# 安装

* 下载
* 添加引用：

<script src="https://code.highcharts.com/highcharts.js"></script>

# 第一个图表

1. Html标签：

<div id="container" style="width:100%; height:400px;"></div>

1. JavaScript脚本：

    document.addEventListener('DOMContentLoaded', function () {

        var myChart = Highcharts.chart('container', {

            chart: {

                type: 'bar'

            },

            title: {

                text: 'Fruit Consumption'

            },

            xAxis: {

                categories: ['Apples', 'Bananas', 'Oranges']

            },

            yAxis: {

                title: {

                    text: 'Fruit eaten'

                }

            },

            series: [{

                name: 'Jane',

                data: [1, 0, 4]

            }, {

                name: 'John',

                data: [5, 7, 3]

            }]

        });

    });

## 一般步骤：

1. 创建图表容器（div）
2. 配置参数
3. 绑定容器

# 绑定容器方法：

* 构造函数：

    var charts = Highcharts.chart('container', {

        // Highcharts 配置

    });

* 参数renderTo：

    var charts = Highcharts.chart({

        // Highcharts 配置

        chart: {

            renderTo: 'container'  // 或 document.getElementById('container')

        }

    });

* jQuery插件：

    $("#container").highcharts({

        // Highcharts 配置

    });

# 图表容器

## 宽度、高度

根据div标签的宽度和高度设定。

<div id="container" style="width:100%; height:500px;"></div>

width设置为百分比，图表将占据该百分比的宽度

div标签没有指定时，默认值为400px \* 400px

<div id="container"></div>

图表自适应宽度设置：在div的min-width属性设置

<div id="container" style="min-width:400px;height:400px"></div>

最小宽度为400px

## 图表样式

例：设置字体

            chart: {

                style: {

                    fontFamily: "",

                    fontSize: '12px',

                    fontWeight: 'bold',

                    color: '#006cee'

                }

            }

还可以通过chart.className绑定CSS类

# 指定图表类型：

通过 chart.type 来指定图表类型，表示默认图表类型，即如果 series 中没有指定 type， 那么图表的类型就由该属性来确定。

            chart: {

                type: 'bar'

            },

highcharts 支持的所有图表类型见 plotOptions。

# 标题

标题默认显示在图表的顶部，包括标题和副标题（subTitle），其中副标题是非必须的。设置标题和副标题的示例代码如下：

            title: {

                text: '我是标题'

            },

            subtitle: {

                text: '我是副标题'

            },

## 标题常用属性

|  |  |  |
| --- | --- | --- |
| 属性名 | 描述 | 默认值 |
| text | 标题的文字 | "Chart title" |
| align | 文字水平对齐方式，有left、center、right可选 | "center" |
| verticalAlign | 文字垂直对齐方式，有top、middle、bottom可选 | "" |
| useHTML | 是否解析html标签，设置解析后，可以使用例如a等html标签 | false |
| floating | 是否浮动，设置浮动后，标题将不占用图表区位置 | false |
| margin | 标题和图表区的间隔，当有副标题时，表示标题和副标题之间的间隔 | 15 |
| style | 文字样式，可以设置文字颜色、字体、字号，注意和css有略微的不同，例如font-size用fontSize、font-family用fontFamily表示 | {   color: '#3E576F',   fontSize: '16px' } |
| x | 相对于水平对齐的偏移量，可以是负数，单位是px | 0 |
| y | 相对于垂直对齐的偏移量，可以使负数，单位是px | 0 |

## 动态设置和获取标题

### 获取标题内容

通过Highcharts对象获取标题内容：

    var chart = Highcharts.chart(el, options);    // Highcharts构造函数

    var title = chart.title.textStr;             // 通过chart对象获取标题内容

### 动态设置标题

setTitle() 函数供动态设置标题用

setTitle (Object title, object subtitle, Boolean redraw)

参数说明：

* title : 标题对象
* subtitle: 副标题对象
* redraw：是否重绘，即设置标题后是否重新绘制图表，默认是false

实例：修改副标题

    var subtitle = {

        text:"我是新标题",

        style: {

            color: "#000",

            fontWeight: "bold"

        }

    };

    chart.setTitle(null, subtitle);     //设置副标题，第一个参数设置为null

### 在标题中添加链接

设置 useHTML为 true，然后在标题文字中加入 a 标签

实例：

            title: {

                useHTML: true,

                text: "Highcharts中文网 | <a href='https://www.highcharts.com.cn' target='\_blank'>中文教程</a>"

            }

### 隐藏标题

            title: {

                text: null

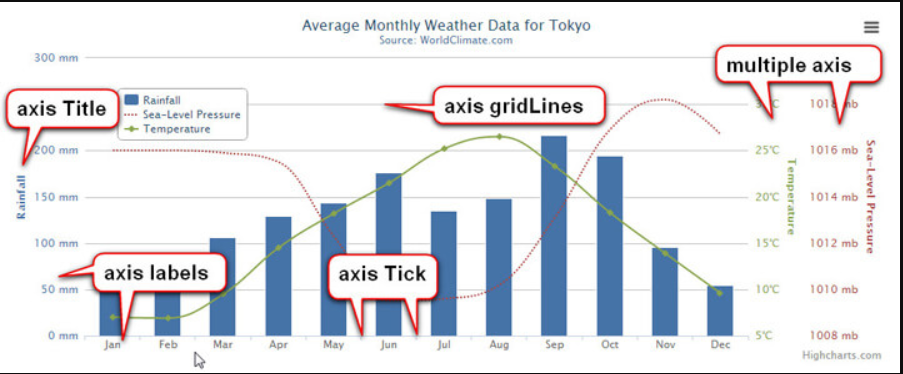
            }

# 坐标轴

笛卡尔图表（普通的二维数据图）都有X轴和Y轴，默认情况下，x轴显示在图表的底部，y轴显示在左侧

通过设置[chart.inverted = true](https://www.highcharts.com.cn/demo/highcharts/spline-inverted) 可以让x，y轴显示位置对调。

坐标轴组成示意图：



## 坐标轴组成：

### title：坐标轴标题

默认情况下，x轴为null（也就是没有title），y轴为数值'Value'

实例：分别设置坐标轴标题

            xAxis: {

                title: {

                    text: 'x轴标题'

                }

            },

            yAxis: {

                title: {

                    text: 'y轴标题'

                }

            }

### Labels：坐标轴刻度标签

Labels坐标轴标签（分类）。Labels常用属性有enabled、formatter、step、staggerLines

略

### Tick：坐标轴刻度

Tick为坐标轴刻度。默认情况下x轴刻度高(tickLength属性)为5px，宽为1px；y轴宽为0px(也就是不显示刻度)。Tick相关的属性主要有tickLength、tickWidth、tickColor、tickInterval、tickmarkPlacement。

### 坐标轴网格线

坐标轴网格线。默认情况下，x轴网格线宽度为0,y轴网格线宽度为1px。网格线共有三个属性可设置，分别是: gridLineWidth、gridLineColor、gridLineDashStyle

### 多轴

多个轴。在Highcharts中， 坐标可以是多个，最常见的是多个y轴。多轴存在时，Axis是一个数组，而在赋值时，通过Axis数组的下标与数据关联。

## type：坐标轴类型

坐标轴中，可以通过Type指定坐标轴类型，有linear、logarithmic、datetime、category可选，默认是：linear。

            // The types are 'linear', 'logarithmic' and 'datetime'

            yAxis: {

                type: 'linear',

            }，

            // Categories are set by using an array

            xAxis: {

                categories: ['Apples', 'Bananas', 'Oranges']

            }

### 线性轴

### 对数轴

### 时间轴

### 数组轴category（常用）

## 相关属性

### inverted图表反转

 Chart.inverted，当其值设置为true时，x轴和y轴显示的位置对调。

### reversed：**坐标轴刻度反转**

坐标轴大小方向反转。

### opposite：**轴倒置。主要控制坐标轴左右或上下显示**

### allowDecimals：**控制数轴是否显示小数**。

### min、max：**控制数轴的最小值和最大值**。

# 数据列

数据列是一组数据集合，例如一条线，一组柱形等。图表中所有点的数据都来自数据列对象

数据列的基本构造是：

            series: [{

                name: '',

                data: []

            }, {

                name: '',

                data: []

            }]

数据列配置是个数组，也就是数据配置可以包含多个数据列{}。

name 代表数据列的名字，并且会显示在数据提示框（Tooltip）及图例（Legend）中

## 数据列中的数据data

在数据列的 data 属性中，我们可以定义图表的数据数组，通常有三种定义方式：

### 数值数组

在这种情况下，配置数组中的数值代表 Y 值;

X 值则根据 X 轴的配置，

* 要么自动计算，要么从 0 起自增，或者是根据 pointStart 及 pointInterval 自增
* 在分类轴中， X 值就是 categoies 配置，

            data : [1, 4, 6, 9, 10]

### 包含两个值的数组集合

* 在这种情况下，集合中数组的第一个值代表 X， 第二个值代表 Y
* 如果第一个值是字符串，则代表该点的名字，并且 X 值会如 **1** 中所说的情况决定。

            data ： [ [5, 2], [6,3], [8,2] ]

### 数据点对象集合

在这种情况下，集合中元素都是数据点对象，对象中可以配置数据见 [plotOptions.series](http://api.hcharts.cn/highcharts#plotOptions.series) 或 plotOptions.{图表类型} 所列。

            data: [{

                name: "point 1",

                color: "#00ff00",

                y: 0

            }, {

                name: "Point 2",

                color: "#ff00ff",

                y: 5

            }]

## 数据点及标记

对特定点进行特殊标记，如颜色，形状。

略。

## 数据列配置

数据列共有三个级别的配置，权重从低到高依次如下：

* plotOptions.series ：针对所有类型图表有效，一般是通用配置。
* plotOptions.{图表类型} ：针对当前类型图表有效，一般是某一种图表的通用配置。
* series： 针对当前数据列有效

向上覆盖

## 常用属性：

### 动画（Animation）

### 颜色（Color）

### 点的选择（Selection）

### 线条宽度（lineWidth）

### 鼠标形状（cursor）

### 数据标签（dataLables）

### 线条样式（Dash Style）

### 数据列分区（zones）

# 颜色

## 数据列颜色

Highcharts 中数据列的颜色是通过 colors 来指定的，colors 是个颜色值数组，默认是：

colors: ['#7cb5ec', '#434348', '#90ed7d', '#f7a35c', '#8085e9',

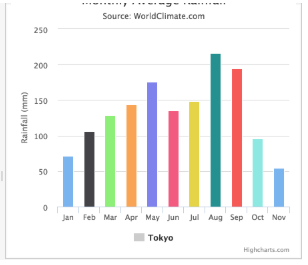
'#f15c80', '#e4d354', '#8085e8', '#8d4653', '#91e8e1']

共有 10个默认颜色，你可以修改颜色值或增加颜色个数来自定义图表数据列颜色。

数据列调用颜色的方式是第 n 个数据列使用第 n 个颜色数组里的值，当序列的数量超过颜色数组的长度，后续的序列将会从头调用。

## 柱形图的颜色

colorByPoint 决定了图表是否给每个数据列或每个点分配一个颜色，默认值是 false， 即默认是给每个数据类分配颜色，设置为 true 则是给每个点分配颜色。



## 图表中文字颜色

图表中所有文字都设置字体、颜色等样式，一般是通过 style.color 来设置文字颜色的

            style: {

                color: '#ff0000',

                fontSize: "12px",

                fontWeight: "blod",

                fontFamily: "Courier new"

            }

下面列举图表中常见文字的配置位置：

* 标题样式：title.style 及 subtitle.style
* 坐标轴标签样式：xAxis.lables.style 及 yAxis.lables.style
* 图例文字样式：legend.itemStyle
* 数据提示框文字样式：tooltip.style

# 数据提示框

数据提示框指的当鼠标悬停在某点上时，以框的形式提示该点的数据，比如该点的值、数据单位等。数据提示框内提示的信息完全可以通过格式化函数动态指定；

通过设置 tooltip.enabled = false 即可不启用提示框。

## 提示框外观

数据提示框的外观的常用配置：

            tooltip: {

                backgroundColor: '#FCFFC5',   // 背景颜色

                borderColor: 'black',         // 边框颜色

                borderRadius: 10,             // 边框圆角

                borderWidth: 3,               // 边框宽度

                shadow: true,                 // 是否显示阴影

                animation: true,               // 是否启用动画效果

                style: {                      // 文字内容相关样式

                    color: "#ff0000",

                    fontSize: "12px",

                    fontWeight: "blod",

                    fontFamily: "Courir new"

                }

            }

## 提示框内容

### 格式化函数

### 格式化字符串

### 时间格式化

### html 内容

### 值的前缀、后缀及小数点

### 共享的提示框（Shared）

## 其他特性

### 十字准星

### 固定位置显示提示框

### 鼠标行为

# 图例

图例是图表中用不同形状、颜色、文字等 标示不同数据列，通过点击标示可以显示或隐藏该数据列；

通过设置 legend.enabled = true | false 来打开或关闭图例。

## 图例样式

### 图例容器样式

| **参数名** | **解释** | **默认值** |
| --- | --- | --- |
| backgroundColor | 背景颜色 | null |
| borderColor | 边框颜色 | '#909090' |
| margin | 外边距 | 15 |
| padding | 内边距 | 8 |
| maxHeight | 最大高度 | null |
| navigation | 导航，当设置了最大高度后，图例无法完整显示时，则会用导航的形式展示（分页），[详见API文档](http://api.hcharts.cn/highcharts#legend.navigation) |  |
| shadow | 图例阴影效果，赋值可以是 boolean 或 Object，[详见API文档](http://api.hcharts.cn/highcharts#legend.shadow) | false |
| width | 图例宽度 | null |
| verticalAlign | 垂直对齐方式，有 'top'， 'middle' 及 'bottom' 可选 | 'bottom' |
| useHTML | 是否以HTML形式渲染（默认是SVG渲染），当使用 HTML 模式渲染是，图例导航无效 | false |

### 图例项样式

| **参数名** | **解释** | **默认值** |
| --- | --- | --- |
| itemDistance | 图例项间距 | 20 |
| itemStyle | 图例样式 | itemStyle: { cursor: 'pointer', color: '#3E576F' } |
| itemHiddenStyle | 图例隐藏时的样式 | itemHiddenStyle: { color: '#CCC' } |
| itemHoverStyle | 图例鼠标划过时样式 | itemHoverStyle: { color: '#000' } |
| itemMarginBottom | 图例项底边距 | 0 |
| itemMarginTop | 图例项顶部边距 | 0 |
| itemWidth | 图例项宽度 | null |
| symbolHeight | 图例项标示高度 | 12 |
| symbolPadding | 图例项标示内边距 | 5 |
| symbolRadius | 图例项标示圆角 | 2 |
| symbolWidth | 图例项标示宽度 | 16 |

## 图例内容及定位

### 图例内容

| **参数名** | **解释** | **默认值** |
| --- | --- | --- |
| labelFormat | 图例文字格式化字符串 | {name} |
| labelFormatter | 图例内容格式化函数 | function() { return this.name} |
| reversed | 是否倒序 | false |
| rtl | 文字是否显示在符号前面，主要针对阅读习惯从右至左的用户 | false |
| title | 图例标题 | title: { text: null, style: { fontWeight: "blod"}} |

### 定位

| **参数名** | **解释** | **默认值** |
| --- | --- | --- |
| align | 图例在图表中的对齐方式，有 “left”, "center", "right" 可选 | “center” |
| floating | 图例是否浮动，设置浮动后，图例将不占位置 | false |
| layout | 图例内容布局方式，有水平布局及垂直布局可选，对应的配置值是： “horizontal”， “vertical” | "horizontal" |
| x | 水平偏移 | 0 |
| y | 竖直偏移 | 0 |

## 图例点击事件

### 默认图例点击事件

### 禁用图例点击隐藏效果

### 饼图图表点击事件

### 自定义图例点击事件

# 版权信息

图表版权信息。显示在图表右下方的包含链接的文字，默认的文字是 Highcharts，链接是Highcharts官网地址。通过指定credits.enabled=false即可不显示该信息。

## 版权信息的常用属性

|  |  |  |
| --- | --- | --- |
| **属性名** | **描述** | **默认值** |
| enabled | 是否显示版权信息 | true |
| href | 版权信息点击之后指向的URL （设置自己版权信息地址时记得加 “http://“ ） | http://www.highcharts.com |
| text | 显示的版权信息文字 | Highcharts.com |
| position | 文字显示位置。支持的属性有align （左右对齐）, verticalAlign（上下对齐）, x（水平偏移）, y（竖直偏移） | position: {     align: 'right',     x: -10,     verticalAlign: 'bottom',     y: -5 } |
| style | 版权信息标签的CSS样式 | style: {     cursor: 'pointer',     color: '#909090',     fontSize: '10px' } |

## 版权信息常见设置

### 屏蔽版权信息

            credits: {

                enabled: false // 禁用版权信息

            }

### 设置位置

默认显示右下角，**align**可配置参数：left、right、center；

**verticalAlign** 可配置参数： bottom、top、middle ；

**x** 和 **y** 分别配置的是指定位置的偏移量，负数代表向左或向上偏移，正数代表向右或者向下偏移。

### 完整的例子

            credits: {

                // enabled:true,                    // 默认值，如果想去掉版权信息，设置为false即可

                text: 'www.hcharts.cn',             // 显示的文字

                href: 'http://www.hcharts.cn',      // 链接地址

                position: {                         // 位置设置

                    align: 'left',

                    x: 400,

                    verticalAlign: 'bottom',

                    y: -100

                },

                style: {                            // 样式设置

                    cursor: 'pointer',

                    color: 'red',

                    fontSize: '30px'

                }

            }

# 读取 JSON 并创建图表

    $(document).ready(function () {

        var options = {

            chart: {

                type: 'spline'

            },

            series: [{}]

        };

        $.getJSON('data.json', function (data) {

            options.series[0].data = data;

            var chart = Highcharts.chart('container', options);

        });

    });