

目录

1. Highcharts 简介.....	2
➤ 下载的 highcharts 文件结构介绍.....	2
2. 使用说明.....	3
➤ 基础图表基本函数使用配置.....	3
➤ 高级功能使用配置.....	3
3. Highcharts 图表的基本结构.....	3
➤ 图表基本组成.....	3
➤ 坐标轴组成部分详解.....	4
4. Highcharts 相关函数.....	5
5. 创建一个图表对象实例.....	6
6. Highcharts 相关参数详解.....	10
1. chart 图表选项设定.....	10
2. 图表标题 title&subtitle.....	11
3. 坐标轴 Axis 选项设定.....	12
4. series 数据序列.....	13
5. 数据提示框 tooltip.....	15
6. drilldown 深度数据详细.....	16
7. 图例选项设定 legend.....	17
8. plotOptions 数据点相关选项设定.....	18
9. 颜色 color 选项设定.....	19
10. Credits 版权信息显示选项.....	19
11. exporting 导出选项属性设定.....	19
12. Labels 标签选项设定.....	20
13. Loading 选项设定.....	20
14. Navigation 导航选项设定.....	20
15. nodata 选项设定.....	21
16. Pane 窗格选项设定.....	21
7. 导出服务器配置.....	21
➤ Highcharts 导出服务器配置——PHP 版.....	21
➤ Highcharts 导出服务器配置——JAVA 版.....	23
8. 图表与数据.....	24

Highcharts

1. Highcharts 简介

Highcharts:功能强大、开源、美观、图表丰富、兼容绝大多数浏览器的纯 **js** 图表库。 **Highcharts** 是一款纯 **javascript** 编写的图表库，能够很简单便捷的在 **Web** 网站或 **Web** 应用中添加交互性的图表。**Highcharts** 目前支持直线图、曲线图、面积图、柱状图、饼图、散点图等多达 **18** 种不同类型的图表如下表所示：

➤ 下载的 highcharts 文件结构介绍

名称	修改日期	类型	大小
examples	2014/1/1...	文件夹	例子代码目录
exporting-server	2014/1/1...	文件夹	导出服务器目录
gfx	2014/1/1...	文件夹	例子用到的图片
graphics	2014/1/1...	文件夹	
js	2014/1/1...	文件夹	所有js文件
index.htm	2014/1/1...	Chrome HTML Document	例子入口页面

打开index.htm即可看到Highcharts提供的所有例子



2. 使用说明

➤ 基础图表基本函数使用配置

Highcharts 运行需要两个 js 文件，**Highcharts.js** 及 jQuery、MooTools、Prototype 中的一个框架文件;或者只用 Highcharts Standalone Framework 这个框架文件

1. **最简单的**: 使用 jQuery 的 js 框架 (**最好用 1.8.2 以上版本的**) 和 **Highcharts.js**
2. Highcharts.js 搭配 MooTools 或 Prototype 框架
注: 但是需要额外引入 Highcharts 提供的相应的 js 适配器来支持
3. 只用 Highcharts Standalone 框架, 较小有助于减少网络流量, 获得更好的用户体验

➤ 高级功能使用配置

显示 3D 图表设置: 需要加载 highcharts-3d.js 文件

将笛卡尔坐标图显示为极坐标图: 需要加载 highcharts-more.js 文件

图表导出功能配置文件: 需要添加 exporting.js 文件

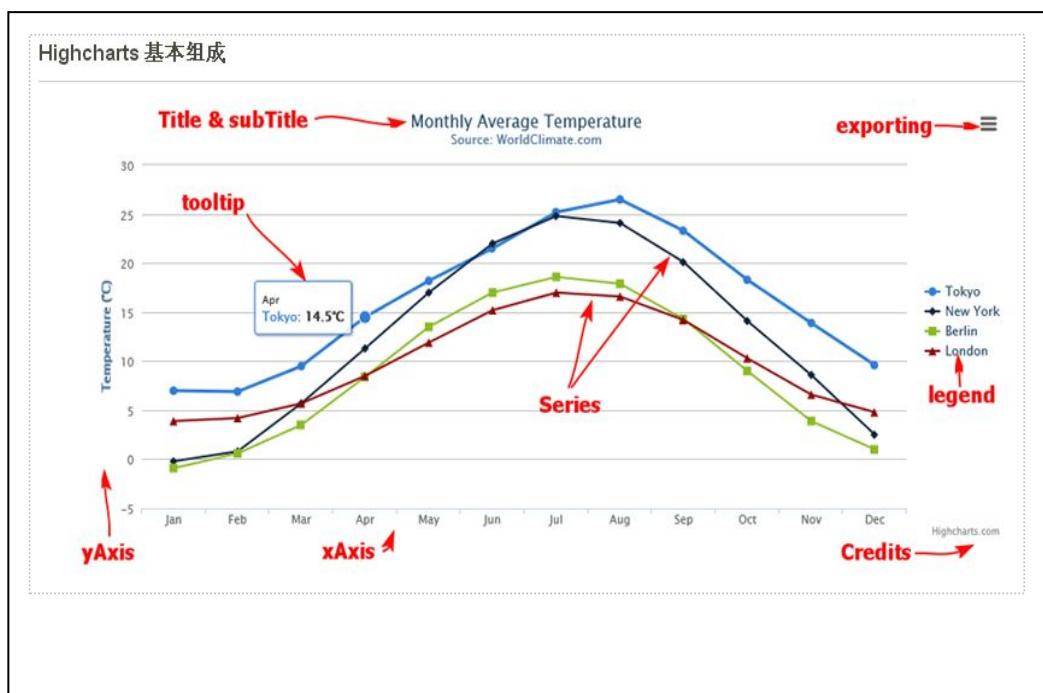
图表主题配置: Highcharts 提供的主题文件存在 **/Highcharts-x.x.x/js/themes/** 目录下只需要引入该文件即可

若图表需要数据深度显示功能: 加载 drilldown.js 文件

从网页上获取数据来显示需要: 加载 data.js 文件

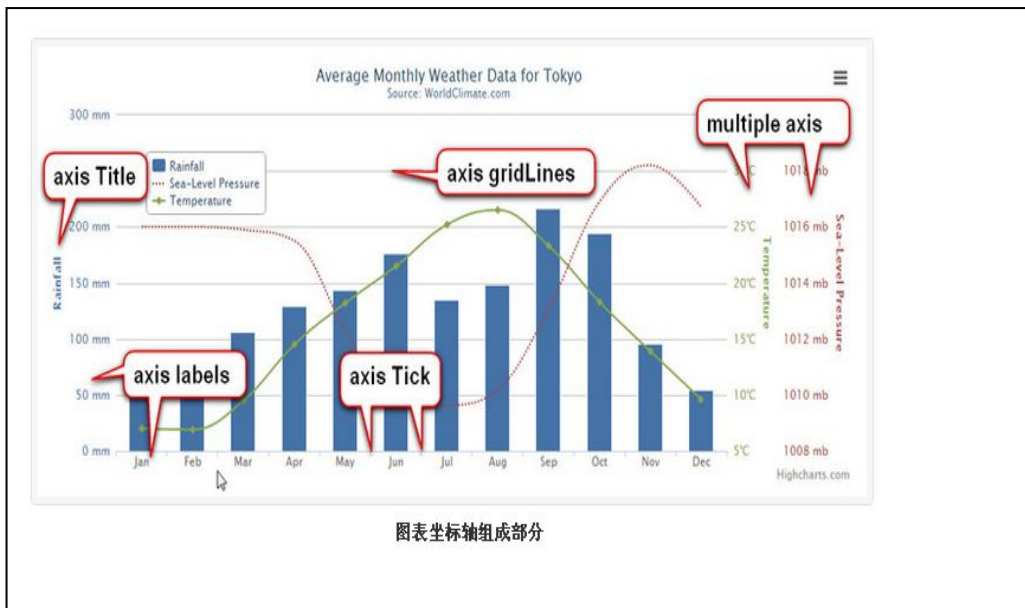
3. Highcharts 图表的基本结构

➤ 图表基本组成



英文名	中文名	描述
lang	语言文字对象	所有Highcharts文字相关的设置
chart	图表	图表区、图形区和通用图表配置选项
colors	颜色	图表数据列颜色配置，是一个颜色数组
credits	版权信息	Highcharts在图表的右下方放置的版权信息及链
drilldown	向下钻取	向下钻取数据，深入到其中的具体数据
exporting	导出模块	导出功能配置，导出即将图表下载为图片或打印图表
labels	标签	可以放置到图表区域内任何位置的HTML标签
legend	图例	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
loading	加载中	加载选项控制覆盖绘图区的加载屏的外观和文字
navigation	导航	导出模块按钮和菜单配置选项组
noData	没有数据	没有数据时显示的内容
pane	分块	针对仪表盘图和雷达图专用的配置，主要设置弧度及背景色
plotOptions	数据点配置	针对不同类型图表的配置。Highcharts所有图表类型请看下表
series	数据列	图表上一个或多个数据系列，比如图表中的一条曲线，一个柱形
title	标题	包括即标题和副标题，其中副标题为非必须的
tooltip	数据点提示框	当鼠标滑过某点时，以框的形式提示改点的数据，比如该点的值，数据单位等
Axis	坐标轴	包括x轴和y轴。多个不同的数据列可共用同一个X轴或Y轴，当然，还可以有两个X轴或Y轴，分别显示在图表的上下或左右。

➤ 坐标轴组成部分详解



4. Highcharts 相关函数

Highcharts 其实就像是用 javascript 编写的一个类对象，其相关参数如下：

全局参数

```
Highcharts.setOptions({
  global: {
    VMLRadialGradientURL: http://code.highcharts.com/vml-radial-gradient.png
    canvasToolsURL: http://code.highcharts.com/canvas-tools.js
    timezoneOffset: 0
    useUTC: true
  }
  lang: {
    contextButtonTitle: Chart context menu
    decimalPoint: .
    downloadJPEG: Download JPEG image
    downloadPDF: Download PDF document
    downloadPNG: Download PNG image
    downloadSVG: Download SVG vector
    drillUpText: Back to {series.name}
    loading: Loading...
    months: [ "January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December" ]
    noData: No data to display
    numericSymbols: [ "k", "M", "G" ]
    printChart: Print chart
    resetZoom: Reset zoom
    resetZoomTitle: Reset zoom level
    shortMonths: [ "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" ]
    thousandsSep: ,
    weekdays: [ "Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday" ]
  }
});
```

对象参数

```
$("#container").highcharts({
  chart: {...}
  colors: [...]
  credits: {...}
  drilldown: {...}
  exporting: {...}
  labels: {...}
  legend: {...}
  loading: {...}
  navigation: {...}
  noData: {...}
  pane: {...}
  plotOptions: {...}
  series: [...]
  subtitle: {...}
  title: {...}
  tooltip: {...}
  xAxis: {...}
  yAxis: {...}
});
```

- **全局参数**只能够通过 Highcharts.setOptions()来进行设定
- **对象参数**每个图表对象都可以进行单独设定
- 在使用 highcharts 创建图表的时候需要设定的相关选项主要如下图显示，只需要进行一些基本的配置，其它的使用默认的就很好。可是若想要得到比较完美的图表就需要对具体的属性信息进行详细的配置。

```
chart: {...} // 配置 chart 图表区
colors: [...] // 配置主体显示颜色（多个线条和柱体的颜色顺序的）
credits: {...} // 配置右下角版权链接
exporting: {...} // 配置导出及打印
global: {...} // HighCharts 国际化方法调用
labels: {...} // HTML 标签，可以放置在绘图的任何位置
lang: {...} // 语言对象属性配置
legend: {...} // 配置图例选项
loading: {...} // 配置图表加载选项
navigation: {...} // 配置导出按钮属性
pane: {...} // 仅适用于极坐标图表和角仪表
plotOptions: {...} // 配置数据点选项
series: [...] // 配置数据列选项
subtitle: {...} // 配置副标题
title: {...} // 配置标题
tooltip: {...} // 配置数据点提示框
xAxis: {...} // 配置 x 轴选项
```

5. 创建一个图表对象实例

- `$("#container").highcharts();` // 直接初始化参数创建图表对象，图表在 html 页面创建在以 `container` 为 id 的标签上

✧ 需要的 js 插件

```
<script type="text/javascript" src="http://libs.baidu.com/jquery/1.8.2/jquery.min.js"></script><!--jQuery文件-->
<script src="../../js/highcharts.js"></script><!--图表插件-->
<script src="../../js/modules/exporting.js"></script><!--图片导出模块配置-->
```

✧ 在 html 页面上为图表创建一个容器

```
<div id="container" style="min-width: 310px; height: 400px; margin: 0 auto"></div>
```

✧ Javascript 代码创建图表对象

```
$(function () {
    $('#container').highcharts({

        //=====
        // 图表相关属性配置
        //
        chart: {
            type: 'line' // 图表默认数据类型配置
        },
        //=====
        // 图表标题以及相关属性设置
        //
        title: {
            text: 'Monthly Average Temperature'
        },
        subtitle: {
            text: 'Source: WorldClimate.com'
        },

        //=====
        // 坐标轴设置
        xAxis: {
            // 坐标轴分组刻度值设置
            categories: ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'],
            tickInterval: 1 // 设置刻度的密度
        },
        yAxis: {
            title: {
                text: 'Temperature (°C)'
            }
        },
    },
});
```

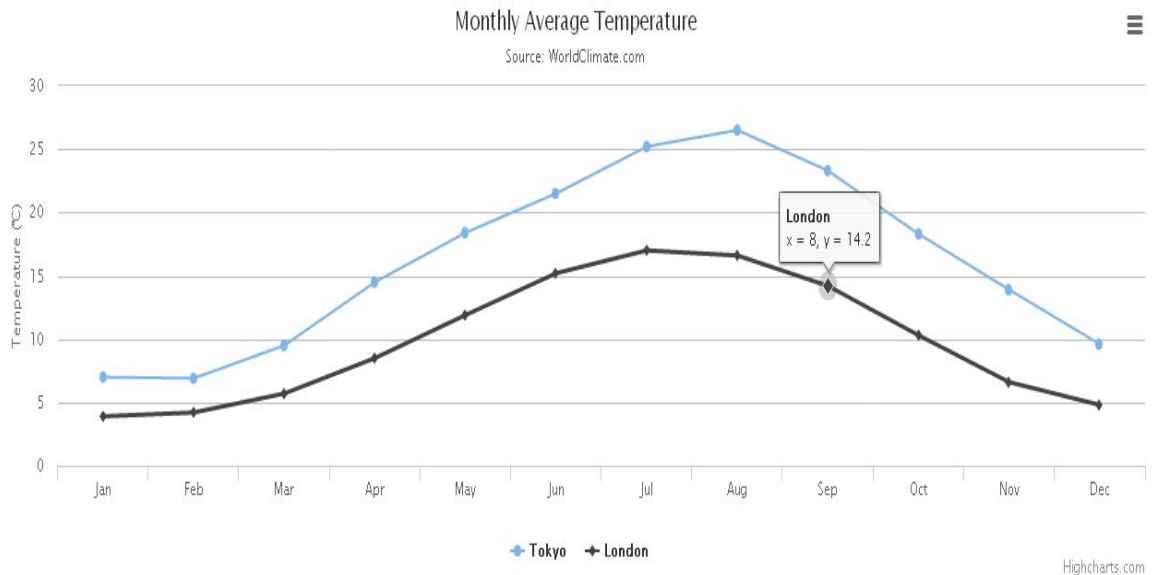


```
//=====
//数据序列
//存放的是显示在图表上的数据系列数组
//中有多少个数据那么在图表上就会显示几组数据
//
series:[{
  name: 'Tokyo',//序列的名称
  //type:'column',//这里设置的是这个数据序列显示的图表类型，若不设置那么使用的图表的总体属性设置
  data: [7.0, 6.9, 9.5, 14.5, 18.4, 21.5, 25.2, 26.5, 23.3, 18.3, 13.9, 9.6]
}, {
  name: 'London',
  data: [3.9, 4.2, 5.7, 8.5, 11.9, 15.2, 17.0, 16.6, 14.2, 10.3, 6.6, 4.8]
}],

//=====
//其它图表属性对象，每个属性之间用','号隔开
//
// add other feature here

//=====
//数据提示框，当鼠标划过的时候显示的提示信息
tooltip: {
  headerFormat: '<b>{series.name}</b><br />',
  pointFormat: 'x = {point.x}, y = {point.y}'
},
});
```

◇ 实例 (1) 图表效果（折线型图表）



- `var chart=new Highcharts.Chart(options);` //带初始化参数这里 options 定义里图表的各种属性信息，可以只是对图表的基本配置，不带数据；也可以是一个完整的图表

✧ Options 对象设定图表基本属性

```
//图表属性，不包含数据
var options = {

  chart: { //图表属性设置
    renderTo: 'container', //这个图表被创建在以container为id的div层上
    type: 'spline', //图表类型
  },
  title: { //标题
    text: '苏州市2012年天气情况表',
    x: -20 //center
  },
  //=====
  //坐标轴设定
  xAxis: {
    //这里没有对X轴进行设定
  },
  yAxis: [ //多y轴设定，使得yAxis变为一个数组
    //=====第一条y轴，位置默认在左侧
    {
      title: {
        text: 'Temperature (°C)'
      },
      plotLines: [{ //数据线的属性设定
        value: 0,
        width: 1,
        color: '#808080'
      }]
    },
    //=====第二条y轴
    {
      title: {
        text: 'Sunshinehours (h)'
      },
      plotLines: [{
        value: 0,
        width: 1,
        color: '#334455'
      }],
      opposite: true //设定其位于与默认轴线位置相对
    }
  ],
  //数据序列
  series: [
    //没有数据
    { //第一个数据系列，使用chart设置的spline图表类型显示数据
      name: '温度',
      zIndex: 1,
    },
    { //第二个数据系列
      name: '日照时间',
      type: 'column', //设置该数据系列使用的图表类型
      yAxis: 1,
      zIndex: 0
    }
  ]
}; //end of options=====
```


✧ queryData()函数从数据库获取数据

```
//初始函数，从数据库取数据
$(function ()
{
    queryData();//执行该函数，从数据库获取数据后创建图表
});

var categories = [];
var datal = [];
var data2 = [];
//Ajax 获取数据并解析创建Highcharts图表
function queryData()
{
    $.ajax({
        url: 'get_data.php',
        dataType: "json",
        success: function(data) {
            //将返回的数据进行解析
            $.each(data, function(i,n) {
                if(i==12) categories[i] = '全年';
                else categories[i] = n[0]+'月';
                datal[i] = n[1]*1;
                data2[i] = n[2]*1;
            });

            //为x轴设定刻度值
            options.xAxis.categories = categories;
            //为各个数据系列的data赋值
            options.series[0].data = datal;
            options.series[1].data = data2;

            //利用options对象作为参数，创建图表对象显示图表对象
            chart = new Highcharts.Chart(options);
        }
    });
}
```

✧ get_data.php 文件

```
<?php

require('config.php');

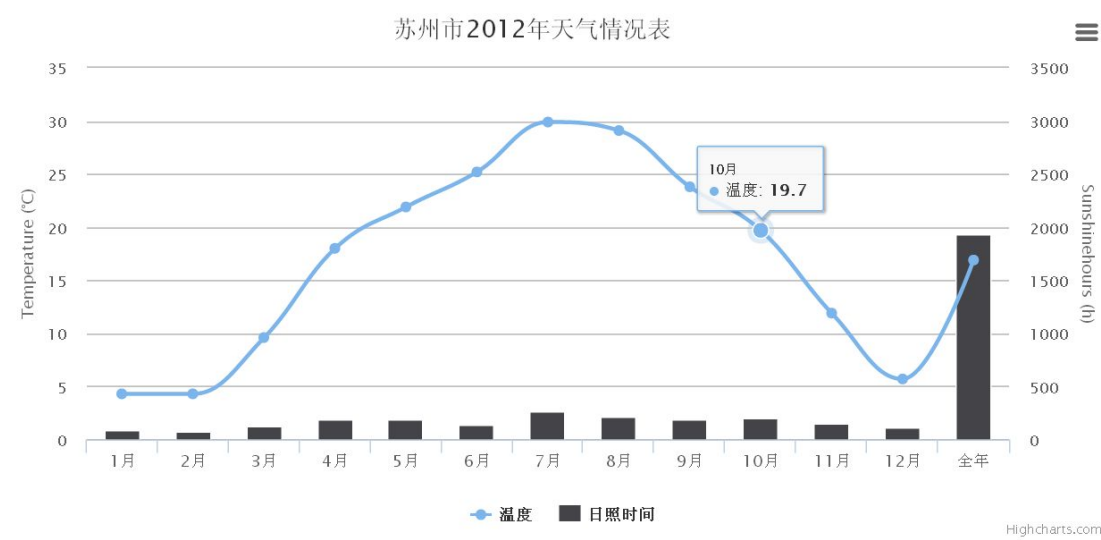
$sql_data="select Month, Temper, sunhours from soochow_weather ";
$res_data=mysql_query($sql_data) or die(" 查询出错<br/>");

$data=array();
while($row=mysql_fetch_row($res_data))
{
    $data[]=$row;
}

echo json_encode($data);

?>
```

◇ 实例(2)图表效果（曲线和柱状混合图表）



6. Highcharts 相关参数详解

1. chart 图表选项设定

参数	描述	默认值
Type:	图表的默认样式 (基本属性)	line
renderTo	图表存放的 html 元素的 id(基本属性)	null
height	图表高度 (外观)	null
width:	图表宽度 (外观)	null
shadow	图表是否启用阴影效果	false
backgroundColor	背景颜色 (外观)	#FFFFFF
borderColor	边框颜色 (外观)	#4572A7
borderRadius	边框圆角半径 (外观)	5.0
borderWidth:	边界宽度 (外观)	0.0
margin	图表的外边框到绘图区域的距离,该项设定包含以下属性 (外观)	null
marginTop	上边距	null
marginRight	右边距	null
marginBottom	下边距	null
marginLeft	左边距	null
spacing	图表边界到内容的距离, 该项设定包含以下属性 (外观)	[10, 10, 15, 10].
spacingTop	上	10
spacingRight	右	10
spacingBottom	下	15
spacingLeft	左	10

参数	描述	默认值
plotBackgroundColor	绘图区域背景色 (外观)	null
plotBackgroundImage	绘图区域背景图表 url(外观)	null
plotBorderColor	绘图区的边框颜色(外观)	#C0C0C0
plotBorderWidth	绘图区域的边框宽度(外观)	0.0
plotShadow	绘图区域是否启用阴影效果(外观)	false
Events	图表的事件监听器，当监听的事件发生时执行相应的自定义函数。包括：addSeries、click、drilldown、load、redraw、selection	null
options3d	图表 3D 属性设置，默认不启用图表 3D 显示	不启用
polar	是否显示为极坐标图表	false
pankey	设置一个副键，使得图表在放缩与拖动之间进行转换	null
panning	是否启用拖动功能	false
zoomtype	图表放缩方式设置 (x, y, xy)，默认不启用	null
alignTicks	多轴时最小刻度是否自动计算	true
animation	加载数据的时候是否启用动画效果	true
className	设置图表所在容器的类名，方便 css 统一样式	null
ignoreHiddenSeries	如果 true,一旦一个系列被隐藏，轴将会扩展剩余的可见系列	true
inverted	是否倒置轴，使得 x 轴时垂直 y 轴是水平	false
reflow	当浏览器窗口变动的时候图表是否需要自适应显示	true
selectionMarkerFill	选区背景颜色设置	rgba(69,114,167,0.25).
resetZoomButton	放缩重置按钮相关设置 位置 颜色等	详见 api

2. 图表标题 title&subtitle

参数	描述	默认值
text	标题显示的文字内容 (基本)	
align	对齐方式: center, right, left	center
verticalAlign	垂直对齐方式: top, middle, bottom	top
margin	标题到 plot 的距离	
floating	是否浮动	
style	Css 样式设置标题的字体大小和颜色	
useHTML	是否启用 text 中的 html 元素对标题进行修饰	false
x	标题所在的 x 位置	
y	标题所在的 y 位置	

3. 坐标轴 Axis 选项设定

参数	描述	默认值
title	坐标轴标题设置 这里只要设定 title 的 text 即可（基础）	
type	坐标轴类型设定：liner、category、datetime、logarithmic（基础）	
categories	若设定为：category 须给这个属性赋值（基础），如果没有赋值那么僵默认使用 series.name 作为刻度值显示	
Tick / minorTick	主刻度以及副刻度线相关属性的设定	
tickInterval/minorTickInterval	设置刻度的密度（基础）（刻度线）	
Tickcolor /minorTickColor	刻度线的颜色（刻度线）	#C0D0E0
tickLength/minorTickLength	刻度线的长度（刻度线）	
tickPixelInterval	刻度线的像素间距不适用 categorized axis（刻度线）	
tickmarkPlacement	刻度值放置的位置：on、between（刻度线）	
tickPosition/minorTickPosition	刻度线的位置：outside 、 inside（刻度线）	
tickWidth/minorTickWidth	刻度线的宽度（刻度线）	
minTickInterval	刻度线最小递增单位（刻度线）	
labels	刻度值相关属性设置，具体有很多属性详见 API，一般使用默认即可（刻度值）	
max	该坐标轴可显示的最大刻度	
maxcolor	最大刻度的颜色	
maxpadding	数据的递增幅度,为了使数据的显示不接近标示区的边沿可以设置小些	
min	该坐标轴可显示的最小刻度	
mincolor	最小刻度颜色	
minpadding	数据的递增幅度,为了使数据的显示不接近标示区的边沿可以设置小些	
minRange	可放缩的最小单位设置	
ceiling	限制该坐标轴数据的可显示的最大数值	
floor	限制坐标轴可显示的数据的最小值	
lineColor	坐标轴线的颜色（坐标线）	
lineWidth	坐标轴线的宽度（坐标线）	
linkedTo	参考的父坐标轴的下标（坐标线）	
alternateGridColor	主网格参考线设定（绘图区网格线）	
gridLineColor	颜色	
gridLineDashStyle	样式：实线、虚线等	
gridLineWidth	宽度	
gridZIndex	Z 索引	
minorGridLineColor	颜色 副网格线属性相关设定	
minorGridLineDashStyle	样式	
minorGridLineWidth	宽度	
plotBands	标示区	
plotLines	标示线	

参数	描述	默认
allowDecimals	是否允许刻度值为小数	True
offset	标示区到刻度的距离，单位为像素点	
opposite	是否将该坐标轴显示在默认位置的对立面	
reversed	是否将刻度递减显示	flase
reversedStacks	是否倒置堆叠显示	flase
showEmpty	当没有数据的时候是否显示该坐标轴	
showFirstLabel	是否显示第一个刻度值	
showLastLabel	是否显示最后一个刻度值	
startofWeek	当坐标轴显示刻度值为事件值得时候，这个属性用于设定每周的第一天是星期几：0-星期日 1-星期一	
startOnTick	是否以整刻度开始显示	
dateTimeLabelFormats	如果坐标轴时时间轴，这里可以设置其时间显示的格式	
endOnTick	坐标是否以整刻度结束	
id	给坐标轴设定一个 id	

4. series 数据序列

参数	描述	默认
name	序列的名称	null
type	这里设定该数据系列显示数据时，使用的图表类型	默认使用 chart 设置的 type 类型
Data {}	需要绘制的数据	null
id	该数据序列的 id 号	null
index	数据序列的下标	Null
legendIndex	该数据图例的下标	null
stack	一个（字符串或数值）做为分组标记，当数据堆叠显示的时候同组的数据会堆叠在一起显示	null
xAxis	当有多个坐标时，需指定该数据序列所对应的 x 坐标	0
yAxis	当有多个坐标时，需指定该数据序列所对应的 Y 坐标	0
zIndex	数据序列的 Z 轴坐标，默认的话按数据系列加入的先后顺序递增，最后面加入的数据序列显示在最顶端	

series 的 data 数据项的

参数	描述	默认值
color	单点数据颜色设定	Null
dataLabels	单点数据标签设定	Null
drilldown	该点所对应的在 drilldown.series 里面的数据序列的 id	Null
Event{click 、 mouseOut 、 mouseOver 、 remove 、 select 、 unselect、 update}	单点数据事件, 这些事件都可以自定义一个函数当事件发生时被执行	null
Id	单点数据的 id	
isIntermediateSum	只用于 Waterfall	false
isSum	只用于 Waterfall	false
legendIndex	只用于 pie	pie
sliced	只用于 pie, 分片显示	false
Marker{}	数据点标志设定	详见 api
x	X 值	null
y	Y 值	null

注：由上表可以知道一个数据点可以只是一个简单的数值，也可以是具有各种属性的复杂的对象。

简单的说数据 data 的格式可以是以下 3 种

- 一般的带有坐标轴刻度的只需要一个 Y 值就可以画出图表，Data 只需要设定为一个数值类型的数组即可。如下格式：

```
data: [0, 5, 3, 5]
```

- 多维数组类型的

1. 每个数据点有两个值分别对应 x 和 Y 坐标

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

2. 第一个点是字符串那么它被当成该点的名字，此时 x 轴的值 为自动增长

```
data: [[string, 2], [string, 3], [string, 2]]
```

3. 对于 range 图表 类型的数据序列

```
data: [[x, low, high] , [x, low, high] , [x, low, high] ]
```

- 数据点对象数组类型，每个数据是一个对象

```
data: [{
  name: 'Point 1',
  color: '#00FF00',
  y: 0
}, {
  name: 'Point 2',
  color: '#FF00FF',
  y: 5
}]
```


5. 数据提示框 tooltip

参数	描述	默认值
enabled	是否启用数据提示框	true
animation	是否动态显示	true
backgroundColor	数据提示框的背景颜色（外观）	rgba(255, 255, 255, 0.85)
borderColor	数据提示框的边框颜色（外观）默认和该数据点的颜色一样	auto
borderRadius	数据提示框的边框角度半径（外观）	3
borderWidth	数据提示框的边框线的宽度（外观）	1
crosshairs	显示数据提示框的时候是否显示十字准线这个值可以是一个 boolean 值、boolean 数组 或者 是一个二维数组对象 [{width:color:dashStyle:Zindex:},{}]	Null 默认不显示
dateTimeLabelFormats	对于坐标轴为 Dtatetime 类型的，该属性用于控制刻度值在 tooltip 上的显示样式	详细参见 api
followPointer	数据提示框的显示是否跟随鼠标的移动	false
followTouchMove	数据提示框的显示是否跟随指针的移动	false
headerFormat	数据提示框头部的显示样式设定	<code>{point.key}
</code>
formatter	提示框数据点数据显示样式，由一个回调函数返回 tooltip 的 text 内容的样式	null
footerFormat	提示框脚注设定	false
hideDlay	提示框隐藏延时	500ms
pointFormat	一个 html 标签显示数据点的值	<code>\u25CF {series.name}: {point.y}
</code>
positioner	一个函数返回一个位置的坐标，来设定 tooltip 的位置	null
shadow	是否启用阴影	true
shape	数据提示框的形状	callout
shared	在绘图区域中如果存在多个数据序列的时候，是否共用一个数据提示框	False

参数	描述	默认值
snap	数据提示框敏感距离	10
style	数据提示框样式 CSS 对象	详见 api
useHTML	是否启用 html 修饰 tooltip	false
valueDecimals	显示的小数点位数	
valuePrefix	数据前缀设定	null
valueSuffix	数据后缀设定	null
xDateFormat	当 X 轴为 datetime 类型的时候可以设定其在 tooltip 的显示样式	

6. drilldown 深度数据详细

参数	描述	默认值
series	一个数据系列对象详见 series (基本)	null
activeAxisLabelStyle	一个 CSS 对象，用于修饰那些拥有 drilldown 数据的刻度值	cursor: 'pointer', color: '#0d233a', fontWeight: 'bold', textDecoration: 'underline'
activeDataLabelStyle	一个 CSS 对象，用于修饰那些拥有 drilldown 数据的刻度值	同上
animation	drilldown 深度现实数据的时候是否需要动态显示。	true
drillUpButton	position: {x,y,align,verticalAlign} // 这些都是相对于 relativeTo 指定的对象所标示的位置 relativeTo: 'plotBox', // 默认 其他: sapcingBox theme: // 按钮的主题样式, 值为一个包含 SVG 属性的对象	

7. 图例选项设定 legend

参数	描述	默认值
enabled	是否启用图例模块	true
Align	水平对齐方式	center
verticalAlign	垂直对齐方式	bottom
floating	是否浮动显示	false
backgroundColor	背景颜色	null
borderColor	边框颜色	#909090.
borderRadius	边框角度半径	0
borderWidth	边框线宽度	0
itemDistance	图例之间的间距	20
itemHiddenStyle	图例隐藏条目样式设定	color: '#CCC'
itemHoverStyle	鼠标指针悬停在该条目上的时候条目的样式设定	color: '#000'
itemMarginBottom	图例条目间的底部边距设定	0
itemMarginTop	图例条目间的顶部边距设定	0
itemStyle	图例条目样式	{ "color": "#333333", "cursor": "pointer", "fontSize": "12px", "fontWeight": "bold" }.
itemWidth	图例条目宽度设定	null
labelFormat	图例条目显示的格式	{name}这里的 name 是 series 的 name
labelFormatter	由一个 function 返回值设定每个数据序列的格式	null
reversed	是否倒叙显示图例条目	false
rtl	是否将条目标记显示在文本的右侧	false
shadow	是否启用阴影显示	false
symbolHeight	图例条目相关属性设定	12
symbolPadding		5
symbolRadius		2
symbolWidth		16
useHTML	是否启用图例中修饰条目文本的 html	false
width	图例框的宽度	null
x	图例框相对于 align 的偏移量	0
y	图例框相对于 verticalAlign 的偏移量	0
layout	图例条目的显示样式 horizontal" or "vertical"	horizontal
lineHeight	行高	16

参数	描述	默认值
lineHeight	行高	16
margin	图例和坐标轴刻度值或者绘图区域间的边距	15
maxHeight	图例的最大高度	null
padding	图例的内边距	8
navigation	当图例条目过多时候会分页显示，这个属性可以对分页显示进行相关设定	详见 api

8. plotOptions 数据点相关选项设定

参数	描述	默认值
area	图表类型不一样, 且每种图表所具有的属性也有差别。可以通过这个选项为其进行个性设置	详细内容请参考 API
arearange		
areaSpline		
areasplinerange		
bar		
boxplot		
bubble		
cloumn		
cloumnrange		
errorbar		
funnel		
gauge		
heatmap		
line		
pie		
pyramid		
scatter		
solidgauge		
spline		
spline		
waterfall		
series	这是一个全局的, 使用于所有类型的数据序列	

9. 颜色 color 选项设定

参数	描述	默认
colors	一个数组包含图表数据序列的使用的默认颜色，当所有颜色都用完的时候，新的颜色将从最初的颜色从新开始	colors:['#4572A7','#AA4643','#89A54E','#80699B','#3D96AE','#DB843D','#92A8CD','#A47D7C','#B5CA92']

10. Credits 版权信息显示选项

参数	描述	默认
enabled	是否显示 credits 的文本	true
href	credits 标签的 href	http://www.highcharts.com
position	credits 标签的位置。align, verticalAlign, x 和 y	null
style	credits 标签的 CSS 样式	null
text	credits 标签的文本	Highcharts.com

11. exporting 导出选项属性设定

参数	描述	默认值
buttons	按钮外观设定一般用默认的就好	详见 api
chartOptions	导出图片的相关设定例如宽度，高度等	null
enabled	是否显示该模块	true
filename	导出的图片的名字默认值	chart
formAttributes	一个对象，为导出的图表设定其他的属性例如编码、加密方式等等	
scale	导出图表的放大比例倍数，这里设定为 2，那么导出的图表放大了一倍	2
sourceHeight	图表的原始高度	
sourceWidth	宽度	
type	导出图表的默认 MIME 类型	image/png
url	导出服务器地址（需要设定一个服务器，能够将 web 上的 SVG 字符串转化为相应格式的图片）	http://export.highcharts.com
width	导出图表的宽度设定	undefined

12. Labels 标签选项设定

参数	描述	默认值
Items {html, style}	Html 标签，其中 html：设定标签的 text，style 设定该标签的 CSS 样式	null
style	设定所有标签的 Css 样式	null

13. Loading 选项设定

参数	描述	默认
hideDuration	淡出效果的持续时间（以毫秒为单位）	100
labelStyle	加载标签的 CSS 样式	labelStyle: {fontWeight: 'bold',position: 'relative',top: '1em'}
showDuration	淡入效果的持续时间（以毫秒为单位）	100
style	覆盖在绘图区的加载页面的样式设定	style: {position: 'absolute',backgroundColor: 'white',opacity: 0.5,textAlign: 'center'}

说明： 加载选项通过图表的操作控制着覆盖在绘图区域的加载页面是否显示。 这个页面仅在显示调用 `chart.showLoading()`时，显示。

14. Navigation 导航选项设定

参数	描述	默认
buttonOptions	打印按钮和导出按钮 具体参考： exporting 打印和导出选项。	
menuItemHoverStyle	当导出图表被点击时弹出的菜单的单独选项的鼠标滑过的 CSS 样式	menuItemHoverStyle: {background: '#4572A5',color: 'FFFFFF'}
menuItemStyle	当导出图表被点击时弹出的菜单的单独选项的 CSS 样式	menuItemStyle: {padding: '0 5px',background: NONE,color: '#303030'}
menuStyle	当导出按钮被点击时，弹出菜单的 CSS 样式	menuStyle: {border: '1px solid #A0A0A0',background: 'FFFFFF'}

15. noData 选项设定

参数	描述	默认值
attr	设定一个 SVG 对象用于显示没有数据的图表	null
position	控制无数据标签的显示位置	<pre>{ "x": 0, "y": 0, "align": "center", "verticalAlign": "middle" }</pre>
style	一个 CSS 对象用于修饰无数据标签	<pre>{ "fontSize": "12px", "fontWeight": "bold", "color": "#60606a" }.</pre>

16. Pane 窗格选项设定

参数	描述	默认
background	背景的一个对象或一组对象。子选项包括 backgroundColor、innerWidth、outerWidth、borderWidth、borderColor	null
center	极坐标图或角度测量仪的中心点，像数组[x,y]一样定位。位置可以是以像素为单位的整数或者是绘图区域的百分比	['50%', '50%']
startAngle	x 轴或测量轴的开始度数，以度数的方式给出。0 是北	0
endAngle	x 轴极坐标或角度轴的最终度数，以度数的方式给出。0 是北	startAngle+360

说明：该属性只适用在极坐标图和角度测量仪。此可配置对象持有组合 x 轴和 y 周的常规选项。每个 x 轴和 y 轴都可以通过索引关联到窗格中

7. 导出服务器配置

➤ Highcharts 导出服务器配置——PHP 版

一、运行环境及准备工作

1、运行环境

- Php
Apache+php 环境
- java 运行环境
安装 java 运行环境

2、文件下载

- Highcharts
下载地址：<http://code.highcharts.com/zips/Highcharts-x.x.x.zip>
- Batik
下载地址：<http://apache.fayea.com/apache-mirror/xmlgraphics/batik>

将上述压缩包解压，得到 highcharts 和 batik 目录

二、搭建服务器

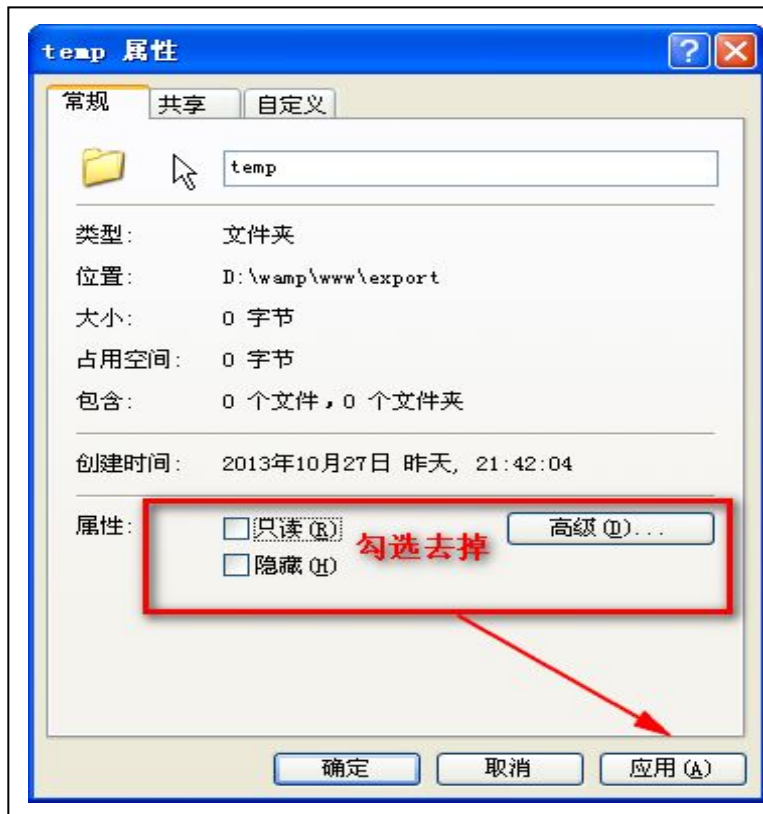
1、在 web 服务（apache）根目录新建 export 目录

2、拷贝 index.php 到 export

index.php 在 highchartsx.x.x\exporting-server\php\php-batik\目录下

3、在 export 目录里新建 temp 目录，并给定可读可写权限，linux 下给定目录 777 权限

- Window 下设置可读可写的方法如下



4、拷贝 batik-rasterizer.jar 到 export 目录，batik-rasterizer.jar 在 batik-目录下，batik-rasterizer.jar 的引用在 index.php 的 17 行，记得填写好 batik-rasterizer.jar 的路径

```
16 // Options
17 define ('BATIK_PATH', 'batik-rasterizer.jar');
```

5、拷贝 lib 目录到 export 目录

lib 目录存在 batik-1.7 目录下，注意是拷贝 lib 目录所有内容

这时候的 export 目录为



测试 batik 可用的方式是 cmd 中运行 `java -jar batik-rasterizer.jar`，如图

```
C:\Documents and Settings\admin>D:
D:\>cd wamp\www\export
D:\wamp\www\export>java -jar batik-rasterizer.jar
SUG Rasterizer options are:
-d <dir|file>          output directory. If there is a single input file, this
-m <contentType>      output mime type.
-w <width>             output width. This is a floating point value.
-h <height>           output height. This is a floating point value.
```

出现该提示表示环境可用

6、修改 Highcharts 的 exporting 属性，指定本地的导出服务地址

```
exporting :{
    url: 'http://你的服务器 ip 地址/export/index.php',
}
```

如果是想所有的图表都使用该地址，可以直接修改 exporting.js

➤ Highcharts 导出服务器配置——JAVA 版

一、运行环境以及准备工作

1. 安装 java 应用服务器（例如 Apache Tomcat, Jboss, TC server, Resin）
2. Java 运行环境
3. Maven 2 或者更新的版本
4. 在服务器上安装 PhantomJS
5. 下载 highcharts

下载地址: <http://code.highcharts.com/zips/Highcharts-x.x.x.zip>

二、配置

1. 拷贝 highcharts-export 到服务器上的适当位置（例如根目录下创建 export 文件夹）
highcharts-export 在 highchartsx.x.x\exporting-server\java\目录下
2. 打开 app-convert.properties 文件，阅读注释根据具体需要进行修改
位于 highcharts-export/highcharts-export-web/src/main/webapp/WEB-INF/spring 目录下

三、执行安装命令

1. 打开一个 dos/Shell 命令执行窗口，进入到导出服务器 highcharts-export 文件目录下
执行命令进行安装
\$ cd highcharts-export/
\$ mvn install

四、图表导出服务器地址修改

7、修改 Highcharts 的 exporting 属性，指定本地的导出服务地址

```
exporting :{
    url: 'http://你的服务器 ip 地址/export/highcharts-export'
}
```

如果是想所有的图表都使用该地址，可以直接修改 exporting.js

8. 图表与数据

1. 从文件获取数据

➤ Csv 文件

文件结构 -----没有限定只要数据取出来再预处理比较方便就好

下列中 data.csv 文件

首行是数据分组-----画横坐标用的

其次是数据序列格式为: name, value, value.....

```
例如: Categories,Apples,Pears,Oranges,Bananas
John,8,4,6,5
Jane,3,4,2,3
Joe,86,76,79,77
Janet,3,16,13,15
```

① 图表对象的初始化设置

空的数组存放数据分组数据和数据序列对象

```
var options = {
  chart: {
    renderTo: 'container',
    defaultSeriesType: 'column'
  },
  title: {
    text: 'Fruit Consumption'
  },
  xAxis: {
    categories: []
  },
  yAxis: {
    title: {
      text: 'Units'
    }
  },
  series: []
};
```

② 从 data.csv 文件解析数据

```
$.get('data.csv', function(data) {  
    // Split the lines  
    var lines = data.split('\n');  
  
    // Iterate over the lines and add categories or series  
    $.each(lines, function(lineNo, line) {  
        var items = line.split(',');  
  
        // header line contains categories  
        if (lineNo == 0) {  
            $.each(items, function(itemNo, item) {  
                if (itemNo > 0) options.xAxis.categories.push(item);  
            });  
        }  
  
        // the rest of the lines contain data with their name in the  
first  
        // position  
        else {  
            var series = {  
                data: []  
            };  
            $.each(items, function(itemNo, item) {  
                if (itemNo == 0) {  
                    series.name = item;  
                } else {  
                    series.data.push(parseFloat(item));  
                }  
            });  
            options.series.push(series);  
        }  
    });  
  
    // Create the chart  
    var chart = new Highcharts.Chart(options);  
});
```

➤ Xml 文件

优点：可以直接使用 jQuery doc 解析器解析文件数据

缺点：数据文件的冗余多

➤ JSON 文件格式

优点：不需要进行预处理，php 中有可以返回 javascript 数组的函数 json_encode()

对于这类文件可以使用 jQuery .getJSON 方法来加载它

2、从数据库获取数据

```
<?php
while ($row = mysql_fetch_array($result)) {
    $data[] = $row['value'];
}
?>

var chart = new Highcharts.Chart({
    chart: {
        renderTo: 'container'
    },
    series: [{
        data: [<?php echo join($data, ',') ?>],
        pointStart: 0,
        pointInterval
    }]
});
```