

# Bootstrap

---

响应式网页

1G 只能通话

2G 可以通话，短信

2.5G GPRS 通话，短信，上网 -- WML

3G IOS/Android(Linux) 通话，短信，视频，上网 -- HTML

Response Web page：响应式网页 / 自适应网页，2010

一个网页，会自动根据用户浏览设备不同，自动改变布局，被 PC/PAD/PHONE 正常浏览

---

## 响应式网页

---

手工创建使用 css3 Media 查询；浮动

响应式布局网页--手工创建(重点)

一个网页，会自动根据用户浏览设备不同,自动改变布局

可以被 PC/PAD/PHONE 正常浏览.

(1)声明viewport元标签

(2)使用流式布局

(3)所有容器/文字/图片 相对大小不使用绝对大小

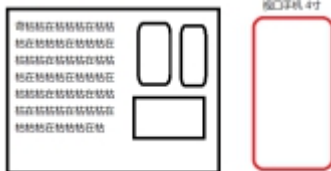
(4)使用 CSS3 Media Query 技术(重要原则)

viewport元素标签(视口)

早期的3G手机为浏览大尺寸网页,只能强行把页面缩小,  
导致网页内容,图片,文字很小



IOS 提出了"视口"概念, 视口用于盛放网页的内容尺寸大小可以任意指定



Android也借鉴概念

```
< meta name="viewport" content="width=device-width, initialscale=1.0">
```

width=device-width 表示视口宽度屏幕宽度initial-scale=1.0

初始化缩放比例 1.0

使用流式布局

实现方法: float:left

所有容器, 图片,文字 相对大小不使用绝对

```
.container{
    /\ width:500px;*/
    width:100%;
    /* font-size:19px; */
    font-size:1em; /*width:55px; */
    width:25%
}
```

## CSS3 媒体查询

- Media : 指浏览网页的设备,如screen(pc/pad/phone),print,tv

Media Query:查询出当前浏览网页的设备类型以及特性  
(解析度;对比度;尺寸;手持方向),有选择执行某些CSS代码  
而忽略另外一些.

- 媒体查询两种使用方法

(1) 根据媒体查询执行不同css文件

```
<link media="screen and (min-width:768px)" rel="stylesheet" href="pc.css" />
```

(2) 根据媒体查询执行不同css片段

```
@media screen and (min-width:768px) {
    选择器{样式}
```

}

## Bootstrap

---

<http://www.bootcss.com/>

Bootstrap 是 HTML;CSS;JS 框架，用于开发响应式布局

内容分为五部分

- (1) 起步                      下载;配置
- (2) 全局css样式            {栅格系统}
- (3) 组件
- (4) js插件
- (5) 定制                      {less动态样式语言}

## Bootstrap 起步

---

### 创建基本的模板

```
<html lang="zh-CN">
language : 指定当前网页所用自然语言
作用两个：为浏览器翻译功能确定基础发音，
           为读屏软件确定基础发音

<meta http-equiv="X-UA-Compatible" content="IE=edge">
指定 IE 浏览器启动最新内核渲染网页
网景--浏览器 100%-->FireFox
IE5.5 绑定 windows 98 100%
IE6 ->内核5.5;6
IE7 ->5.5;6;7
IE8 ->5.5;6;7;8
IE9 ->5.5;6;7;8;9
IE11 ->5.5;6;7;8;9;10;11 17%
chrome 44% 360安全

<meta name="viewport" content="width=device-width, initial-scale=1">
```

### Bootstrap屏幕的分类

大型PC屏幕	lg w>=1200px
中等PC屏幕	md 1200px>w>=992px
小型PAD屏幕	sm 992px>w>=768px
超小PHONE屏幕	xs 768px>w

## Bootstrap 全局样式 —— 按钮

.btn	按钮基础样式
.btn-default	白底黑字的按钮
.btn-danger .btn-warning .btn-success .btn-info .btn-primary	五种颜色
btn-lg;.btn-sm;.btn-xs	按钮四种大小

## Bootstrap 全局样式 —— 图片

.img-rounded	圆角图片
.img-circle	圆形图片
.img-thumbnail	缩略图()
.img-responsive	响应式图片 ( 大小变化 , 最大不会超过自身大小 )

## Bootstrap 全局样式 —— 文本

.text-danger; warning; success; info; primary
.bg-danger; warning; success; info; primary
.text-left; .text-right; .text-center;.text-justify
.text-uppercase; lowercase; capitalize

## Bootstrap 全局样式 —— 列表

---

.list-unstyled

.list-inline

## Bootstrap 全局样式 —— 表格

---

.table	表格基础样式
.table-bordered	带边框表格
.table-striped	隔行变色
.table-hover	带悬变色
.table-responsive	响应式表格（保存table父元素）

## Bootstrap 全局样式 —— 栅格(重点)

---

网页布局方式:

- (1) TABLE 布局
- 好处:简单易控制
- 不足:网页内容丰富太复杂,渲染效率
- (2) DIV+CSS 布局
- 好处:页面渲染快
- 不足:控制有不少难度(兼容)

### (3) Bootstrap 栅格布局

好处:渲染快,支持响应式

缺点:不适合开发复杂网页

## 栅格 —— 使用方法

---

### (1) 最外层必须使用容器

`div.container` 或 `div.container-fluid`

### (2) 容器里可以放置任何内容,若想使用栅格,必须 `div.row`

`.container > .row`

### (3) 一个`.row`中不能放置其它内容,只能放`col`,列中可以放置任何内容

`.container > .row > .col`

### (4) Bootstrap中的行默认均分12等分,每个列必须指在行中占比

### (5) 栅格系统针对不同屏幕提供不同列

`.col-lg-1/2/3/4/.../12`

`.col-md-1/2/.../12`

`.col-sm-1/2/3/...4`

`.col-xs-1/2/3/.../12`

### (6) 可以使用"列偏移",实现指定列及其后的列向右偏移

`.col-lg-offset-1/...12`

`.col-md-offset-1/...12`

`.col-sm-offset-1/...12`

`.col-xs-offset-1/...12`

### (7) 需要注意不同屏幕下适用性问题

`.col-xs-*` 适用于xs/sm/md/lg 屏幕

`.col-sm-*` 适用于 sm/md/lg 屏幕

`.col-md-*` 适用于 md/lg 屏幕

`.col-lg-*` 适用于 lg 屏幕

### (8) 一个列可以指定在不同屏幕不同宽度占比

`div.col-xs-12 .col-sm-9 .col-md-6`

`div.col-xs-12 .col-md-6`

div.col-xs-12 .col-sm-6 .col-md-6

(9) 一个列可以在指定屏幕下隐藏

.hidden-xs 仅在xs屏幕

.hidden-sm 仅在sm屏幕

.hidden-md 仅在md屏幕

.hidden-lg 仅在lg屏幕

(10) 栅格可以嵌套 规则:在列中再嵌入行,行中再有列

.container >

.row >

.col >

.row >

.col >

## Bootstrap 全局样式 —— 表单

### (1) 默认表单

```
1 <form>
2   <div class="form-group">
3     <label class="control-label">提示文字</label>
4     <input type="" class="form-control"/>
5     <span class="help-block">说明文字</span>
6   </div>
7 </form>
```

### (2) 行内表单

```
1 <form class="form-inline">
2   <div class="form-group">
3     <label class="sr-only">搜索内容</label>
4     <input class="form-control"/>
5   </div>
6 </form>
```

`sr-only` : 仅读屏软件可以识别

### (3) 水平表单

水平表单=表单+栅格系统(变型)

	栅格	水平表单中栅格
最外元素	div.container	form.form-horizontal
行	div.row	div.from-group
列	div.col	div.col--

```
1 <form class="form-horizontal">
2   <div class="form-group">
3     <div class="col-*-*">
4       <label></labe>
5     </div>
6   </div>
7 </form>
```

# Bootstrap 组件 —— 下拉菜单

下拉菜单必需三级结构

```
1 <div class="dropdown">
2   <a data-toggle="dropdown">触发元素</a>
3   <ul/div class="dropdown-menu">隐藏元素</div>
4 </div>
```

# bootstrap 组件 —— 导航

提示：Bootstrap 中导航(nav) 不同于导航条(navbar)

## (1) 标签页导航 -- 页签组件

```
1 <ul class="nav nab-tabs">
2   <li><a data-toggle="tab">十元套餐</a></li>
3   <li class="active"><a data-toggle="tab">二十元套餐</a></li>
4   <li><a data-toggle="tab">三十元套餐</a></li>
5 </ul>
```

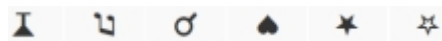
## (2) 胶囊式导航



```
1 <ul class="nav nav-pills">
2   <li><a data-toggle="tab">十元套餐</a></li>
3   <li class="active"><a data-toggle="tab">二十元套餐</a></li>
4   <li><a data-toggle="tab">三十元套餐</a></li>
5 </ul>
```

## bootstrap 组件 —— 图标字体

---



web 项目常用图标字体

- (1) FontAesome - 675个免费图标
- (2) Glyphicons - 800个收费图标(200个免费)

服务器端字体使用方法:

- (1) web服务器的项目目录下必需有字体
- (2) css声明一个新字体，并指定该字体文件下载位置

```
@font-face {
    font-family: 'Glyphicons Halflings';
    src: url('../fonts/glyphicons-halflings-regular.eot');
}
```

- (3) 声明一个基础class,使用到了该字体

```
.glyphicon{
    font-family: 'Glyphicons Halflings';
}
```

- (4) 在html页面中输入对应图标 unicode码

```
.glyphicon-plus:before {
    content: "\002b";
}
```

对于页面开发者:

```
<span class="glyphicon glyphicon-plus "></span>
```

## bootstrap 组件 —— 警告框

---

```
1 <div class="alert alert-四种颜色">
2   <span class="close" data-dismiss="alert">X</span>
3   <p>...</p>
4 </div>
```

## bootstrap 组件 —— 进度条

---

.progress > .progress-bar

## bootstrap 组件 —— 小组件

---

面包屑 .bread-crumb

分页条 .pagination

分页器 .pager

徽章 .badge

巨幕 .jumbotron

水井 .well

## bootstrap 组件 —— 缩略图

---

```
1 <div class="thumbnail">
2   <img src="">
3   <div class="caption"></div>
4 </div>
```

注意：缩略图本身是block组件，需要配合栅格系统控制宽度

## bootstrap 组件 —— 媒体对象

---

以"左中/中右/左中右" 横向三部分来展示一个商品/评论

```
1 <div class="media">
2   <div class="media-left">img/</div>
3   <div class="media-body"></div>
4   <div class="media-right"></div>
5 </div>
```

## bootstrap 组件 —— 面板--(手风琴)

以“中/上中/上中下” 纵向三部分来展示数据

```
1 <div class="panel panel-颜色">
2   <div class="panel-heading"></div>
3   <div class="panel-body"></div>
4   <div class="panel-footer"></div>
5 </div>
```

## bootstrap 插件--jQuery插件--折叠(collapse)

Bootstrap 基于 jQuery 提供十几个插件函数

调用方法两种

(1) js编码方式

```
$(".dropdown > a").dropdown();
```

(2) data扩展方式

```
<a data-toggle="dropdown"></a>
```

```
1 <a href="#my-target" data-toggle="collapse">展开/收起</a>
2 <button data-toggle="collapse" data-target="#my-target">展开/收起</button>
3 <div id="my-target" class="collapse">..</div>
```

## bootstrap插件--jQuery插件--导航条--(最复杂)



## Bootstrap--定制--(less重点)

---

Bootstrap--定制--分析样式语言分类

(1) 静态样式语言：

CSS:可以直接被浏览器所解析,但作为一门语言，CSS  
并不完整，缺少编程语言必需:变量;函数;数据类型  
导致样式可维护性差.

(2) 动态样式语言/样式预处理语言：

有三种: Sass/Stylus/Less 在CSS的基础之上添加了动态  
编程言必须特性,如：数据类型,变量,运算符,函数...  
提高样式的可维护性.  
需要注意:浏览器只能识别CSS,所有动态样式语言的代码必  
需转换CSS,该过程称为"编译(Complie)"

## Bootstrap--定制--less 样式语言

---

<http://lesscss.cn>

less是一门CSS预处理语言--它扩展了CSS语言，并增加变量  
函数等特性，使用CSS更易于维护.

less两种用法

(1) 在客户端使用less---(学习)

编写一个x.less

在HTML中引入x.less文件，同时引入less编译程序 less.js  
客户端请求html/下载 less.js x.less 在客户端用less.js编译  
x.less编译css，会减慢客户端样式呈现速度

## (2) 在服务器端使用less

程序员编写 x.less  
在开发电脑上安装less编译环境，执行它把x.less编译x.css  
再编写html文件,引入x.css  
客户请求HTML/CSS

## Bootstrap--定制--搭建less编译环境

---

- (1) 下载并安装独立js解释器---Node.JS  
在命令行执行 `node -v`
- (2) 下载less编译程序 -- lessc.js  
在命令行执行 `lessc`
- (3) 使用less编译程序lessc将e:/01.less文件编译 e:/01.css  
`lessc e:/01.less e:/01.css`

## Bootstrap--定制--语法学习

---

- (1) Less支持所有的css语法
- (2) Less支持多行注释/多行注释,但只多行注释会编译到  
css文件中--推荐单行注释
- (3) Less支持"变量(Variable)"的概念  
定义:@变量名:值;  
使用:color:@变量名;  
变量可以取值为任何合法的样式值
- (4) Less支持变量和常量的运算  
`+ - * / %`
- (5) Less 支持在一个选择器中"混入Mixin" 另一个选择器  
定义样式  
选择器1{....}

选择器2{...选择器1...}

#### (6) Less在样式混入时可以指定参数

选择器1(@参数1,@参数2){}

选择器2{...选择器1(值1,值2)}

#### (7) less支持样式嵌套

```
选择器1{  
    选择器2{...}  
}
```

上述代码会被编译

```
选择器1{..  
    选择器1 选择器2{...}
```

#### (8) less支持提供几十个样式函数操作

```
ceil();  
floor();  
percentage(num);    把小数转换为百分比形式  
darken(color,percentage) 把指定颜色变暗  
lighten(color,percentage) 把指定颜色变亮  
image-width(url);  
image-height(url);  
...
```

#### (9) less支持文件包含

less中可以使用 @import实现包含文件

可以把一个大项目中所需CSS样式分开保存在不同文件中,  
有利于实现分工协作.

```
button.less  
text.less  
boot.less [包含button.less;text.less]
```

## Bootstrap--定制

---

Bootstrap的定制主要为了达到三个目标

#### (1) 瘦身:删除不需的样式

只需要注释掉bootstrap.less不需要 @import

lessc bootstrap.less bootstrap.css

## (2) 粗粒度定制

只需要修改 variables.less中定义变量值即可

lessc bootstrap.less bootstrap.css

## (3) 细粒度定制

修改组件对应.less , 如:dropdown.less

# Bootstrap--插件(js)

---

## (1) 图片轮播

```
1 <div class="carousel" data-ride="carousel">
2   <div class="carousel-inner"> 滚动
3     <div class="item active">img+.carousel-caption</div>
4   </div>
5 </div>
```