 高为 10。  
    # 如果需要每个数据的图形大小不一样, 可以设置为如下格式的回调函数:  
    # (value: Array|number, params: Object) => number|Array  
    # 其中第一个参数 value 为 data 中的数据值。第二个参数 params 是其它的数据项参数。  
    symbol_size: Union[None, Numeric] = None,  
  
    # 标签配置项, 参考 `series_options.LabelOpts`  
    label_opts: LabelOpts = LabelOpts(position="inside", color="#fff"),  
)
```

MarkLineItem: 标记线数据项

class pyecharts.options.MarkLineItem

```
class MarkLineItem(
    # 标注名称。
    name: Optional[str] = None,

    # 特殊的标注类型，用于标注最大值最小值等。可选：
    # 'min' 最大值。
    # 'max' 最大值。
    # 'average' 平均值。
    type_: Optional[str] = None,

    # 相对容器的屏幕 x 坐标，单位像素。
    x: Union[str, Numeric, None] = None,

    # x 数据坐标。
    xcoord: Union[str, Numeric, None] = None,

    # 相对容器的屏幕 y 坐标，单位像素。
    y: Union[str, Numeric, None] = None,

    # y 数据坐标。
    ycoord: Union[str, Numeric, None] = None,

    # 在使用 type 时有效，用于指定在哪个维度上指定最大值最小值，可以是
    # 0 (xAxis, radiusAxis),
    # 1 (yAxis, angleAxis)，默认使用第一个数值轴所在的维度。
    value_index: Optional[Numeric] = None,
```



```

# 在使用 type 时有效，用于指定在哪个维度上指定最大值最小值。这可以是维度的直接名称，
# 例如折线图时可以是 x、angle 等、candlestick 图时可以是 open、close 等维度名称。
value_dim: Optional[str] = None,

# 起点或终点的坐标。坐标格式视系列的坐标系而定，可以是直角坐标系上的 x, y,
# 也可以是极坐标系上的 radius, angle。
coord: Optional[Sequence] = None,

# 终点标记的图形。
# ECharts 提供的标记类型包括 'circle', 'rect', 'roundRect', 'triangle',
# 'diamond', 'pin', 'arrow', 'none'
# 可以通过 'image://url' 设置为图片，其中 URL 为图片的链接，或者 dataURI。
symbol: Optional[str] = None,

# 标记的大小，可以设置成诸如 10 这样单一的数字，也可以用数组分开表示宽和高，
# 例如 [20, 10] 表示标记宽为 20，高为 10。
symbol_size: Optional[Numeric] = None,
)

```

MarkLineOpts: 标记线配置项

`class pyecharts.options.MarkLineOpts`

```

class MarkLineOpts(
    # 图形是否不响应和触发鼠标事件，默认为 false，即响应和触发鼠标事件。
    is_silent: bool = False,

```

```
# 标记线数据, 参考 `series_options.MarkLineItem`
data: Sequence[Union[MarkLineItem, dict]] = None,

# 标线两端的标记类型, 可以是一个数组分别指定两端, 也可以是单个统一指定, 具体格式见 data.symbol。
symbol: Optional[str] = None,

# 标线两端的标记大小, 可以是一个数组分别指定两端, 也可以是单个统一指定。
symbol_size: Union[None, Numeric] = None,

# 标线数值的精度, 在显示平均值线的时候有用。
precision: int = 2,

# 标签配置项, 参考 `series_options.LabelOpts`
label_opts: LabelOpts = LabelOpts(),

# 标记线样式配置项, 参考 `series_options.LineStyleOpts`
linestyle_opts: Union[LineStyleOpts, dict, None] = None,
)
```

MarkAreaItem: 标记区域数据项

class pyecharts.options.MarkAreaItem

```
class MarkAreaItem(
    # 区域名称, 仅仅就是一个名称而已
    name: Optional[str] = None,
```

特殊的标注类型，用于标注最大值最小值等。

'min' 最大值。

'max' 最大值。

'average' 平均值。

type_: Sequence[Optional[str], Optional[str]] = (None, None),

在使用 type 时有效，用于指定在哪个维度上指定最大值最小值，可以是 0 (xAxis, radiusAxis), 1 (yAxis, angleAxis)。

默认使用第一个数值轴所在的维度。

value_index: Sequence[Optional[Numeric], Optional[Numeric]] = (None, None),

在使用 type 时有效，用于指定在哪个维度上指定最大值最小值。

这可以是维度的直接名称，例如折线图时可以是 x、angle 等、candlestick 图时可以是 open、close 等维度名称。

value_dim: Sequence[Optional[str], Optional[str]] = (None, None),

相对容器的屏幕 x 坐标，单位像素，支持百分比形式，例如 '20%'。

x: Sequence[Union[str, Numeric, None], Union[str, Numeric, None]] = (None, None),

相对容器的屏幕 y 坐标，单位像素，支持百分比形式，例如 '20%'。

y: Sequence[Union[str, Numeric, None], Union[str, Numeric, None]] = (None, None),

标签配置项，参考 `series_options.LabelOpts`

label_opts: Union[LabelOpts, dict, None] = None,

该数据项区域的样式，起点和终点项的 itemStyle 会合并到一起。参考 `series_options.ItemStyleOpts`

itemstyle_opts: Union[ItemStyleOpts, dict, None] = None,

)

MarkAreaOpts: 标记区域配置项

class pyecharts.options.MarkAreaOpts

```
class MarkAreaOpts(  
    # 图形是否不响应和触发鼠标事件，默认为 False，即响应和触发鼠标事件。  
    is_silent: bool = False,  
  
    # 标签配置项，参考 `series_options.LabelOpts`  
    label_opts: LabelOpts = LabelOpts(),  
  
    # 标记区域数据，参考 `series_options.MarkAreaItem`  
    data: Sequence[Union[MarkAreaItem, Sequence, dict]] = None,  
  
    # 该数据项区域的样式。参考 `series_options.ItemStyleOpts`  
    itemstyle_opts: ItemStyleOpts = None,  
)
```

EffectOpts: 涟漪特效配置项

class pyecharts.EffectOpts.EffectOpts

```
class EffectOpts(  
    # 是否显示特效。  
    is_show: bool = True,
```

```
# 波纹的绘制方式，可选 'stroke' 和 'fill', Scatter 类型有效。
brush_type: str = "stroke",

# 动画中波纹的最大缩放比例，Scatter 类型有效。
scale: Numeric = 2.5,

# 动画的周期，秒数，Scatter 类型有效。
period: Numeric = 4,

# 特效标记的颜色
color: Optional[str] = None,

# 特效图形的标记。
# ECharts 提供的标记类型包括 'circle', 'rect', 'roundRect', 'triangle',
# 'diamond', 'pin', 'arrow', 'none'
# 可以通过 'image://url' 设置为图片，其中 URL 为图片的链接，或者 dataURI。
symbol: Optional[str] = None,

# 特效标记的大小，可以设置成诸如 10 这样单一的数字，也可以用数组分开表示高和宽，
# 例如 [20, 10] 表示标记宽为 20，高为 10。
symbol_size: Optional[Numeric] = None,

# 特效尾迹的长度。取从 0 到 1 的值，数值越大尾迹越长。Geo 图设置 Lines 类型时有效。
trail_length: Optional[Numeric] = None,

)
```

AreaStyleOpts：区域填充样式配置项

class pyecharts.options.AreaStyleOpts

```
class AreaStyleOpts(  
    # 图形透明度。支持从 0 到 1 的数字，为 0 时不绘制该图形。  
    opacity: Optional[Numeric] = 0,  
    # 填充的颜色。  
    # 颜色可以使用 RGB 表示，比如 'rgb(128, 128, 128)'，如果想要加上 alpha 通道表示不透明度，  
    # 可以使用 RGBA，比如 'rgba(128, 128, 128, 0.5)'，也可以使用十六进制格式，比如 '#ccc'。  
    # 除了纯色之外颜色也支持渐变色和纹理填充  
    #  
    # 线性渐变，前四个参数分别是 x0, y0, x2, y2，范围从 0 - 1，相当于在图形包围盒中的百分比，  
    # 如果 globalCoord 为 `true`，则该四个值是绝对的像素位置  
    # color: {  
    #     type: 'linear',  
    #     x: 0,  
    #     y: 0,  
    #     x2: 0,  
    #     y2: 1,  
    #     colorStops: [{  
    #         offset: 0, color: 'red' // 0% 处的颜色  
    #     }, {  
    #         offset: 1, color: 'blue' // 100% 处的颜色  
    #     }],  
    #     global: false // 缺省为 false  
    # }  
    #  
    # 径向渐变，前三个参数分别是圆心 x, y 和半径，取值同线性渐变  
    # color: {
```

```

#     type: 'radial',
#     x: 0.5,
#     y: 0.5,
#     r: 0.5,
#     colorStops: [{
#         offset: 0, color: 'red' // 0% 处的颜色
#     }, {
#         offset: 1, color: 'blue' // 100% 处的颜色
#     }],
#     global: false // 缺省为 false
# }
#
# 纹理填充
# color: {
#     image: imageDom, // 支持为 HTMLImageElement, HTMLCanvasElement, 不支持路径字符串
#     repeat: 'repeat' // 是否平铺, 可以是 'repeat-x', 'repeat-y', 'no-repeat'
# }
color: Optional[str] = None
)

```

SplitAreaOpts: 分隔区域配置项

class pyecharts.options.SplitAreaOpts

```

class SplitAreaOpts(
    # 是否显示分隔区域。
    is_show=True,

```

```
# 分隔区域的样式配置项，参考 `series_options.AreaStyleOpts`  
areastyle_opts: AreaStyleOpts = AreaStyleOpts()  
)
```

MinorTickOpts：次级刻度配置项

class pyecharts.options.MinorTickOpts

```
class MinorTickOpts(  
    # 是否显示次刻度线。  
    is_show: bool = False,  
  
    # 次刻度线分割数，默认会分割成 5 段  
    split_number: Numeric = 5,  
  
    # 次刻度线的长度。  
    length: Numeric = 3,  
  
    # 次刻度线的样式  
    linestyle_opts: Union[LineStyleOpts, dict, None] = None,  
)
```

MinorSplitLineOpts：次级分割线配置项

class pyecharts.options.MinorSplitLineOpts


```
class MinorSplitLineOpts(  
    # 是否显示次分隔线。默认不显示。  
    is_show: bool = False,  
  
    # 次分隔线线宽。  
    width: Numeric = 1,  
  
    # 次分隔线线的类型。可选: 'solid', 'dashed', 'dotted'  
    type_: str = "solid",  
  
    # 图形透明度。支持从 0 到 1 的数字, 为 0 时不绘制该图形。  
    opacity: Union[Numeric, None] = None,  
  
    # 线的样式  
    linestyle_opts: Union[LineStyleOpts, dict, None] = None,  
)
```