

目录

目录.....	1
HTML 部分.....	4
webstrom 的常用配置及快捷键.....	4
什么是 HTML.....	5
DTD 文档声明.....	6
基本标签.....	8
行内元素与块元素.....	15
请求响应模型.....	19
Frameset 框架集.....	20
表格布局.....	24
页面中常见的图像格式.....	27
HTML 实体字符.....	27
表单.....	28
HTML5 中的 input 类型:	33
post 方式与 get 方式的区别:	34
服务器搭建.....	34
HTML5 新增语义化标签.....	36
CSS 部分.....	38
CSS 的基础选择器.....	39
CSS 基础样式.....	40

CSS 的其它选择器.....	44
浮动.....	50
定位.....	51
display 属性.....	53
CSS 优先级计算.....	55
盒子模型.....	56
怪异盒子模型.....	59
伸缩盒子模型.....	60
浏览器内核以及其前缀.....	62
清除浮动方式总结.....	63
CSS 继承.....	64
CSS3 基础样式.....	65
过度 Transition.....	67
Transform 转换.....	71
动画 animation.....	77
CSS 特殊图形.....	79
reset 文件.....	88
有衬线和无衬线字体的区别.....	88
PS 快捷键.....	89
阿里字体图标.....	91
CSS Sprite（雪碧图）.....	92
雪碧图制作工具.....	93
滑动门.....	94
多列.....	105
resize.....	106

Box-sizing.....	107
META 标签详解.....	108
媒体查询.....	111
Bootstrap 使用.....	113
Bootstrap 栅格系统.....	118
Bootstrap（一）全局样式.....	120
Bootstrap（二）组件.....	129
实例代码.....	143
常见分辨率.....	151

HTML 部分

webstrom 的常用配置及快捷键

1.如何更改主题（字体&配色）

file->setting->editor->colors&fonts->scheme 选择你的主题（darcula）

2.如何让 webstorm 启动的时候不打开工程文件

file->settings->Appearance&Behavior->System settings 去掉 reopen last project on startup（在启动的时候总是打开最后的工程）

3.如何完美显示中文

file->settings->appearance 中勾选 override default fonts by(not recommended)，设置 name:NsimSum,Size:12

(appearance 外观选项，选中“重写默认字体”，设置名字为 NsimSum，它是 windows 下的宋体字，字号 12)。这里设置的是整个 IDE 显示的文字，包括菜单列表的字。你可以试着把它改成英文，那样再在菜单项中看，中文有的会不正常显示，变成小方块了。

4.如何显示行号

在代码显示区的左边右击选择“show line number”。

5.如何代码自动换行

file-settings-editor->general-> "use soft wraps in editor" 打上钩，代码就自动换行了。

你先试下，在一行中写很长的句子，看会不会换行。

webstrom 快捷键

tab 键 补全代码

ctrl+/ 快速注释一行

ctrl+y 快速删除一行

ctrl+d 快速复制一行

ctrl+w 快速选中，增量（按一下选中一行，再按选中一段，依次向外选择更多的内容）

ctrl+f4 关闭当前文件

ctrl+f 在当前文件快速查找

ctrl+r 查找并替换

ctrl+shift+f 在指定路径查找

ctrl+shift+r 在指定路径查找并替换

什么是 HTML

它是“超文本标记语言”，英文全称是 HyperText Markup Language，简称 HTML。HTML 文档是由标签组成的描述性文本，HTML 标签可以描述文字、图形、动画、声音、表格、链接等。HTML 的结构包括头部(Head)、主体(Body)两大部分。头部描述浏览器所需的信息，主体包含所要说明的具体内容。

html 分为 闭合标签（双闭合） 自闭合标签（单闭合）

html 文档基本结构

```
<!DOCTYPE html>    //告诉浏览器使用了哪个 html 版本
<html>              //html 文档的开始
<head lang="en">      //head 中多定义了搜索引擎和浏览器会读取的内容 zh-cn
    <meta charset="UTF-8">    //字符编码
    <title></title>        //文档标题 收藏夹
</head>
<body>              //body 中的内容是用户可以看得到的，我们的主“战场”
    ...
</body>
</html>
```

DTD 文档声明

DTD 文档模型也称 DOCTYPE 文档声明，它是 Document Type Definition 的英文缩写，分为两种，html5 版本的 doctype 和 html4 版本的 doctype，其中

html4 版本的分为三类

HTML 5

```
<!DOCTYPE html>
```

HTML4

三种类型的声明：严格 strict 过渡 Transitional 框架 frameset

HTML 4.01 Strict 严格

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

该 DTD 包含所有 HTML 元素和属性，但不包括展示性的和弃用的元素（比如 font）。不允许框架集（Framesets）

HTML 4.01 Transitional 过渡

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

该 DTD 包含所有 HTML 元素和属性，包括展示性的和弃用的元素（比如 font）。不允许框架集。现在在使用的 90%是它

HTML 4.01 Frameset 框架

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
```

该 DTD 等同于 HTML 4.01 Transitional，但允许框架集内容。

基本标签

html head body

head 包含

<title>、<meta>、<link>、<script>、<style>

常见的 meta 标签

Keywords(关键字) keywords 用来告诉搜索引擎你网页的关键字是什么。

<meta name="keywords" content="北京 JAVA 培训,北京 java 培训机构,java 视频下载,java 视频,马士兵,马士兵 java 视频下载">

description(网站内容描述) description 用来告诉搜索引擎你的网站主要内容。

<meta name="description" content="北京尚学堂-cctv 央视网广告合作伙伴, 专业 IT 培训机构,口碑最好的 java 培训、,iOS 培训, android 培训, 大数据培训, web 前端, 0 元入学, 先就业后付款">

<meta http-equiv="Content-Type" content="text/html; charset=utf-8">

在 html5 中被精简为了<meta charset=" utf-8" />

body 包含

<body>标签

属性 background、bgcolor

<h1>~<h6>标题

从 h1 到 h6 权重依次下降，h1 最高，大小依次不同。 <h1></h6>这是错误写法，也没有<h7>标签

h 标签拥有 align 属性(块元素都可使用 align 属性)，它有如下几个常用值

left

center

right

justify 两端对齐（用 firebug 可看到和 left 的区别，并且，是需要多行文本才可以看出效果）



示例：使用 align 属性让标题居中

```
<h1 align="center">标题</h1>
```

<p>标签

标签定义段落。元素会自动在其前后创建一些空白。浏览器会自动添加这些空间，您也可以在样式表中规定。你在编辑器中敲的空格，浏览器是会忽略的。

标签 它用来组合文档中的行内元素，以便通过样式来格式化它们。如果不对 `span` 应用样式，那么 `span` 元素中的文本与其他文本不会任何视觉上的差异。

<div>标签 可定义文档中的分区或节，标签可以把文档分割为独立的、不同的部分。它可以作为元素的组织工具

标签 可插入一个简单的换行符。在刚才的例子中完全可以使用它来换行。它是自闭合标签（意味着它没有结束标签，因此这是错误的：
</br>）。应该把结束标签放在开始标签中，也就是
。

<hr>标签 在 HTML 页面中创建一条水平线。可以在视觉上将文档分隔成各个部分。

<!-- -->标签 注释标签用于在源代码中插入注释，注释不会显示在浏览器中。这样做有助于在以后的时间对代码的编辑。当编写了大量代码时尤其有用。

格式类的基本标签

<address> 标签

标签定义文档或文章的作者联系信息。定义在 `body` 内则表示文档联系信息。这个标签是带有语义的，也就是说我们看到 `a` 标签，并不能从直观意义上知道它是一个超链接，但是 `address`，通过翻译，也能明白这是定义联系信息，这就是语义化的好处。它以斜体显示。

 标签

呈现粗体文本效果。它属于字体样式元素，并不反对使用这些标签，但是如果只是希望通过这些标签改变文本的样式，建议使用样式表。

 标签

定义被删除文本，用它包含的内容会显示一道中划线。

 标签

用它包裹的内容是告诉浏览器要强调的内容。用 **em** 包裹的这段文字用斜体来显示。如淘宝会用来强调价格、成效量等。



<i> 标签

标签显示斜体文本效果。它的显示和 **em** 显示的一样，不过它对于浏览器来说没有强调的意思

<pre> 预格式化文本标签

被包围在 **pre** 元素中的文本通常会保留空格和换行符。

<small> 标签

标签呈现小号字体效果。它将字体缩小，如果被包裹的字已经是最小了，就不会有任何效果。它和 **big** 标签对应。

<big> 标签

big 标签呈现大号字体效果。它和 **small** 都是有语义化的标签

 标签

和 `` 标签一样，用于强调文本，但它强调的程度更强一些。`strong` > `em`，一个是斜体显示，一个正常加粗显示。


`<u>`标签 为文本添加下划线

`<sup>`标签

定义上标文本。 $(a-b)^2 = a^2 - 2ab + b^2$

`<sub>`标签

定义下标文本。

``标签  向网页中嵌入一幅图像。`` 标签有两个必需的属性：`src` 属性 和 `alt` 属性。

`alt` 属性：如果无法显示图像，浏览器将显示替代文本

`src` 属性：它的值是图像文件的 URL，也就是引用该图像的文件的绝对路径或相对路径。

`title` 属性：鼠标放上时显示的文字

`width` 属性：设置图像的宽

`height` 属性：设置图像的高

`border` 属性：定义边

`align` 属性：它有五个可选值 `top` `bottom` `middle` `left` `right` 它设置图像在文本中的一个位置。`top` 是指以图像的头部去和文本对齐，`bottom` 是底部也是默认，`middle` 是中间显示，`left` 和 `right` 分别是在文本的左右位置。



<a> 标签

a 标签定义超链接，用于从一张页面链接到另一张页面。元素最重要的属性是 href 属性，它指示链接的 URL 地址。有时我们也会通过 a 标签来定义一个锚点，它的作用是在文档中设置位置标记，并给该标记一个名称，以便引用。

实例

请点击	通过#和锚点名字找到 aName 这个名字的标签所在位置
这是页面底部	通过 name 属性来定义锚点名字
点击返回顶部	也可以直接通过#来返回页面的顶部，#代表了页面的顶部

target 属性 规定在何处打开链接文档。有以下四个值

_blank 在新标签中打开

_parent 在父标签中打开，如果本身已经是顶级，那么就是在自身窗口打开

_self 默认就是在自己窗口打开

_top 清除所有框架，载入到浏览器窗口

列表相关标签

无序列表

无序列表是一个项目的列表，此列项目使用粗体圆点（典型的小黑圆圈）进行标记。无序列表始于 `` 标签。每个列表项始于 ``。

```
<ul>
  <li>Coffee</li>
  <li>Milk</li>
```

```
</ul>
```

改变无序列表的项目符号通过 `type` 属性

`type="disc"` 默认黑色圆点

`type="circle"` 空心圆点

`type="square"` 方块

有序列表

列表项目使用数字进行标记，有序列表始于 `` 标签。每个列表项始于 `` 标签。和无序列表不同的是有序列表有顺序。

```
<ol>
  <li>Coffee</li>
  <li>Milk</li>
```

```
</ol>
```

改变有序列表的项目符号也是通过 `type` 属性，如果页面中使用时，尽量不要用它的数字，兼容有问题

数字列表 默认即是数字列表

字母列表 `type="A"` `type="a"`

罗马字母列表 type="I" type="i"

定义列表

自定义列表不仅仅是一列项目，而是项目及其注释的组合。自定义列表以 `<dl>` 标签开始。每个自定义列表项以 `<dt>` 开始。每个自定义列表项的定义以 `<dd>` 开始。定义列表前是没有项目符号和类型的，所以给它指定 `type` 是不起作用的。

```
<dl>
  <dt>支付方式</dt>
  <dd>信用卡</dd>
  <dd>支付宝</dd>
  <dd>银行卡</dd>
</dl>
```

行内元素与块元素

行内元素也叫内联元素，块元素有的也叫行元素。

行级元素和块级元素

元素是通过 `display` 的属性值来划分的：`block` 块元素 `inline` 行内元素 `inline-block` 行内块元素



块级元素具有以下特点：

- ①总是在新行上开始，占据一整行；
- ②高度，行高以及外边距和内边距都可控制；
- ③宽在未设置情况下与浏览器宽度一样，与内容无关；
- ④它可以容纳内联元素和其他块元素。

行内元素的特点：

- ①和其他元素都在一行上；
- ②内联元素不可以设置宽和高、垂直的 margin、padding!!
- ③宽度只与内容有关；
- ④行内元素只能容纳文本或者其他行内元素。

	独占一行	能设置宽高	margin,padding值
块级元素	是	是	四个方向都可以设置
内联元素	否	否	竖直的不可设置

常见内联元素和块元素

常见的有

内联元素(inline element)

* a - 锚点

* b - 粗体(不推荐)

* big - 大字体

* em - 强调

* font - 字体设定(不推荐)

* i - 斜体

* img - 图片(行内元素，但可设置宽高)

* label - 提示标签

* select - 列表选择

* small - 小字体文本

* span - 常用内联容器，定义文本内区块

* strong - 粗体强调

* sub - 下标

* sup - 上标

* textarea - 多行文本输入框

* u - 下划线

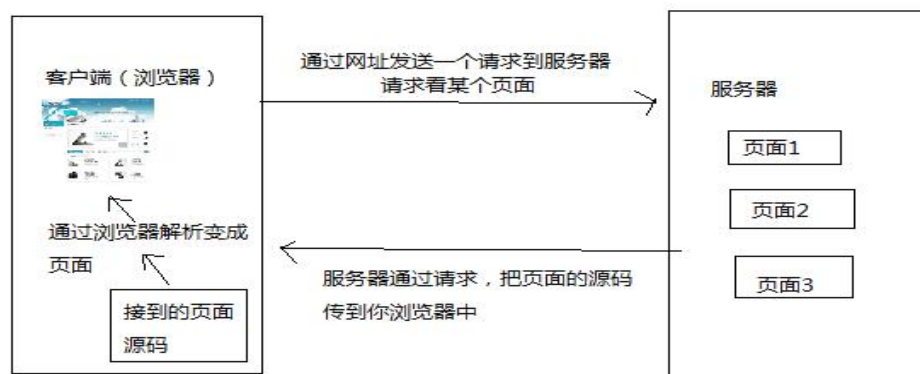
块元素

块元素(block element)

- * address - 地址
- * div - 常用块容器
- * dl - 定义列表
- * fieldset - 元素集
- * form - 交互表单
- * h1 - h6 标题
- * hr - 水平分隔线
- * ol - 有序列表
- * p - 段落
- * pre - 格式化文本
- * table - 表格
- * ul - 无序列表

input 是行内块元素

请求响应模型



「HTML 文档&浏览器解析器&DTD 文档的关系」

HTML 文档: 是一堆代码, 用户无法直接看

浏览器解析: 把代码解释转换成图片、文字用户可以看的内容

DTD 文档: 你 HTML 文档的规范, 规定了使用哪种版本的 HTML

页面布局

常见三种方式：

Frameset 布局

Table 布局

Div 布局

html 文档流

HTML 中的文档流是把元素按从上而下，从左到右的顺序默认排列。不在一行的元素从上而下（块元素），在一行的从左到右排列（行内元素）。元素可以脱离文档流显示。

Frameset 框架集

和框架集相关的三个标签 frameset 标签、frame 标签、iframe 标签。使用 frameset 时不可以和 body 同时使用，因为它们是同级的。一个 frame 引导了一个 html 文档。一个框架集可包含多个 html 文档。

一个基本的框架集示例

```
<frameset rows="200px,*">  
  <frame src="http://www.baidu.com" />  
  <frameset cols="30%,*">  
    <frame src="http://www.baidu.com" />  
    <frame src="http://www.baidu.com" />  
  </frameset>  
</frameset>
```

「Frameset 的一些属性」

cols="" 把一个浏览器窗口按列进行切割，也就是把浏览器竖着切分

cols="20%,200,*" 左边部分占 20%，中间部分占 200px，剩下的为第三部分。

rows="20%,200,*" 以行方式分割，横着划分

border="1" 通过值来设置

bordercolor="red" 框架边的颜色

「Frame 的常用属性」

①name 设置框架名称。

②src 设置此框架要显示的网页名称或路径。此为必须设置的属性。

③scrolling="" 设置是否要显示滚动条。设定值为 yes, no。

④noresize="noresize" noresize="noresize" 设置了 noresize 的 frame 会影响其它两旁的 frame, 如果当前 frame 在最边上, 则只影响它本身。chrome 可设置在 frameset 中, 这样全局都不可拖动, 而火狐要设置在具体的 frame 上。

如果不支持 frameset 框架集的话, 要使用 noframes 标签来替代

```
<noframes>
```

```
    <body>您的浏览器无法处理框架! </body>
```

```
</noframes>
```

iframe 内联框架

可以包含在 body 中, 和普通的元素一样

相关属性

①width 可设置内联框架的宽

②height 可设置内联框架的高

③name 设置框架名称

④src 设置页面的路径

⑤scrolling="" yes/no 滚动条

框架集的示例

index.html

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<frameset rows="200px,*">
  <frame src="http://www.baidu.com"/>
  <frameset cols="30%,*">
    <frame src="list.html"/>
    <frame src="http://www.baidu.com" name="TestName" />
  </frameset>
</frameset>
</html>
```

list.html

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
```

```
<a href="http://www.360.com" target="TestName">百度</a>
</body>
</html>
```

关于框架集的 a 标签的 target 属性值

_blank，在新的窗口中打开被链接文档

_self，默认，在相同的框架中打开被链接文档

_parent，在父框架集中打开被链接文档（在最外面的 frameset 中打开）

_top，在整个窗口中打开

frameName，通过名字在指定的框架中打开（最常用）

表格布局

table > tr > td/th

table 相关的属性

属性

边 border，定义 border 边框

跨行 rowspan= “2”

跨列 colspan= “3”

宽 width

高 height

排列属性 align ， 可居中显示表格

基本表格代码示例

```
<table border="1">
  <caption>这是表格的标题</caption>
  <tr><th colspan="5">课程表</th></tr>    //th 是加粗居中的 td，代表了表头
  <tr>
    <td>星期一</td>
    <td>星期二</td>
    <td>星期三</td>
    <td>星期四</td>
    <td rowspan="2">星期五</td>
  </tr>
  <tr>
    <td>上课</td>
    <td>上课</td>
    <td>上课</td>
    <td>上课</td>
  </tr>
</table>
```

thead tbody tfoot

对于结构复杂的表格，可以将表格分割成三个部分：题头、正文和脚注。而这三部分分别用: `thead`, `tbody`, `tfoot` 来标注。

`thead` 表格的头 用来放标题之类的东西

`tbody` 表格的身体 放数据本体

`tfoot` 表格的脚 放表格的脚注之类

这三个标签的作用是：为了让表格(`table`)在下载的时候可以分段的显示,就是说在浏览器解析 HTML 时，`table` 是作为一个整体解释的，而使用了 `tbody` 标签后，`tbody` 标签中的内容可以优先显示。

它与 `th`、`tr`、`td` 有明显的区别，使用 `thead`、`tbody`、`tfoot` 是为了控制**显示顺序**，表格本来是从上向下显示的，但是应用了 `thead`/`tbody`/`tfoot` 以后，无论它们前后顺序如何改变，`<thead>` 内的元素总是在表的最上面，`<tfoot>` 总在表的最下面，`thead` 元素应该与 `tbody` 和 `tfoot` 元素结合起来使用。但如果只有一个 `tbody` 的时候经常不写 `tbody`，我们平时不写时，浏览器都默认认为内容放在了 `tbody` 中。

用 html 的方式改变 `table` 的边，变为单线

`cellspacing="0" cellpadding="0"`

页面中常见的图像格式

图片分为矢量图与位图

矢量图是放大缩小不会失真的图形，如 `svg` 画出的图

位图，又叫像素图，位图好比是一幅大的拼图，每一块就是一个像素点。我们在 `web` 页面中所使用的 `JPG`、`PNG`、`GIF` 格式的图像都是位图。

总结：

由于图片的压缩形式不同，不同格式的图片的大小也不同，`jpg`、`bmp` 格式的图片要比 `png` 格式的体积大。

1.web 优先使用 `png`

2.照片、色彩要求高的图片 `jpg`

3.web 中的小动画 `gif`

4.UI 设计图一般是 `PSD`

*网页中不要大范围的使用文件较大的 `bmp`、`jpg` 格式文件

HTML 实体字符

在 `html` 开发中，一些字符不适于直接写出，因为它有可能引起标签混乱，如 `<` 和标签的开始标记冲突，这时候我们可以使用“实体字符”来替换。

实体字符是对大小写是敏感的。



实体字符一般格式 &+ 实体 +;

常用实体字符

大于> 小于< 引号" >>» <<« 空格 版权©

实体字符对照表

<http://tool.xker.com/htmlchar.php>

表单

表单的作用

1. 表单用于搜集不同类型的用户输入（搜集用户信息）
2. 用户提交表单时向服务器传输数据，从而实现用户与 Web 服务器的交互
3. 表单是一个包含表单元素的区域
4. 表单使用表单标签（<form>）定义

form 表单的常见属性

action= “URL”：一个 URL 地址；指定 form 表单向何处发送数据。

它里面放一个路径 ,我们在代码编辑器中写 form 时总会自动带着它, `<form action="" />` 什么意思呢？我们表单提交到服务器需要一段程序来处理，那由哪个程序来处理就可以在 action 中指定，如果我是注册，可以由注册的程序来处理，登录由登录的程序处理。

name

作用是给表单起名，为了便于我们操作

`<form name="regFrm"/>` 通常我们给表单起名都遵循这么个规则，前面的 reg 是这个表单是干什么的（reg 注册）后面以 Frm 结尾，就是 form 的一个缩写

enctype = “string”：规定在发送表单数据之前，如何对表单数据进行编码。通常在使用文件上传时，我们会写 `enctype="multipart/form-data"`，以二进制传输

method = “get/post”：指定表单以何种方式发送到指定的服务器程序。

该属性定义浏览器将表单中的数据提交给服务器处理程序的方式。关于 method 的取值，最常用的是 get 和 post。

一个完整的表单包含三个基本组成部分：

表单标签、表单域、表单按钮。

1. 表单标签

是指 `<form>` 标签本身，它是一个包含表单元素的区域，使用 `<form></form>` 定义

2. 表单域

是<form>标签中用来收集用户输入的每一项，通常用 input 标签来定义，input 标签有不同的类型，对应用户不同的数据。

3. 表单按钮

用来提交<form>表单中的所有信息到服务器

表单元素

单行文本框<input type="text">默认值是 type="text"

密码框<input type="password"/>

单选按钮<input type="radio"/>

复选框<input type="checkbox"/>

隐藏域<input type="hidden"/>

文件上传<input type="file"/>

```
<form action="" method="post" enctype="multipart/form-data">
    <input type="file" name="file1"/>
    <input type="submit" />
</form>
```

下拉框<select>标签

```
<select>
```

```
    <option value="1">北京</option>
```

```
    <option value="2">天津</option>
```

```
</select>
```

disabled="disabled" 规定禁用该下拉列表



使用 optgroup 为 option 分组，通过 label 指定分组的名称

```
<select name="" id="">
```

```
    <optgroup label="中国">
```

```
        <option value="tj">天津</option>
```

```
        <option value="2">北京</option>
```

```
        <option value="3">上海</option>
```

```
    </optgroup>
```

```
    <optgroup label="食物">
```

```
        <option value="4">北京烤鸭</option>
```

```
        <option value="5">北京烤鸭</option>
```

```
        <option value="6">北京烤鸭</option>
```

```
    </optgroup>
```

```
</select>
```

多行文本<textarea></textarea>

多行文本是可以拖动改变大小的，如果不希望你的布局被改变，可以使用 css 来控制 style="resize: none;"

placeholder 规定描述文本区域预期值的简短提示

readonly 规定文本区为只读

required 规定文本区域是必填的，不能和 readonly 一起用

标签<label></label>

```
<label for="MyId"></label>
```

```
<input type="radio" id="MyId" />
```

元素集

```
<fieldset>
```

```
<legend>健康信息</legend> //通过 legend 指定元素集的标题
```

```
身高: <input type="text" />
```

```
体重: <input type="text" />
```

```
</fieldset>
```

提交按钮<input type="submit"/>

普通按钮<input type="button"/>

重置按钮<input type="reset"/>

图片按钮<input type="image"> 也有提交作用，容易提交两次

默认选中

单选、复选 `checked="checked"`

下拉列表 `selected="selected"`

HTML5 中的 input 类型:

1.email

2.url

3.number

4.range

`range` 类型用于应该包含一定范围内数字值的输入域。`range` 类型显示为滑动条。规定了范围（最大最小值），还可以设置 `value`。

属性 `min` 最小值 `max` 最大值 `value` 当前值，最大最小值规定了一个范围

5.Date

6.search

7.color

post 方式与 get 方式的区别:

一、安全性

get 方式不够安全，以 url 方式进行提交

post 以请求实体提交，不会显示地址栏，足够安全

二、提交大小限制

get 方式提交的内容大小有限，一个地址栏放不了多少东西，1k 左右

post 大小无限制，可以放大文件（音频、视频、图像。。。）

服务器搭建

XAMPP 下载及安装

1. 下载地址: https://www.apachefriends.org/zh_cn/index.html
2. 找到 XAMPP for Windows 点击下载
3. 双击下载好的安装包进行安装
4. 最好不要安装在 C 盘，因为后期我们会在它的安装目录中放我们的页面，这样会占 C 盘的内存
5. 出现黑色的 DOC 窗口，把它关闭，点击 next 继续安装

怎么使用：

- 1) 打开 xampp
- 2) 在安装路径下找到 htdocs 文件夹
- 3) 把你写的带有 form 的 html 文件拷贝到 htdocs 下
- 4) 访问 localhost (本地地址) / 文件名
- 5) 把 localhost 换成自己的 IP，其它人也可以访问了
192.168.0.101/综合实例.html

查看自己电脑 IP 地址的方法

在电脑的开始按钮中找到“运行”，或是按快捷快捷 win+r

输入 cmd 命令

在弹出的 DOC 窗口输入 ipconfig，注意，中间没有空格

在其中找到 IPv4 地址，这是你电脑的 IP

访问的时候即可访问 <http://192.168.0.102/index.html>

自己访问也可以访问 localhost/index.html



HTML5 新增语义化标签

(一)

- 1.header 页面头部
- 2.footer 页脚
- 3.article 定义页面独立的内容区域一般为文章
- 4.aside 定义页面的侧边栏内容
- 5.details 文档某个部分的细节
- 6.summary 是 details 中的标题
- 7.time 定义日期或时间
- 8.ruby 标注拼音
- 9.mark 标记
- 10.nav 导航链接

(二)

- 1.progress 进度标签
 - value 当前值 max 最大值 最小值默认为 0，不用设置
- 2.section 文档当中的节或文章
- 3.video 定义视频

```
<video controls>  
  <source src="/i/horse.ogg" type="video/ogg">
```

```
<source src="/i/horse.mp4" type="video/mp4">
```

你的浏览器不支持 audio 标签，即使是上面的视频都不能播也会显示个框，不支持是指浏览器不支持 video 标签

```
</video>
```

目前支持的视频类型，其它格式如 avi 需要转换

video/ogg

video/mp4

video/webm （webm 也是一种媒体格式）

4.audio 定义音频

定义方式同 video

音频的可用类型：

audio/ogg

audio/mpeg

5.source 资源

6.datalist 提示可能的值

```
<input type="text" list="MyId"/>
```

```
<datalist id="MyId">
```

```
  <option value="苹果"></option>
```

```
  <option value="橙子"></option>
```

```
  <option value="梨"></option>
```

```
</datalist>
```

7.embed 引入 flash 或插件

定义嵌入的内容，比如插件、flash。它中间不要加内容，和 video 不同，会显示出来。

```
<embed src="demo.swf" type="application/x-shockwave-flash" ></embed>
```

8.canvas 画布标签

CSS 部分

CSS 指层叠样式表 (Cascading Style Sheets)，它们控制网页内容的外观。使用 CSS 设置页面样式时，可以将内容与表现形式分开。网页内容（即 HTML 代码）驻留在 HTML 文件自身中，而用于定义代码表现形式的 CSS 规则驻留在另一个文件（外部样式表*.css）或 HTML 文档的另一部分（通常为文件头部分）中。

写页面时要做到 结构(html)、样式(css)、行为(js)相分离，尽量做到完全解耦。

css 样式表的结构

CSS 样式表由选择器和声明两部分组成，其中选择器是通过名字来标识某个元素的（如元素名、类名或 id 名），声明用于定义元素样式。

```
a{ background-color : red; 声明 2; 声明 3; }
```

层叠样式表一共有几种使用（引入）方式？

行内样式表 `style=" "`

内部样式表 `<style>`

外部样式表 `<link rel="stylesheet" href=" " >`

导入 `@ url ("test.css");`

CSS 的基础选择器

id 选择器，class 选择器，标签选择器

`#id 名 { 属性名:属性值; }`

`.class 名 { 属性名:属性值; }`

`标签名 { 属性名:属性值; }`

css 基础选择器的优先级

`ID > CLASS > 元素 > 通配符 *`

CSS 基础样式



常见的一些样式

`color:red;` 字体颜色

`font-family:arial;` 字体样式

`text-align:center;` 文字内容居中显示，它有其它可选值 `left`、`right`

`background-color:red` 背景色

`border:1px solid green;` 为元素添加边框，设置 1 像素宽，实线，绿色

设置字的间距

```
p{letter-spacing:8px; }
```

<p>我想改变此行段落的字间距</p>

多出后隐藏

```
p{ height: 20px; width:100px; background:#ddd; overflow: hidden;}
```

<p>文字溢出时文本被隐藏，不信把 `overflow:hidden` 去掉试试</p>

表格边框变为单线 `border-collapse:collapse;`


```
table{border-collapse:collapse; }  
td{ border:1px solid red; } 用样式表只需要给 td 加边框，但要在 table 中设置 collapse
```

文字道行缩进 text-indent

尺寸 width height

文字溢出用省略号显示

```
p{ white-space:nowrap; overflow:hidden; text-overflow: ellipsis;}
```

text-decoration 指定文字有无装饰

可选值

underline 指定文字的装饰是下划线

overline 指定文字的装饰是上划线

line-through 指定文字的装饰是贯穿线，类似于删除标签的效果

text-align 内容对齐方式，它是针对于元素的内容或行内元素（块元素，行内元素不起作用）

属性值 left center right

背景相关

1.background-color 背景颜色

2.background-image 背景图片

```
background-image: url("../04I58PICaWv_1024.jpg"); width:100px; height:100px; background-size:100px; 需要指定宽和高。
```

3.background-repeat 背景图片是否重复

可选值

repeat-x 图片在横向上平铺

repeat-y 图片在纵向上平铺

no-repeat 不重复平铺

4.background-size 背景图片大小

5.background-attachment: 背景图片是否随内容滚动

fixed 背景图像相对于窗体固定，滚动条滚动图像不动，但它的容器会消失

6.background-position 背景图片位置（相对于外层容器）

可使用方位英文、百分比、确切的值进行定位

7.background 复合属性

```
background: url("boy.jpg") red no-repeat left top;
```

颜色相关

指定颜色的几种方式

颜色名称 {color:green;}

RGB {color:rgb(0,0,0);} 即是代表红、绿、蓝，以这三种颜色拼出的

RGBA {color:rgba(0,0,0,.5);} 前面三位 rgb 和上面一样，最后一位代表了透明度，这里是 0.5 的透明度

十六进制 {color:#ff0000;}

1.color 规定文本的颜色

2.opacity 设置元素颜色的透明度，它会进行继承，如果只是让背景有透明度，而不想让它里面的其它元素进行继承，可使用 background:rgba();

1 为不透明，0 为完全透明

字体相关

1. font-style 指定文本的字体样式（正常、斜体）

normal 正常字体

italic 斜体字

2. font-weight 指定文字的粗细（只有粗体和正常字体）

normal 正常的字体

lighter 细体 效果和 normal 一样

bold 粗体，相当于 number 为 700

bolder 也是粗体

以数字表示粗细，范围 100~900（600 以下是正常字体，600 以上是粗体）

3. font-size 指定文本字体大小

4. font-family 定义文本使用某个字体

SimSun、SimHei、Microsoft YaHei

列表相关

设置列表项标记的图像

list-style-image

```
.test{list-style-image:url(skin/ico.png);} //作用于 ul 而非 li 上
```

```
<ul class="test">
```

```
<li>列表项一</li>
<li>列表项一</li>
<li>列表项一</li>
</ul>
```

去掉列表的项目符号

```
list-style:none;
```

CSS 的其它选择器

首先 CSS 选择器分这么几大类：基础选择器、关系选择器、属性选择器、伪类选择器、伪对象选择器。

关系选择器

1. 包含选择符 (E F)

选择所有被 E 元素包含的 F 元素，中间用空格隔开 `ul li{ color:red; }`

1. 子选择符 (E > F)

选择所有作为 E 元素的直接子元素 F，对更深一层的元素不起作用，用大于号表示。

2. 相邻选择符 (E+F)

选择紧贴在 E 元素之后 F 元素，用加号表示，**选择相邻的第一个兄弟元素。**

3. 兄弟选择符 (E~F)

选择 E 元素之后的所有兄弟元素 F，作用于多个元素，用~号隔开。

属性选择器

属性选择器通过[] 进行定义。

1.E[att] 选择具有 attr 属性的 E 元素

option[disabled]{ color: red;} //选择带有 display 属性的 option 标签

2 E[att="val"] 选择具有 att 属性且属性值等于 val 的 E 元素。

input [type="text"] { ... }

input [type="submit"] { ... }

3 E[att\$="val"] 选择具有 att 属性且属性值为以 val 结尾的字符串的 E 元素

p[class\$="1"]{ color:red;} //选择 p 元素中 class 属性值以 p 结尾的元素

q<p class="p1 paragraph p11p">这是个段落</p>

<p class="p1 paragraph p11p">这是个段落</p>

<p class="p1paragraph p11">这是个段落</p>

<p class="p1">这是个段落</p>

4 E[att^="val"] 选择 E 元素中有 att 属性，且以" val" 开头的元素

p[class^="p"]{ color:red;} //选择带有 class 属性，属性值以 p 开头的元素

<p class="a1">这是个段落</p>

<p class="naragraph p11">这是个段落</p>

<p class="p11">这是个段落</p>

5 E[att*="val"] 选择具有 att 属性且属性值包含 val 的字符串的 E 元素。

a[href*="www"]{font-weight: bold;} //选择 a 标签中 href 属性值包含 baidu 的元素

百度

空的超链接

有道

6 E[att~="val"] 选择有 att 属性的元素，且属性值列表中含有一个符合 val 的元素

p[class~="p1"]{ color:red;} //选中 p 元素下 class 值包含 p1 的元素

<p class="p1 paragraph p11">这是个段落</p>

<p class="paragraph p11">这是个段落</p>

<p class="p">这是个段落</p>

伪类选择器

一般通过冒号进行定义

1.E:link 设置超链接 a 在未被访问前的样式（特指 a 标签）

2.E:visited 设置超链接 a 在其链接地址已被访问过时的样式（特指 a 标签）

3.E:hover 设置鼠标悬停在元素上时的样式。不限于 a 标签，p、li 也可以使用

4.E:active 设置元素在鼠标按下时的样式（不限于 a 标签）

5.E:not(s) 匹配不含有 s 选择器的元素 E。

```
p:not(.p1){ color:red;}
```

```
<p class="p1"></p>
```

```
<p></p>
```

```
<p></p>
```

6.E:first-child 父元素的第一个子元素 E。

满足两个条件 a.必须有父元素 b.它是父元素的第一个子元素

7.E:last-child 父元素的最后一个子元素 E

条件同上

8.E:only-child E 元素是唯一的子元素时被匹配

```
<body>
```

```
<p>段落</p> 此时 body 中只有这一个元素，所以 p:only-child{ color: deepskyblue;} 会生效
```

```
</body>
```

9.E:empty 匹配没有任何子元素（并且不包括空格、回车）的元素 E

```
p:empty{ border:1px solid red;padding:10px;}
```

```
<p></p>
```

```
<p> </p> //不会被选中
```

```
<div></div> //空，但不是 P 元素
```

```
<p>我是一个 p 元素</p> //有 text 文本节点
```

```
<p><a href="#">百度</a></p> //有子元素
```

10.E:checked 匹配用户界面上处于选中状态的元素 E。(用于 input type 为 radio 与 checkbox 时)

```
input:checked+span{color:red;}
```

```
<input type="radio" name="gender" /><span>男生</span>
```

```
<input type="radio" name="gender" checked="checked" /><span>女生</span>
```

*input 下被选中的元素紧挨着的 span 颜色为红色，因为选中状态会变，所以看着像动态的

伪对象选择器

伪对象也叫伪元素，在过去，伪类和伪元素都被书写成前面只加一个冒号，实际上伪对象应该用两个冒号来定义。

伪类和伪对象（元素）的区别

伪类一般反映无法在 CSS 中轻松或者可靠检测到的某个元素的状态或者属性；

伪元素表示 DOM 外部的某种文档结构

常用伪元素

1.E:before/E::before

2.E:after/E::after

伪元素是在元素内容的前面或后面定义的，必须要添加 content:"" 这个属性，不然伪元素无法定义成功。

```
p::after{ content:"在 P 标签元素内容前加了内容"; color: red;}
```

checked+伪元素练习

选中下面的项试试

- ☐ 蓝色
- ☒ 红色 我被选中了
- ☐ 黑色

选中下面的项试试

- ☐ 蓝色
- ☐ 红色
- ☐ 黑色

```
<style>
    input:checked+span{background:#f00;}
    input:checked+span::after{content:" 我被选中了";}
</style>

<form method="post" action="#">
    <fieldset>
        <legend>选中下面的项试试</legend>
        <ul>
            <li><label><input type="radio" name="colour-group" value="0" /><span>蓝色</span></label></li>
            <li><label><input type="radio" name="colour-group" value="1" /><span>红色</span></label></li>
            <li><label><input type="radio" name="colour-group" value="2" /><span>黑色</span></label></li>
        </ul>
    </fieldset>
    <fieldset>
        <legend>选中下面的项试试</legend>
        <ul>
            <li><label><input type="checkbox" name="colour-group2" value="0" /><span>蓝色</span></label></li>
```

```
<li><label><input type="checkbox" name="colour-group2" value="1" /><span>红色</span></label></li>
<li><label><input type="checkbox" name="colour-group2" value="2" /><span>黑色</span></label></li>
</ul>
</fieldset>
</form>
```

浮动

float 是 css 样式中的定位属性，用于设置标签的居左浮动和居右浮动，浮动后的元素不属于 html 文档流，需要用清除浮动把文档拽回到文档流中。

浮动值：

left: 向左浮动

right: 向右浮动

网页中大部分对象默认是占用文档流，也有一些对象是不占文档流的，比如表单中隐藏域。当然我们也可以让占用文档流的元素转换成不占文档流，这就要用到 CSS 中属性 position、float、display 来控制。默认情况下，所有元素都处在文档流中，四种情况将使得元素离开文档流：浮动 float、绝对定位 absolute、相对浏览器定位 fixed、元素不显示 display:none，这种情况不占文档流的空间，而普通元素的位置基于文档流。（了解）

清除浮动

1. 写一个空的 div

2. 给 div 添加 clear:both 属性

加浮动的目的是让两个块元素可并排显示，如果清除浮动，只是让块元素脱离文档流的特性消失，而并排显示的特性还在。

定位

通过 **position** 这个属性来定义，**position** 属性是指本体相对于上级的定位，**position** 定位又分绝对定位和相对定位。它的默认值是 **static**，意味着元素没有被定位，出现在文档流中应该出现的位置。如果用 **position** 来布局页面，父级元素的 **position** 属性必须为 **relative** 或 **absolute**。行元素加了 **position:absolute** 后可设置宽和高（加了 **float** 和 **fixed** 以后也可以设置宽和高）

常见属性值：

1.**static**：无特殊定位，对象遵循正常文档流。

2.**relative**：对象遵循正常文档流。

3.**absolute**：对象脱离正常文档流，使用 **top**, **right**, **bottom**, **left** 等属性进行绝对定位。使用绝对定位，那父级元素的 **position** 属性必须为 **relative** 或 **absolute**，如果不指定父级定义容器，将以浏览器窗口进行定位。

4.**fixed**：对象脱离正常文档流，使用 **top**, **right**, **bottom**, **left** 等属性以**浏览器窗口**为参考点进行定位，当出现滚动条时，对象不会随着滚动。IE6 及以下不支持

static、relative 正常文档流 **absolute、fixed** 脱离文档流

z-index

可以改变元素的叠放顺序，只有在使用了 **position** 并且，脱离了文档流（**absolute、fixed**）的情况下可以使用，其它情况下使用 **z-index** 不起作用。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    div{ width:100px; height: 100px; }
    .d1{ background: red; position: fixed; z-index: 99;}    //可改变 position 的四个值来查看效果（static、relative 不起作用，absolute、fixed 起作用）
    .d2{ background: blue;}
  </style>
</head>
<body>
  <div class="d1"></div>
  <div class="d2"></div>
</body>
</html>
```

display 属性

根据 CSS 规范的规定，每一个网页元素都有一个 **display** 属性，用于确定该元素的类型，每一个元素都有默认的 **display** 属性值，比如 **div** 元素，它的默认 **display** 属性值为“**block**”，称为块元素，而 **span** 元素的默认 **display** 属性值为“**inline**”，称为“行内”元素。

块元素与行元素是可以转换的，也就是说 **display** 的属性值可以由我们来改变。

display 常见属性值

1. **none**: 隐藏对象。
2. **inline**: 指定对象为内联元素
3. **block**: 指定对象为块元素
4. **inline-block**: 指定对象为内联块元素
5. **table-cell**: 指定对象作为表格单元格

使用它通常的作法是，配合 `vertical-align: middle;` `text-align: center;`使大小不一样的图片都可以居中显示(元素也可以居中)。

图片居中实例



```
.cell{display:table-cell; width:300px; height: 300px; border: 1px solid red; vertical-align: middle; text-align: center;}  
<div class="cell">  
    
</div>  
<div class="cell">  
    
</div>  
<div class="cell">  
    
</div>
```

CSS 优先级计算

选择器	特殊性 (a,b,c,d)	优先级
Style=" " "	1,0,0,0	1000
#wrapper #content {}	0,2,0,0	200
#content .datePosted {}	0,1,1,0	110
div#content {}	0,1,0,1	101
#content p {}	0,1,0,1	101
#content {}	0,1,0,0	100
p.comment .dateposted {}	0,0,2,1	21
div.comment p {}	0,0,1,2	12
.comment p {}	0,0,1,1	11
p.comment {}	0,0,1,1	11
.comment {}	0,0,1,0	10
div p {}	0,0,0,2	2
p {}	0,0,0,1	1

优先级数值比较：以数值大的为准
记住以下四个然后进行计算

style = 1000

id = 100

class = 10

element = 1

通常面试时问到的 CSS 选择器优先级是指“基础选择器”的优先级，如下

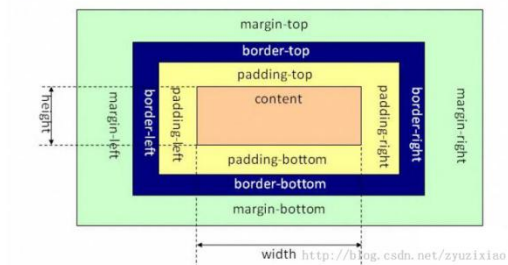
ID>CLASS>元素>*

important>内联>ID>类>标签|伪类|属性选择器>伪元素>继承>通配符

盒子模型

W3C 组织建议把网页上元素看成是一个个盒子。盒模型主要定义四个区域：内容(content)、内边距(padding)、边框(border)、外边距(margin)。

■ 标准盒子模型



CSS 盒子模型的属性

内容属性：宽=width 高=height（宽和高组成了 content 内容区）

内边距属性：padding

如果设置了背景图，那图片会延伸到 padding 的位置。padding 值不可以为负值。

单独设置某一个方向的内边距

1. padding-top:20px;
2. padding-right:30px;
3. padding-bottom:40px;
4. padding-left:50px;

边框属性：border

border 属性设置一个元素的边框，它有三个要素：宽、样式、颜色，统称“边框三要素”

border 的样式属性是必须的，如果不加 border 不显示，宽度、颜色可不设置，默认是 3 像素、黑色。

单独设置的写法

border-width
border-style
border-color

border-style 设置边框的样式，有五种常用样式可选

点状 dotted

实线 solid

双线 double (border:double 13px red; 需要最起码设置为 3 像素，不然显示不下)

虚线 dashed

无边框 none

外边距属性：margin（使用该属性的时候注意浏览器的兼容性）

margin 设置元素外边距的宽度，它有这么几个特点

1. 块级元素的垂直相邻外边距会合并（以大的值为准）
2. 而行内元素实际上不占上下外边距。行内元素的左右外边距不合并
3. 浮动元素的外边距也不会合并
4. 允许指定负的外边距值，不过使用时要小心

书写内边距与外边距时的规则（TRBL 顺序永远都是这个！）

如果有四个参数：四个参数分别表示上右下左

```
h2{ margin:10px 20px 30px 40px;}
```

如果只有一个参数：表示上右下左四个方向都为这一个值

```
h2{ margin:10px;}
```

如果有两个参数： 第一个参数表示上下 第二个参数表示左右（取对面的值）

```
h2{ margin:10px 20px;}
```

如果三个参数：表示上右 下，左边位置取右边的值。

```
h2{ margin:10px 20px 30px;} //最后一位是左，取对面右的值 20
```

*实际上，有缺省值时，如只有 2 个值（上右），剩下的下左会取对面的值

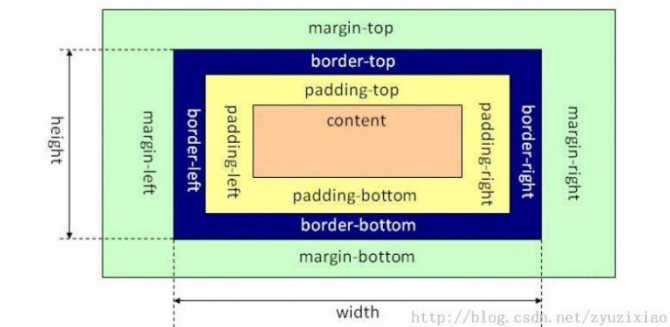
一个盒子实际占了多少空间？

content+padding+border+margin，这几个属性都要加起来，才是盒子的大小。也就是内容区域+内边距+边框+外边距。

☞ 怪异盒子模型

在不加 doctype 的情况下，IE 浏览器中会显示为怪异盒模型（也叫 IE 盒模型），它的宽和高包含了 border 和 Padding。

■ IE 盒子模型



伸缩盒子模型

伸缩盒模型也叫弹性盒模型，或 flexBox。可以轻松的创建“自适应”浏览器窗口的流动布局。它是 CSS3 中为 display 新添加的值类型。

常用属性

1. 父容器中的常用属性

```
display: flex;
```

定义一个 flex 容器

```
justify-content: flex-end;
```

设置项目的对齐方式，flex-end 代表内容右对齐，space-around(默认)代表每个项目两侧的间隔相等。

2. 项目中的常用属性

*伸缩盒中的每一个元素称为一个项目。

```
flex-grow: 1
```

属性定义项目的比例

```
.item1{ flex-grow: 1; background:pink}
```

```
.item2{ flex-grow: 2; background:orange}
```

```
.item3{ flex-grow: 3; background:red}
```

旧的伸缩盒

伸缩盒从被提出到如今一直在修改，所以这里涉及到了新老写法

伸缩盒最老版本

`display:box;`

将对象作为弹性伸缩盒显示（火狐和 webkit 内核都支持 `display:-webkit-box;`或 `display:-moz-box;`）

```
.cont{ display: -webkit-box;}    定义伸缩盒容器
.item{
    -moz-box-flex: 1;    每个项目按 1 比例划分
    -webkit-box-flex: 1;  多个兼容
    -ms-box-flex: 1;
    background: red;
    margin:5px;
}
```

伸缩盒过渡版本

`display:flexbox;`

将对象作为弹性伸缩盒显示

伸缩盒最新版本

`display:flex;`

将对象作为弹性伸缩盒显示

浏览器内核以及其前缀

根据不同的浏览器内核，css 前缀会有不同。最基本的浏览器内核有如下四种，其它的内核都是基于此四种进行再研发的。

1.Gecko 内核 前缀为-moz- 火狐浏览器

2.Webkit 内核 前缀为-webkit- 也叫谷歌内核，chrome 浏览器最先开发使用，safari 浏览器也使用该内核。国内很多浏览器也使用了 webkit 内核，如 360 极速、世界之窗、猎豹等。

3.Trident 内核 前缀为-ms- 也称 IE 内核

4.Presto 内核 前缀-o- 只有 opera 采用，但现在 opera 使用了由 webkit 衍生在 blink 内核

*由于 IE 浏览器在国内使用率非常高，所以造成了很多网银和支付只支持 IE 的 Trident 内核。但是它速度和兼容比不上 webkit 和 gecko 内核，于是出现了很多双核浏览器，比如遨游、搜狗、360。

`-webkit-border-radius: 25px;`

```
-moz-border-radius: 25px;
```

```
border-radius: 25px;    border:1px solid #ff6700;    width:100%;    height: 40px;;
```



清除浮动方式总结

1.使用伪元素清除浮动

```
.box::after{ content: " "; clear: both; display: block;}
```

```
.left,.right{ width:100px; height: 100px; background:#388bff; float: left;}
```

```
<div class="box">  
    <div class="left"></div>  
    <div class="right"></div>  
</div>
```

2.使用 overflow:hidden 清除浮动

```
.box{ overflow: hidden;} //overflow:auto 也是可以的
```

```
.left,.right{ width:100px; height: 100px; background:#388bff; float: left;}
```

```
<div class="box">  
    <div class="left"></div>
```

```
<div class="right"></div>  
</div>
```

3.使用空 div

```
.clear{ clear:both; }  
<div class=" clear" ></div>
```

CSS 继承

所谓 css 的继承是指被包在内部的标签将拥有外部标签的样式性质。继承是一种机制，它允许样式不仅可以应用于某个特定的元素，还可以应用于它的后代。例如一个 **body** 定义了颜色值也会应用到段落的文本中。

继承的局限性

在 css 中，继承是非常自然的行为，但是继承也有局限性。有些属性是不能继承的，比如 **border** 属性用来设置边框，它就没有继承性。**padding** 和 **margin** 也不能继承。

```
p{ color:red; border:1px solid blue; padding:20px; margin:10px }  
<p><i>这是 I 元素</i>这是 P 元素</p>
```




能被继承的常用属性有哪些？

color, cursor, font-family, font-size, font-style, font-weight, font, letter-spacing, line-height, list-style, text-align, text-indent

CSS3 基础样式

关于边框

border-radius

border-radius 还可以分别设置四个方向的值

```
border-radius:0px 10px 10px 0px;
```

border-radius可以快捷创建圆角

box-shadow

CSS3 中的 box-shadow 属性被用来添加阴影。

书写格式

box-shadow: 水平阴影的位置 垂直阴影的位置 模糊距离 颜色;

```
box-shadow:10px 20px 5px blue;
```

CSS3 文本效果

text-shadow (文字阴影)

通过这个属性可指定水平阴影，垂直阴影，模糊的距离，以及阴影的颜色。

```
p{  
    text-shadow: 5px 5px 5px #FF0000;  
}
```

<p>这里是文字效果</p>

CSS3 字体

以前 CSS3 的版本，网页设计师不得使用用户计算机上已经安装的字体。使用 CSS3，网页设计师可以使用他/她喜欢的任何字体。当你发现您要使用的字体文件时，只需简单的将字体文件包含在网站中，它会自动下载给需要的用户。

使用时，在项目中建立好 res 文件夹，用来放资源，把下载好的字体放入其中。

先用 @font-face 定义字体名和地址，再在样式表中，像使用其它字体一样使用

```
@font-face{  
    font-family: myFirstFont; //你自定义的字体名
```

```
src: url('res/lxksf.ttf'); //字体存放的路径
}
p{
font-family:myFirstFont; //使用
font-size:50px;
}
<p>当前文本使用了你不曾见过的字体</p>
```

过度 Transition

css 的 transition 允许 css 的属性值在一定的时间区间内平滑地过渡。

语法格式

transition: 属性名 时间 速度曲线 何时开始;

transition:width 2s ease-in 2s;

-webkit-transition: width 2s ease-in 3s;

width 因为我们是为宽度写了 hover 后的值，所以如果这里写 Height 属性是不会有过渡效果的

2s 整个过渡效果持续的时间

ease-in 指定了过渡的动画效果

3s 过 3 秒后开始执行过渡效果

动画效果的可选值

值	描述
linear	规定以相同速度开始至结束的过渡效果（等于 cubic-bezier(0,0,1,1)）。
ease	规定慢速开始，然后变快，然后慢速结束的过渡效果（cubic-bezier(0.25,0.1,0.25,1)）。
ease-in	规定以慢速开始的过渡效果（等于 cubic-bezier(0.42,0,1,1)）。
ease-out	规定以慢速结束的过渡效果（等于 cubic-bezier(0,0,0.58,1)）。
ease-in-out	规定以慢速开始和结束的过渡效果（等于 cubic-bezier(0.42,0,0.58,1)）。
cubic-bezier(<i>n</i> , <i>n</i> , <i>n</i> , <i>n</i>)	在 cubic-bezier 函数中定义自己的值。可能的值是 0 至 1 之间的数值。



示例网址 <http://www.mi.com/>

```
<style>
  a{
    background: #ddd;
    display: block;
    width:100px;
    height: 20px;
    line-height: 20px;
    text-decoration: none;
    color:#fff;
    font-family: "Microsoft YaHei";
    font-size:12px;
    text-align: center;
```

```
        transition:background .2s ease-in;
    }
    a:hover{ background:#ff6700;}
</style>
<a href="#">红米 Note 3</a>
```

微医健康 ^

示例网址 <http://www.guahao.com/>

```
a{
    display: block;
    width:100px; height: 20px; background:url("img/nav_arrow.png") no-repeat 80px 5px;
    transition:.5s;
}
a:hover{ background-position:80px -45px; }
<a href="#">北京尚学堂</a>
```



过渡改变 border 实例

```
.myborder{
    margin:300px;
    width:100px;
    height:100px;
    border-radius:50%;
    border:5px solid green;
}
.myborder:hover{
    border:50px solid deeppink;
}
.myborder{
    -webkit-transition: border 2s;
    -moz-transition: border 2s;
    -ms-transition: border 2s;
    -o-transition: border 2s;
    transition: border 2s;
}
```

Transform 转换

通常的属性包含了属性名和属性值，而 CSS3 的 transform 属性是用函数来定义的。

Transform 2D 最常用到的函数是 rotate(30deg)

书写格式

```
transform: rotate(30deg);
```

通过 rotate() 方法，元素顺时针旋转给定的角度。允许负值，元素将逆时针旋转。它以 deg 为单位，代表了旋转的角度。

实例



```
div{
  width:100px; height: 100px;
  background:url("img/stop.png") no-repeat center;
  background-size:50px;
  transition:0.5s linear;    //重点，不加即使转了也看不到效果
}
div:hover{ transform:rotate(360deg);}
<div></div>
```



```
div{
  width:100px; height: 100px;
  background:url("img/jt.png") no-repeat center;
  background-size:50px;
  transition:0.4s;
}
div:hover{ transform:rotate(180deg);}
```



```
<div></div>
```

transform3D 相关函数

rotateX()

rotate 的翻转，好比一张纸，在根据 X 轴水平的翻转，它是带有透视的。下图是从 10deg 到 180deg 的过程。

```
.myrotateX{  
    -webkit-transform: rotateX(120deg);  
    -moz-transform: rotateX(120deg);  
    -ms-transform: rotateX(120deg);  
    -o-transform: rotateX(120deg);  
    transform: rotateX(120deg);  
}
```

rotateY()

rotateY 与 rotate 函数一样，只不过是通过 Y 轴垂直翻转。

translate3d()

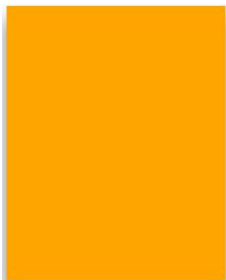
此函数用来规定指定元素在三维空间中的位移。它有三个值，x 轴,y 轴,z 轴。

语法结构:

`translate3d(x,y,z)`

- 1.x:表示在 x 轴方向的位移。
- 2.y:表示在 y 轴方向的位移。
- 3.z:表示在 z 轴方向的位移。

示例



```
div{  
  width:240px;  
  height: 300px;  
  background: orange;  
  transition:.2s linear;  
}  
div:hover{  
  transform:translate3d(0px,-5px,0px);  
  box-shadow: 0px 15px 30px rgba(0,0,0,.1);  
}
```

<div></div>



```
*{ margin:0px; padding:0px;}
div{
  width:200px;
  height: 300px;
  background:orange;
  position:relative;
  overflow: hidden;
  margin:50px;
```

```
        transition:.2s linear;
    }
    p{
        width:200px;
        height: 50px;
        background:red;
        position:absolute;
        bottom:-50px;
        transition:.2s linear;
    }
    div:hover{
        transform:translate3d(0px,-5px,0px);
        box-shadow: 0px 15px 30px rgba(0,0,0,.1);
    }
    div:hover p{ bottom:0px;}

<div>
    <p></p>
</div>
```

动画 animation

@keyframes 规则

如需在 CSS3 中创建动画，需要学习 @keyframes 规则。@keyframes 规则用于创建动画。在 @keyframes 中规定某项 CSS 样式，就能创建由当前样式逐渐改为新样式的动画效果。

写规则时至少包含以下两个属性

规定动画的名称

规定动画的时长

写好规则，再使用 animation 属性来应用到某个元素

```
animation: myAnimation 5s;
```

值	描述
linear	动画从头到尾的速度是相同的。
ease	默认。动画以低速开始，然后加快，在结束前变慢。
ease-in	动画以低速开始。
ease-out	动画以低速结束。
ease-in-out	动画以低速开始和结束。



```
img{
  width:100px; height: 100px; border-radius: 100%;
}
```

```
@keyframes CDturn{
  from{transform:rotate(0deg)}
```

```
        to{transform:rotate(360deg)}
    }

    img:hover{
        animation: 3s linear infinite CDturn;
    }

    
```

CSS 特殊图形

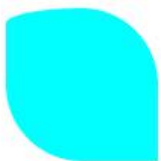
制作三角型使用的是 border 属性，内容区宽高值为 0

```
span{
    display:inline-block;
    width:0px;
    height: 0px;
    border-top:50px solid red;
    border-left:50px solid blue;
    border-right:50px solid orange;
    border-bottom:50px solid green;
```

```
}
```

梯形（和三角形不同的是添加了宽度）

```
div {  
    border-bottom: 80px solid red;  
    border-left: 50px solid transparent;  
    border-right: 50px solid transparent;  
    height: 0;  
    width: 100px;  
}
```



```
.huxing{  
    height: 100px;  
    width: 100px;  
    background-color: cyan;  
    border-bottom-left-radius: 50%;  
    border-top-left-radius: 50% 10%;  
}
```



```
} border-top-right-radius: 50%;
```

微信对话框





微信完整实例

```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
```

```
<title></title>
<style>
    body{
        background:url("img/new-bg.jpeg") ;
        background-size:100%;
        min-width:420px;
    }
    /*对方*/
    .other-side img{
        width:50px;
        height: 50px;
        border-radius:5px;
        float:left;
    }
    .send{
        position: relative;
        width: 300px;
        height: 50px;
        margin-bottom:40px;
        float:left;
    }
    .arrow{
        width:0px;
        height:0px;
        border:8px solid #000;
```

```
        border-color: transparent rgb(172,218,71) transparent transparent;
        position: absolute;
        z-index: 99;
        top: 10px;
        left: 4px
    }
    .text{
        width: auto;
        height: auto;
        background: rgb(172,218,71);
        border-radius: 5px;
        position: absolute;
        left: 20px;
        color: #000;
        padding: 12px;
        box-sizing: border-box;
        font-family: "microsoft YaHei";
    }

    /*自己*/
    .self{
        float: right;
        clear: both;
    }
    .self img{
```

```
float: right;
width: 50px;
height: 50px;
border-radius: 5px;
}
.right-send{
float: right;
margin-right: 20px;
}
.right-arrow{
width: 0px;
height: 0px;
border: 8px solid #000;
border-color: transparent transparent transparent #fff; /*rgb(172,218,71)*/
position: absolute;
z-index: 99;
top: 10px;
right: -16px;
}
.right-text{
background: #fff;
right: 0px;
left: auto;
}
.time{
```

```
        background-color:rgba(0,0,0,.4);
        width:80px;
        height: 20px;
        line-height: 20px;
        font-size:12px;
        color:#fff;
        text-align: center;
        clear: both;
        margin:0px auto;
        border-radius:4px;
        margin-bottom:20px;
    }
</style>
</head>
<body>
    <!--对方-->
    <div class="other-side">
        
        <div class="send">
            <div class="arrow"></div>
            <div class="text">
                欢迎回到微信 欢迎回到微信 欢迎回到微信 欢迎回到微信
            </div>
        </div>
    </div>
</div>
```

```
<!--自己-->
<div class="self">
  
  <div class="right-send send">
    <div class="right-arrow"></div>
    <div class="text right-text">
      哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈
    </div>
  </div>
</div>
<!--时间-->
<div class="time">晚上 19: 32</div>

<!--对方-->
<div class="other-side">
  
  <div class="send">
    <div class="arrow"></div>
    <div class="text">
      欢迎回到微信 欢迎回到微信 欢迎回到微信 欢迎回到微信
    </div>
  </div>
</div>
</body>
```

```
</html>
```

reset 文件

reset.css 可以将浏览器默认的样式清除掉，这样做会使我们的 CSS 或 html 标签更加方便准确。假如我们不初始化 CSS 样式属性，将会增大 CSS 代码量，所以使用初始化文件会为我们节约网页代码，节约网页下载时间；还会使得我们开发网页内容时更加方便简洁，不用考虑很多。

为什么我们不建议使用通配符？

我们有的时候会使用 `* {padding:0;margin:0;}` 来清除浏览器的一些样式，这种方式很实用，*号匹配所有元素，省去了一个一个写元素名称的麻烦。但是它在性能方面是会对页面造成影响的，试想一下，你的页面非常丰富，嵌套多、元素多，这时候，*号会去每一个元素都去渲染一遍。而我们通常只需要对表格、列表、标题等标签进行初始化样式。

有衬线和无衬线字体的区别

西方人一般把字体分为有衬线和无衬线，有衬线体是指字的开始、结束笔划中有修饰（也就是粗细的变化），如宋体字。而无衬线字体是指字体的组细没有变化，如微软雅黑。

简短，醒目的文字多用无衬线字体，而段落性的文字要使用有衬线字体，因为这样具有更好的可读性。



PS 快捷键

ctrl+tab 不同文件间的切换

alt+空格+鼠标左键 缩小

ctrl+空格+鼠标左键 放大

ctrl+ “+” 放大

ctrl+ ” - ” 缩小

还原到原始尺寸 空格+右键，选择实际像素

空格+左键 可拖动过大的图片

ctrl+backspace 当前图层快速填充背景色

alt+backspace 当前图层快速填充前景色

ctrl+} 向上切换图层

ctrl+[向下切换图层

关闭文件 ctrl+w

打开文件 双击空白处

创建文件 ctrl+双击空白处

创建当前图层复本 ctr+J

创建新图层 ctrl+shift+n

面板

F7 显示图层

F8 显示当前操作信息

Tab 隐藏其它工具栏，只留图片本身

工具

切片工具 k （右击切片，可编辑切片）

钢笔工具 p

魔棒 w

选择工具 v （ctrl+v+左键，可快捷选中被点击的图层）

放大镜 z

矩形/圆形 m （通过组合 shift 键可切换）



阿里字体图标

使用方式

- 1.地址 <http://www.iconfont.cn/>
- 2.使用相关平台登录
- 3.点击图标库
- 4.选中喜欢的图标，点击购物车按钮
- 5.点击存储为项目，起个名字
- 6.在生成的项目中可获取在线链接，这是使用了 cdn 的，阿里的图标库 cdn 不是很稳定
- 7.我们一般选择“下载到本地”
- 8.解压完，它里面有几个文件，除了 demo.css 和 demo.html，其它文件放进项目
- 9.在 html 中引入 iconfont.css
- 10.使用时要在元素中添加” iconfont” 类
``

CSS Sprite（雪碧图）

什么是 CSS Sprite

CSS Sprite 也叫 CSS 精灵、CSS 雪碧图，是一种网页图片应用处理方式。它允许你将一个页面涉及到的所有零星图片都包含到一张大图中去，这样一来，当访问该页面时，由从前的多次请求变为一次请求。客户端每显示一图片都会向服务器发送请求，所以图片越多请求越多，造成延迟的可能性越大。所以页面有许多 icon 时，推荐使用 CSS Sprite。



CSS Sprite 原理

其实就是把网页中的一些背景图片整合到一张图片文件中，再利用 CSS 的 background-image、background-position 属性，对这张大图进行定位。background-position 可以用数字能精确的定位出背景图片的位置。

CSS Sprites 优点

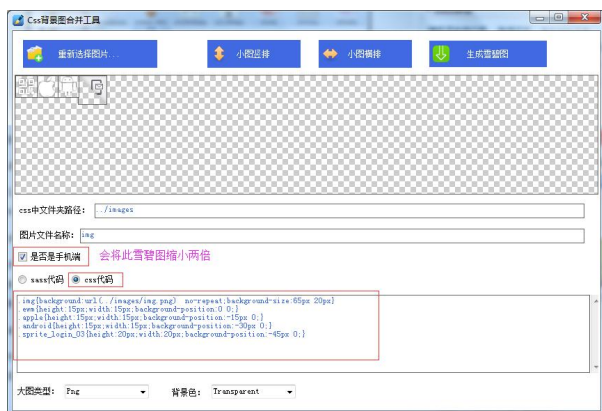
1. 利用 CSS Sprites 能很好地减少了网页的 http 请求，从而大大的提高了页面的性能，这也是 CSS Sprites 最大的优点，也是其被广泛传播和应用的主要原因；
2. CSS sprites 能减少图片的字节，(网络传输以字节为单位，1MB=1024 千字节)，三张图片大小的总合大于拼成一张图片的大小。
3. 解决了命名困扰，只需要对一张图片命名，而非数十个小图片命名

用一句话来概述就是

“CSS Sprites(图片整合技术) 的目的就是通过整合图片，减少对服务器的请求数量，从而加快页面加载速度。”



雪碧图制作工具



如何使用雪碧生成工具

1.用 PS 把需要的图切出来

2.打开 CssSprite.exe 文件

下载地址: <http://download.csdn.net/detail/wx247919365/8641795>

3.点击左上角按钮打开图片

4.排列图片

可以点击按钮来排列,也可以自己拖动排列,拖动完程序会根据图片的位置生成面积最小的雪碧图

5.是否是手机端

勾选后雪碧图的大小会缩小 2 倍

6.代码生成

有 sass 和 css 两个选项，选择后者。sass 是一种 CSS 预处理语言，能清晰的、结构化的描述文件样式。

7.大图类型要选择 png，图片背景色为 Transparent 透明

滑动门

使用 css 实现滑动门，要遵循的结构为

```
<ul>
  <li>
    <a href="#">列表内容一</a>    //鼠标划过时的列表
    <div>滑动后显示的内容区</div>    //鼠标划过要显示的模块，初始时为 display:none 隐藏
  </li>
  <li>
    <a href="#">列表内容二</a>
    <div>滑动后显示的内容区</div>
  </li>
</ul>
```

原理

1.将 li 下的 div 基于 ul 进行定位，如果是基于 li 进行定位，那么每一次滑动都会改变位置

```
ul{ position:relative}
```

```
ul li div{position:absolute;top:0px;left:100px;width: 300px;height: 500px;display: none;background:red}
```

2.当鼠标划过 li 后，让 li 下的 div 的 display 属性值变为 block; (初始为 display:none;)

```
ul li:hover div{display: block}
```

一个基本的 css 滑动门

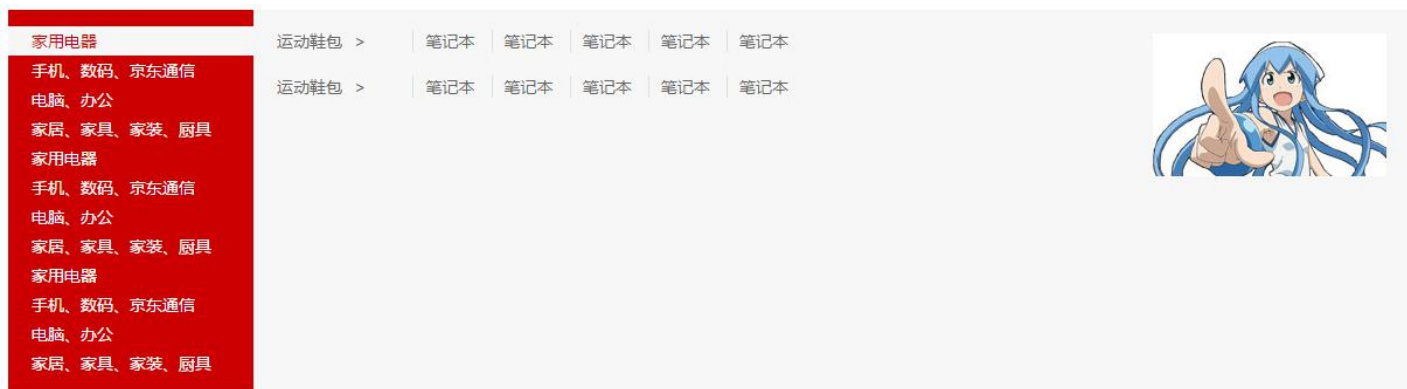
```
<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title></title>
  <style>
    ul{ position:relative}
    ul li div{display: none; width: 300px;height:100px; background:orange; position:absolute; top:0px;left:120px;}
    ul li:hover div{display: block}
  </style>
</head>
<body>
  <ul>
    <li>
      <a href="#">列表内容一</a>
      <div>aaaaa 滑动后显示的内容区 aaaaa</div>
    </li>
    <li>
      <a href="#">列表内容二</a>
```

```

<div>bbbb 滑动后显示的内容区 bbbb</div>
</li>
</ul>
</body>
</html>

```

完整滑动门实例



```

<!DOCTYPE html>
<html>
<head lang="en">
  <meta charset="UTF-8">
  <title>纯 CSS 滑动门导航</title>
  <style>
    body{ font-family:"Microsoft YaHei"; font-size:14px;}
    /*封装清除的类*/

```



```
.clearfix::after{
    content: "";
    display: block;
    clear: both;
    visibility: hidden;
}
/*end*/
.wrap{ position:relative;}
/*导航列表*/
.nav-list{ background-color:#cc0000; width:210px; float: left; }
.nav-list ul{padding-left:0px; width:210px;}
.nav-list ul li{ list-style: none; line-height: 25px; cursor: pointer; padding-left:20px; }
.nav-list ul li:hover{ background-color:#f7f7f7; }
.nav-list ul li:hover>a{ color:#cc0000; }
.nav-list ul li a{ color:#fff; text-decoration: none;}
.nav-list ul li:hover>.nav-cont{ display: block;}
/*内容区*/
.nav-cont{ width:1000px; height: 328px; background:#f7f7f7; float: left; position:absolute; left:210px; top:0px; display: none;}
.cont-href{ width:700px; float: left;}
.nav-cont .cont-href a{ text-decoration:none; color:#666;}
.cont-href dt,.cont-href dd{float: left; }
.cont-href dt{ margin-left:20px; }
.cont-href dt i{ font-style: normal; color:#666; margin-left:10px;}
.cont-href dd a{ padding:0px 10px; border-left:1px solid #e0e0e0}
```

```
/*内容区的图片*/
.cont-icon{ width:auto; float: right;}
.cont-icon img{margin-right: 30px; margin-top:20px;}
</style>
</head>
<body>
  <div class="wrap">
    <div class="nav-list">
      <ul>
        <li>
          <a href="#">家用电器</a>
          <div class="nav-cont">
            <div class="cont-href">
              <dl class="clearfix">
                <dt><a href="#">运动鞋包</a><i>&gt;</i></dt>
                <dd>
                  <a href="#">笔记本</a>
                  <a href="#">笔记本</a>
                  <a href="#">笔记本</a>
                  <a href="#">笔记本</a>
                  <a href="#">笔记本</a>
                </dd>
              </dl>
            <dl class="clearfix">
              <dt><a href="#">运动鞋包</a><i>&gt;</i></dt>
```

```
                <dd>
                    <a href="#">笔记本</a>
                    <a href="#">笔记本</a>
                    <a href="#">笔记本</a>
                    <a href="#">笔记本</a>
                    <a href="#">笔记本</a>
                </dd>
            </dl>
        </div>
        <div class="cont-icon">
            
        </div>
    </div>

</li>
<li>
    <a href="#">手机、数码、京东通信</a>
    <div class="nav-cont">
        <div class="cont-href">
            <dl class="clearfix">
                <dt><a href="#">运动鞋包</a><i>&gt;</i></dt>
                <dd>
                    <a href="#">笔记本</a>
                    <a href="#">笔记本</a>
                    <a href="#">笔记本</a>
```

```
                <a href="#">笔记本</a>
                <a href="#">笔记本</a>
            </dd>
        </dl>
        <dl class="clearfix">
            <dt><a href="#">运动鞋包</a><i>&gt;</i></dt>
            <dd>
                <a href="#">笔记本</a>
                <a href="#">笔记本</a>
                <a href="#">笔记本</a>
                <a href="#">笔记本</a>
                <a href="#">笔记本</a>
            </dd>
        </dl>
    </div>
    <div class="cont-icon">
        
    </div>
</div>
</li>
<li>
    <a href="#">电脑、办公</a>
    <div class="nav-cont">
        <div class="cont-href">
            <dl class="clearfix">
```

```
<dt><a href="#">运动鞋包</a><i>&gt;</i></dt>
<dd>
  <a href="#">汽车汽车</a>
  <a href="#">汽车汽车</a>
  <a href="#">汽车汽车</a>
  <a href="#">汽车汽车</a>
  <a href="#">汽车汽车</a>
</dd>
</dl>
<dl class="clearfix">
  <dt><a href="#">运动鞋包</a><i>&gt;</i></dt>
  <dd>
    <a href="#">汽车汽车</a>
    <a href="#">汽车汽车</a>
    <a href="#">汽车汽车</a>
    <a href="#">汽车汽车</a>
    <a href="#">汽车汽车</a>
  </dd>
</dl>
</div>
<div class="cont-icon">
  
</div>
</div>
</li>
```

```
<li>
  <a href="#">家居、家具、家装、厨具</a>
  <div class="nav-cont">
    <div class="cont-href">
      <dl class="clearfix">
        <dt><a href="#">沙发沙发</a><i>&gt;</i></dt>
        <dd>
          <a href="#">沙发</a>
          <a href="#">沙发</a>
          <a href="#">沙发</a>
          <a href="#">沙发</a>
          <a href="#">沙发</a>
        </dd>
      </dl>
      <dl class="clearfix">
        <dt><a href="#">沙发沙发</a><i>&gt;</i></dt>
        <dd>
          <a href="#">沙发</a>
          <a href="#">沙发</a>
          <a href="#">沙发</a>
          <a href="#">沙发</a>
          <a href="#">沙发</a>
        </dd>
      </dl>
    </div>
  </div>
</li>
```

```
<div class="cont-icon">
    
</div>
</li>
<li>
    <a href="#">家用电器</a>
    <div class="nav-cont">
        <div class="cont-href">
            <dl class="clearfix">
                <dt><a href="#">手机手机</a><i>&gt;</i></dt>
                <dd>
                    <a href="#">手机</a>
                    <a href="#">手机</a>
                    <a href="#">手机</a>
                    <a href="#">手机</a>
                    <a href="#">手机</a>
                </dd>
            </dl>
            <dl class="clearfix">
                <dt><a href="#">手机手机</a><i>&gt;</i></dt>
                <dd>
                    <a href="#">手机</a>
                    <a href="#">手机</a>
                    <a href="#">手机</a>
```

```
                <a href="#">手机</a>
                <a href="#">手机</a>
            </dd>
        </dl>
    </div>
    <div class="cont-icon">
        
    </div>
</div>
</li>
<li><a href="#">手机、数码、京东通信</a></li>
<li><a href="#">电脑、办公</a></li>
<li><a href="#">家居、家具、家装、厨具</a></li>
<li><a href="#">家用电器</a></li>
<li><a href="#">手机、数码、京东通信</a></li>
<li><a href="#">电脑、办公</a></li>
<li><a href="#">家居、家具、家装、厨具</a></li>
</ul>
</div>
</div>
</body>
</html>
```


多列

CSS3 中新出现的多列布局(multi-column)是传统 HTML 网页中块状布局模式的有力扩充。这种新语法能够让 WEB 开发人员轻松的让文本呈现多列显示。它的显示如同 word 中的多列。但在 CSS3 多列出现前，想要多列显示，要么使用定位，要么手动给文本分段落。

多列有这么几个常用属性

column-count 分几列

column-gap 列间距

column-rule 列分割线的样式

多列实例

```
div{
    /*分几列*/
    -webkit-column-count: 3;
    -moz-column-count: 3;
    column-count: 3;
    /*列间距*/
    -webkit-column-gap: 50px;
    -moz-column-gap: 50px;
    column-gap: 50px;
    /*分割线样式*/
    -webkit-column-rule: 3px dotted red;
    -moz-column-rule: 3px dotted red;
```

```
column-rule: 3px dotted red;  
}
```

resize

Resize

CSS 中，**resize** 属性指定一个元素是否应该由用户去调整大小。此属性可以通过指定它，实现类似 **textarea** 的效果。

它的可选值

none 用户无法调整元素的尺寸。

both 用户可调整元素的高度和宽度。

需要注意的是，使用 **resize** 要设置 **overflow** 属性，并且，值不能是 **visible**（**overflow:visible** 是元素的默认属性，该值规定内容不被隐藏），我们通常使用 **resize** 属性时会设置 **overflow:auto**。

***overflow** 属性：规定当内容溢出元素框时发生的事情

```
<style>  
div{  
width:231px;
```

```
        height:155px;
        border:1px solid rgb(169,169,169);
        resize:both;    //both 可调整高和宽
        overflow: auto;    //要加 overflow:auto, 不然不生效, 设置 hidden 不出了滚动条
    }
</style>

<textarea name="" id="" cols="30" rows="10"></textarea>
<div></div>
```

Box-sizing

Box-sizing

box-sizing 属性允许你以“W3C 的盒模型”或“IE 盒模型”来定义元素，以适应区域。换句话说，当前元素使用哪种盒模型，可以由 box-sizing 属性来指定了。

它有两个值

content-box（标准）

padding 和 border 不被包含在 width 和 height 内，元素的实际大小为宽高+border+padding，此为标准模式下的盒模型。

border-box（怪异）

padding 和 border 被包含在定义的 width 和 height 中，元素实际的大小为你定义了多少宽就是多宽。此属性为怪异模式下的盒模型。

```
<style>
  div{ width:300px; height: 100px; background: yellowgreen; padding:10px; box-sizing: border-box; }
</style>
<div>
  微信对话内容，如果加了 padding 不想让对话框的大小改变，可加 box-sizing 属性设置为怪异盒模型
</div>
```

META 标签详解

meta 通常以名值对的形式出现

```
<title>《今日头条》你关心的,才是头条! - TouTiao.com</title>
```

```
<meta name="keywords" content="今日头条,头条,头条网,头条新闻,今日头条官网">
```

meta 标签通常是用来描述 HTML 文档的，而 meta 的“名”的部分是用来说明，描述该文档的哪个部分。而“值”是这个被描述的具体信息。

```
<meta name="关键词" content="关键词 1, 关键词 2, 关键词 3" />
```

meta 标签的常用属性

http-equiv 属性

当服务器向浏览器发送文档时，会先发送请多名值对，来告诉浏览器一些信息（做好准备接收哪种类型文档的准备），它也是 meta 标签中名值对中的“名”。

Content-Type，它告诉浏览器准备接受一个 HTML 文档

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<meta charset="utf-8" />
```

refresh，它在指定时间内自动刷新，并且可跳转到指定的 url

```
<meta http-equiv="refresh" content="2;url=http://www.baidu.com" />
```

 如果只写时间，自动刷新本页面

name 属性

提供了名值对中的名称。

name 的常见值有

author，它定义了该页面的作者

```
<meta name="author" content="bjsxt">
```

keywords, 定义了 html 文档关键字

```
<meta name="keywords" content="meta 总结,html meta,meta 属性,meta 跳转">
```

description, 用于描述文档

```
<meta name="description" content="HTML examples">
```

另外一些常见的 meta

```
<meta http-equiv="X-UA-Compatible" content="IE=7">
```

X-UA-Compatible 是针对 ie8 新加的一个设置, 对于 ie8 之外的浏览器是不识别的, 它的目的是: 将 IE8 使用 IE7 进行渲染, 以避免页面在 IE8 下面出现错误。

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

适用于移动端的 meta 标签, width 属性控制设备的宽度。假设你的网站被带有不同屏幕分辨率的设备浏览, 那么将它设置为 device-width 可以确保它能正确呈现在不同设备上。

initial-scale=1.0 确保网页加载时, 以 1:1 的比例呈现, 不会有任何的缩放。

禁止用户缩放, 可以在混合 APP 时, 为了使 html 页面更逼真, 使页面无法缩放

```
<meta name="viewport" content="initial-scale=1.0, width=device-width, user-scalable=no" /> //user-scalable=no 是禁止缩放
```

媒体查询

响应式的核心 CSS 技术 MediaQuery 媒体查询

为什么需要媒体查询

为的是给不同的媒体类型设定专有的样式表，比如一个页面在电脑屏幕上显示时使用无衬线字体，在打印时要使用有衬线字体。使用媒体查询让样式表有更强的针对性。

CSS 中的 Media Query（媒介查询）是什么？

作为 CSS3 规范的一部分，媒体查询可以针对不同的屏幕尺寸设置不同的样式，它为每种类型的用户提供了最佳的体验，网站在任何尺寸设置下都能有最佳的显示效果。

通过不同的媒体类型和条件定义样式表规则。

媒体查询让 CSS 可以更精确作用于不同的媒体类型和同一媒体的不同条件。

媒体查询的大部分媒体特性都接受 min 和 max 用于表达“大于或等于”和“小与或等于”。如：width 会有 min-width 和 max-width

Media Queries 功能是非常强大的，他可以让你定制不同的分辨率和设备，并在不改变内容的情况下，让你制作的 web 页面在不同的分辨率和设备下

都能显示正常，并且不会因此而丢失样式

媒体查询的基本语法

```
@media mediatype and|not|only (media feature) {  
    CSS-Code;  
}
```

以上通过@media 定义媒体查询，mediatype 代表了设备类型，目前只有 screen 最常用，and|not|only 为条件，media feature 为媒体特点，通常是写设备的宽度。

@media screen and (max-width:960px) 的意思为：当前设备为 screen（电脑、平板、手机）时，并且最大宽度为 960 时，显示的样式。

媒体类型有：

all 所有设备

print 用于打印机和打印预览

screen 用于电脑屏幕，平板电脑，智能手机等。

speech 应用于屏幕阅读器等发声设备

媒体特点：

设备特性	是否允许 min/max 前缀	特性的值	说 明
width	允许	像素值的数值	指定浏览器窗口的宽度大小, 如 480 像素。
height	允许	像素值的数值	指定浏览器窗口的高度大小, 如 320 像素。
device-width	允许	像素值的数值	指定移动设备的屏幕分辨率宽度大小, 如 480 像素。
device-height	允许	像素值的数值	指定移动设备的屏幕分辨率高度大小, 如 320 像素。
orientation	不允许	字符串值	指定移动设备浏览器窗口的方向, 只能指定 portrait (纵向) 和 landscape (横向) 两个值。
aspect-ratio	允许	比例值	指定移动设备浏览器窗口的纵横比例, 如 16:9。
device-aspect-ratio	允许	比例值	指定移动设备屏幕分辨率的纵横比例, 如 16:9。
color	允许	整数值	指定移动设备使用多少位的颜色值。
color-index	允许	整数值	指定色彩表的色彩数。
monochrome	允许	整数值	指定单色缓冲器中每像素的字节数。
resolution	允许	分辨率值	指定移动设备屏幕的分辨率。

Bootstrap 使用

Bootstrap, 来自 Twitter, 是目前很受欢迎的前端框架。Bootstrap 是基于 HTML、CSS、JAVASCRIPT 的, 它简洁灵活, 使 Web 开发更加快捷。它对 HTML、CSS 和 JAVASCRIPT 进行了封装, 使它们使用起来更方便。我们只需要使用它已经设定好的类, 或规则, 即可快速应用它提供的功能。

Bootstrap 使用了一些 HTML5 元素和 CSS3 属性(如媒体查询)。为了让这些正常工作, 需要使用 HTML5 文档类型<!DOCTYPE html>。如果在 Bootstrap 创建的网页开头不使用 HTML5 的文档类型 (Doctype), 可能会面临一些浏览器显示不一致的问题。



怎么使用 bootstrap?

1. 下载

<http://www.bootcss.com/> 点击 bootstrap3 中文文档，点击下载 bootstrap3

2. 下载“用于生产环境的”

下载完后它有如右三个文件夹

3. 在 html 文档中引入 bootstrap

- a) bootstrap.min.css
- b) bootstrap.min.js
- c) jquery

引入 css 中的“bootstrap.min.css”和 js 文件夹下的 bootstrap.min.js。由于 bootstrap 是基于 jquery 的，所以在引入“bootstrap.min.js”前，要先引入 jquery。并且 js 和 jquery 要放在 body 的最底部。

4. 使用 html5 的文档类型声明

<!doctype html>，bootstrap 用到了 h5，所以它包含一个 html5 版本的 DOCTYPE

5. 添加适用于移动端的 meta 标签

<meta name="viewport" content="width=device-width, initial-scale=1.0">

因为现在越来越多的用户使用移动设备,为了让 Bootstrap 开发的网站对移动设备友好，确保适当的绘制和触屏缩放，需要在网页的 head 之中添加 viewport meta 标签

`width` 属性控制设备的宽度。假设你的网站被带有不同屏幕分辨率的设备浏览，那么将它设置为 `device-width` 可以确保它能正确呈现在不同设备上。
`initial-scale=1.0` 确保网页加载时，以 1:1 的比例呈现，不会有任何的缩放。

最方便的方式是使用“基本模板”

在 bootstrap 中文网-->起步-->基本模板

```
<!DOCTYPE html>
<html lang="zh-CN">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- 上述3个meta标签*必须*放在最前面，任何其他内容都*必须*跟随其后! -->
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
    <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
    <!--[if lt IE 9]>
      <script src="//cdn.bootcss.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="//cdn.bootcss.com/respond.js/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>你好，世界！</h1>

    <!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
    <script src="//cdn.bootcss.com/jquery/1.11.3/jquery.min.js"></script>
    <!-- Include all compiled plugins (below), or include individual files as needed -->
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

复制

bootstrap 基本模板详解

```
<!DOCTYPE html>
<html lang="zh-CN">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- 上述 3 个 meta 标签*必须*放在最前面, 任何其他内容都*必须*跟随其后! -->
    <title>Bootstrap 101 Template</title>

    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

    <!--[if lt IE 9]>
      <script src="//cdn.bootcss.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script src="//cdn.bootcss.com/respond.js/1.4.2/respond.min.js"></script>
    <![endif]-->
  </head>
  <body>
    <h1>你好, 世界! </h1>
    <script src="http://apps.bdimg.com/libs/jquery/2.1.4/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
```

[X-UA-Compatible](#) 是 IE8 的一个专有<meta>属性, 它告诉 IE8 采用何种 IE 版本去渲染网页, [IE=edge](#) 会以最高版本 IE 来渲染页面([ie8](#) 有两个引擎)

[html5shiv.min.js](#) 用于解决 IE9 以下版本浏览器对 HTML5 新增标签不认识, 并导致 CSS 不起作用的问题

用于为不支持 CSS3 Media Queries 的浏览器提供媒体查询的功能

可使用 [cdn](#), 要放在 [bootstrap.js](#) 之前

```
</body>
</html>
```

Bootstrap 栅格系统

BS 栅格系统

Bootstrap 包含了一个响应式的、移动设备优先的、不固定的网格系统，可以随着设备或视口大小的增加而适当地扩展到 12 列。它包含了用于简单的布局选项的预定义类，也包含了用于生成更多语义布局的功能强大的混合类。



```
<div class="container">
  <div class="row">
    <div class="col-md-4" style="background:red;">
      <span>这是左边</span>
      <span>这是左边</span>
    </div>
    <div class="col-md-4" style="background:yellow;">
      <h1>你好，世界！</h1>
    </div>
    <div class="col-md-4">
```

```
<h1>你好，世界！</h1>
</div>
</div>
</div>
```

应该怎么使用 bootstrap 栅格呢？

1. 行必须放置在 `.container class` 内，以便获得 bootstrap 设置的对齐（alignment）和内边距（padding）
2. 使用 `.row` 来创建一行，每行又分为多个列。
3. 内容应该放置在列内，且唯有列可以是行的直接子元素。`row>col` 正确写法 行>列
4. 预定义的网格类，比如 `.row` 和 `.col-xs-4`，可用于快速创建网格布局。
5. 列通过内边距（padding）来创建列内容之间的间隙。该内边距是通过 `.rows` 上的外边距（margin）取负值得到，就是说，列和列之间有间隙，你想改变间隙，用负的 margin 来写。
6. 网格系统是通过指定你想要横跨的十二个可用的列来创建的。例如，要创建三个相等的列，则使用三个 `.col-xs-4`。

设备	超小设备手机 (<768px)	小型设备平板电脑 (≥768px)	中型设备台式电脑 (≥992px)	大型设备台式电脑 (≥1200px)
最大容器宽度	None (auto)	750px	970px	1170px
Class 前缀	<code>.col-xs-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>
列数量和	12	12	12	12
最大列宽	Auto	60px	78px	95px
间隙宽度	30px (一个列的每边分别15px)	30px (一个列的每边分别15px)	30px (一个列的每边分别15px)	30px (一个列的每边分别15px)

bootstrap 栅格把设备分为了四种：超小设备（手机）、小型设备（平板）、中型设备（笔记本电脑）、大型设备（大尺寸显示器），这四种设备它

们各自的列数量和都不可以超过 12，它们的列有不同的 class 前缀做标识，超小设备用.col-xs，小型设备用.col-sm，中型设备用.col-md-，大型设备用.col-lg-。不论哪种设备，它们的间隙都是 30px(指列的 padding-left:15px 和 padding-right:15px;)

```
<div class="container">
  <div class="row">
    <div class="col-md-4 col-sm-6">手机时我自占一行</div>
    <div class="col-md-4 col-sm-6">a</div>
    <div class="col-md-4">a</div>
  </div>
</div>
```

Bootstrap（一）全局样式

【副标题】

如果需要向任何标题添加一个内联子标题，只需要在标题标签中添加 <small>，用 small 标签包裹复标题的内容，或者添加 .small 类，这样子您就能得到一个字号更小的颜色更浅的文本。

实例

```
<h1>这是标题<small>这是 h1 中的副标题</small></h1>
```

```
<h1>这是标题<span class="small">这是用 small 类来写的副标题</span></h1>
```


【引导主体副本】

为了给段落添加强调文本，则可以添加 `class="lead"`，这将得到更大、行高更高的文本，它的作用相当于是文章摘要。

实例

```
<h2>引导主体副本</h2>
```

```
<p class="lead">闭包闭包闭包闭包闭包闭包闭包闭包闭包闭包闭包</p>
```

【对齐类】

简单方便将文字对齐的类，它们一般应用在 `p` 标签上（用在 `div` 上不符合规范，用在 `span` 标签则不能生效）。

`.text-left`

```
<p class="text-left">文字左对齐</p>
```

`.text-center`

```
<p class="text-center">文字居中对齐</p>
```

`.text-right`

```
<p class="text-right">文字右对齐</p>
```

【强调类】

这些强调类，通过颜色来表示强调。之前我们学习过强调标签 `small`、`strong`、`em`。现在学习强调类，`P` 元素加了这些类，会显示不同的颜色。

text-muted //muted 减弱的
<p class="text-muted">这些文字以#777 来显示</p>

text-warning
<p class="text-warning">这是警告色，橙灰色</p>

text-danger //危险的 红色
<p class="text-danger">这是错误提示</p>

text-info 蓝色
<p class="text-info">这只是信息</p>

text-success 绿色
<p class="text-success">这是正确的强调信息</p>

【缩略语】

在 html4 中有一个元素 **abbr** 用来指示简写或缩写，它使用 **title** 属性放完整的内容，当鼠标悬停在缩写词上时就会显示 **title** 的内容

<abbr title="这是完整文字内容">简写</abbr>

这是缩写

鼠标放上后显示完整文字

【地址】

bootstrap 中为<address>标签进行了样式的重写,让联系信息以最接近日常使用的格式呈现。在 html 中 address 标签是以斜体显示的,而在 bootstrap 让它以正常字体显示。

【表格】

默认样式

.table

想让你写的 table 标签应用 bootstrap 的样式,需要添加.table 类,这是 bs 表格的基本样式,它会添加很少的 padding,表格的水平线,另外把每个单元格平均分开。

.table-striped

它为表格提供了斑马线的样式,原理是在表格内,通过:nth-child CSS 选择器 (IE7-8 不支持)为表格中的行添加斑马纹样式。

.table-bordered

为表格增加边框(border)

.table-hover

为表格中的每一行赋予鼠标悬停样式。鼠标划过后会添加一个背景色。

.table-condensed

每个单元格的内补(padding)减半,可使表格更紧凑。将 padding 值 8 变为 5

【状态类】

通过这些状态类可以为行或单元格设置颜色。

.active 表示当前元素为活动状态

.success 标识成功或积极的动作
.info 标识普通的提示信息或动作
.warning 标识警告或需要用户注意
.danger 标识危险或潜在的带来负面影响的动作

【表单】

单独的 form 标签会被自动赋予一些全局的样式。所有在设置了.form-control 类里的<input>、<textarea>和<select>元素都被默认设置宽度为 100%，将 label 元素和 input、textarea、select 这些元素一起包裹在.form-group 中可以获得最好的排列。

只有加了 form-control 的表单元素才会应用 bootstrap 的样式（100%宽），最好 label 和 input 要用一个 form-group 来包裹。

【水平排列的表单】

通过为表单添加 .form-horizontal 类，并联合使用 Bootstrap 预置的栅格类，可以将 label 标签和控件组水平并排布局。

就是说，为 form 添加.form-horizontal 类后，它之下的.form-group 会被看做“row”，.form-group 中的 label 和 div 可以按栅格来划分。



```
<form class="form-horizontal">
  <div class="form-group">
    <label for="inputEmail3" class="col-sm-2 control-label">Email</label> //加了 control 会加一个上面的 padding
    <div class="col-sm-10"> //要加 div 然后用栅格分，因为加了 form-control 的宽度会是 100%
      <input type="email" class="form-control" id="inputEmail3" placeholder="Email">
    </div>
  </div>
</form>
```

```
</form>
```

【下拉列表（select）】

在原生的 select 中（如 Safari 和 Chrome），下拉列表的圆角是无法通过修改 border-radius 属性来改变的。而 bootstrap 自动为 select 添加了样式。

```
<select class="form-control">    //form 控件
    <option>1</option>
    <option>2</option>
    <option>3</option>
    <option>4</option>
    <option>5</option>
</select>
```

设置了.form-control 的<input>、<textarea>和<select>元素都将被默认设置为 width: 100%;

禁用状态

为输入框设置 disabled 属性可以禁止其与用户有任何交互（焦点、输入等）。被禁用的输入框颜色更浅，并且还添加了鼠标禁止点击的状态。

```
<input class="form-control" id="disabledInput" type="text" placeholder="Disabled input here..." disabled>
```

如果是我们自己来写，会有很多代码，而在 bootstrap 中我们只需要在元素中加 “disabled”

控件尺寸

input-lg input-sm 默认大小

【按钮】

按钮预定义样式

使用下面列出的类可以快速创建一个带有预定义样式的按钮。

```
<button type="button" class="btn btn-default">（默认样式）Default</button>
<button type="button" class="btn btn-primary">（首选项）Primary</button>
<button type="button" class="btn btn-success">（成功）Success</button>
<button type="button" class="btn btn-info">（一般信息）Info</button>
<button type="button" class="btn btn-warning">（警告）Warning</button>
<button type="button" class="btn btn-danger">（危险）Danger</button>
```

按钮的尺寸

btn-lg btn-sm btn-xs 默认

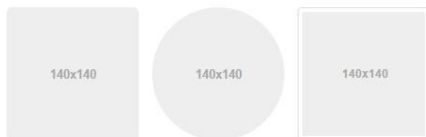
把按钮变为块级元素

通过给按钮添加 `.btn-block` 类可以将其拉伸至父元素 100%的宽度，而且按钮也变为了块级（block）元素。

激活状态

添加`.active` 类

【图片】



```
 //圆角  
 //圆  
 //双线
```

【关闭按钮】



如果我们想写一个关闭按钮，可能需要很多行代码，但是在 bootstrap 中为我们预定义好了，我们只要遵循以下格式来写。

```
<button type="button" class="close" ><span>&times;</span></button>
```

【三角符号】



通过使用三角符号可以指示某个元素具有下拉菜单的功能

```
<span class="caret"></span>
```

【元素位置】

通过添加一个类，可以将任意元素向左或向右浮动。

```
<div class="pull-left">...</div>
<div class="pull-right">...</div>
```

在排列导航条时，需要浮动的话，可以使用这些工具类：.navbar-left 或 .navbar-right

清除浮动

通过为父元素添加 **.clearfix** 类可以很容易地清除浮动（float）。

```
<div class="clearfix">...</div>
```

【让内容块居中】

为元素添加.center-block 类，可让元素居中显示，不论是行元素还是块元素。前提是要设置元素的宽。

```
<span class="center-block" style="width:200px;">这是 span</span>
<div class="center-block" style="width:200px;">这是 div</div>
```

【显示或隐藏内容】

.show 和 .hidden 类可以强制任意元素显示或隐藏。另外，.invisible 类可以被用来仅仅影响元素的可见性，这个元素仍然在文档流中。

```
<div class="show">...</div>
<div class="hidden">...</div>
<div class="invisible">...</div>
```


Bootstrap（二）组件

【Glyphicons 字体图标】

什么是字体图标？我们页面中有很多的图标，以前的作法是需要美工配合，给我们生成一张图片，然后把这个图放在我们需要的位置，而使用 bootstrap 后，我们可以使用它自带的一些图标。用法

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

它的意思是，span 作为图标的容器，而 glyphicon glyphicon-search 可以指向字体图标库中具体的某一个图标。

```
<button type="button" class="btn btn-default btn-lg">  
  <span class="glyphicon glyphicon-star"></span> 关注我  
</button>
```

【下拉菜单】



我们通常写下拉菜单是使用 select 元素，而在 bootstrap 中，我们用 button 和 ul 来模拟一个下拉菜单。

```
<div class="dropdown">
  <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown">
    点击触发下拉列表
    <span class="caret"></span>
  </button>
  <ul class="dropdown-menu" >
    <li><a href="#">Action</a></li>
    <li><a href="#">Another action</a></li>
    <li class="divider"></li>    //分割线
    <li class="disabled"><a href="#">Something else here</a></li>    //不可点击
    <li><a href="#">Separated link</a></li>
  </ul>
</div>
```

【导航】

一个基本的导航结构

```
<nav class="navbar navbar-default" >    //navbar 声明了导航样式，navbar-default 还可以是”反色”
  <div class="container-fluid">        //container-fluid 的 width 是 100%,而 container 的 width 是用媒体查询获得的动态尺寸

    </div>
</nav>
```

可为导航添加 navbar-fixed-top 或 navbar-fixed-bottom 将导航固定在顶部或底部

导航添加 LOGO

```
<div class="navbar-header">    //元素向左浮动
    <a href="#" class="navbar-brand">    //navbar-brand 展示了一个 logo
        
    </a>
</div>
```

导航列表

```
<ul class="nav navbar-nav">    //nav 和 navbar 的样式不同，nav 是专门为导航中的列表加样式，而 navbar 是为整个导航添加样式，navbar-nav 是让 Li 左浮
    <li class="active"><a href="#">首页</a></li>    //active 当前活动的元素
    <li><a href="#">首页</a></li>
    <li><a href="#">首页</a></li>
    <li><a href="#">首页</a></li>
    <li><a href="#">首页</a></li>
</ul>
```

为其中一项添加下拉样式

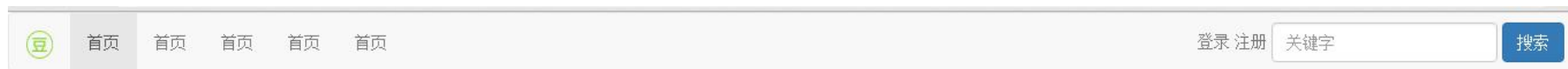
```
<li class="dropdown">
    <a href="#" class="dropdown-toggle"    data-toggle="dropdown">前端<span class="caret"></span></a>
    <ul class="dropdown-menu">
```

```
<li><a href="#">css</a></li>
<li><a href="#">css3</a></li>
<li class="divider"></li>
<li><a href="#">html</a></li>
</ul>
</li>
```

导航中的搜索框

```
<form action="" class="navbar-form navbar-right">
  <a href="#" class="navbar-link">登录</a>    //把链接置为灰色
  <a href="#" class="navbar-link">注册</a>
  <input type="text" class="form-control" placeholder="aaaaa"/>
  <button class="btn btn-primary">搜索</button>
</form>
```

完整导航示例



```
<nav class="navbar navbar-default" role="navigation">
  <div class="container-fluid">
    <!--logo-->
```

```
<div class="navbar-header">
  <a href="#" class="navbar-brand">
    
  </a>
</div>

<!-- 导航列表 -->
<ul class="nav navbar-nav">
  <li class="active"><a href="#">首页</a></li>
  <li><a href="#">首页</a></li>
  <li><a href="#">首页</a></li>
  <li><a href="#">首页</a></li>
  <li><a href="#">首页</a></li>
</ul>

<!-- 搜索框 -->
<div class="navbar-form navbar-right">
  <a href="#" class="navbar-link">登录</a>
  <a href="#" class="navbar-link">注册</a>
  <input type="text" name="" value="" class="form-control" placeholder="关键字" />
  <button class="btn btn-primary">搜索</button>
</div>
</div>
</nav>
```

【分页】



```
<div class="text-center">    //该元素居中对齐
  <ul class="pagination">    //定义了一个分页
    <li><a href="#">&laquo;</a></li>    //实体字符，可换任意 &lt; &gt;
    <li><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li class="active"><a href="#">3</a></li>    //当前为活动状态
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
    <li><a href="#">&raquo;</a></li>
  </ul>
</div>
```

【面包屑导航/路径导航】

A horizontal breadcrumb navigation bar with a light gray background. It contains the text "网站首页 / 前端学习 / bootstrap" in a blue font, where the slashes are also blue.

```
<ul class="breadcrumb">    定义一个面包屑导航
    <li><a href="#">前端学习</a></li>
    <li><a href="#" class="text-muted">第一阶段</a></li>    text-muted 强调类中的减弱样式
</ul>
```

【徽章】

A green rounded rectangular button with the text "未读信息" in white. To its right is a white circular badge with a green border containing the text "99+" in green.

```
<a href="#">Inbox <span class="badge">42</span></a>    //通过 badge 设置一个圆角的”徽章”

<button class="btn btn-primary" type="button">
    未读信息 <span class="badge">99+</span>
</button>
```

【缩略图】



```
<div class="thumbnail">      thumbnail 定义了缩略图样式
          缩略图的图片
    <div class="caption">      定义了标题的样式
        <h3><a href="#">bootstrap 课程</a></h3>
        <p>共 12 课</p>
    </div>
</div>
```


【列表组】

一个列表组包含了多个列表项

前端学习笔记--bootstrap实用教程

发布时间：2016-3-7

点击量：20

前端学习笔记--bootstrap实用教程
前端学习笔记--bootstrap实用教程
前端学习笔记--bootstrap实用教程
前端学习笔记--bootstrap实用教程
前端学习笔记--bootstrap实用教程

前端学习 bootstrap 教程

```
<div class="list-group" >
  <div class="list-group-item" style="border:0; border-bottom:1px dashed #ddd;">

    <h4><a href="#">前端学习笔记--bootstrap 教程</a></h4>
    <p class="text-muted">
      <span>发布时间： 2-16-5-17</span>
      <span class="pull-right">点击量： <span class="badge" style="background: red">20</span></span></p>
    </p>
    <p class="text-muted">这是文章内容这是文章内容这是文章内容</p>
    <p> //一组关键词徽章
      <span class="badge">前端</span>
      <span class="badge">web 前端</span>
      <span class="badge">前端教程</span>
    </p>
  </div>
```

</div>

【选项卡】

作品1	作品2
最新发布个人作品--尚学堂官网链接	2016-3-7
最新发布个人作品--尚学堂官网链接	2016-3-7
最新发布个人作品--尚学堂官网链接	2016-3-7
最新发布个人作品--尚学堂官网链接	2016-3-7
最新发布个人作品--尚学堂官网链接	2016-3-7
最新发布个人作品--尚学堂官网链接	2016-3-7

tab-pane 定义了每个选项卡面板，会将面板默认为隐藏的，点击后再显示

in 将元素显示出来

active 设置当前为显示状态

fade 是切换时的效果

完整选项卡

```
<ul class="nav nav-tabs">  //选项卡的头部
  <li class="active"><a href="#xw1" data-toggle="tab">新闻 1</a></li>
  <li><a href="#xw2" data-toggle="tab">新闻 2</a></li>
</ul>
```

```
<div class="tab-content"> //选项卡内容区的父级
  <div class="tab-pane fade in active" id="xw1"> // 单个的选项卡面板
    <a href="#">测试内容 aaaaaaaa</a>
  </div>
  <div class="tab-pane fade" id="xw2"> // 单个的选项卡面板
    <a href="#">测试内容 bbbbbbbb</a>
  </div>
</div>
```

【折叠后显示为按钮的导航】

浏览器窗口未缩放前



浏览器窗口缩放后



分为两部分，一部分是要在折叠后显示的内容，放在 **navbar-header** 类所在的 div 中。折叠后要隐藏的内容放在带有 **collapse navbar-collapse** 的 div 中。

折叠后要显示的按钮列表

```
<button class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
```

```
<span class="icon-bar"> </span>  
<span class="icon-bar"> </span>  
<span class="icon-bar"> </span>  
</button>
```

完整实例

```
<nav class="navbar navbar-default" >  
  <div class="navbar-header">  
    <button class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">  
      <span class="icon-bar"></span>  
      <span class="icon-bar"></span>  
      <span class="icon-bar"></span>  
    </button>  
  </div>  
  <div class="collapse navbar-collapse">  
    <div class="container-fluid">  
      <form action="" class="navbar-form navbar-right">  
        <a href="#" class="navbar-link">登录</a>  
        <a href="#" class="navbar-link">注册</a>  
        <input type="text" class="form-control" placeholder="aaaaa"/>  
        <button class="btn btn-primary">搜索</button>  
      </form>  
    </div>  
  </div>
```

`</nav>`

bootstrap 完整实例

[我的最新文章](#)

2016-7-12

BootStrap教程

发布时间：2016-7-12

点击量 **100**

[illegible]

关键词 bootstrap 响应式

[前端学习 / 课程概要](#)

实例代码

```
<!DOCTYPE html>
<html lang="zh-CN">
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- 上述 3 个 meta 标签*必须*放在最前面，任何其他内容都*必须*跟随其后！ -->
  <title>Bootstrap 101 Template</title>

  <!-- Bootstrap -->
  <link href="css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="css/main.css"/>

  <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and media queries -->
  <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->
  <!--[if lt IE 9]>
  <script src="//cdn.bootcss.com/html5shiv/3.7.2/html5shiv.min.js"></script>
  <script src="//cdn.bootcss.com/respond.js/1.4.2/respond.min.js"></script>
  <![endif]-->
</head>
<body>
```

```
<nav class="navbar navbar-inverse" role="navigation">
  <div class="container-fluid">

    <div class="navbar-header">
      <a href="#" class="navbar-brand">
        
      </a>
      <button class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
    </div>

    <div class="collapse navbar-collapse">
      <ul class="nav navbar-nav">
        <li><a href="#">首页</a></li>
        <li><a href="#">首页</a></li>
        <li class="active"><a href="#">首页</a></li>
        <li><a href="#">首页</a></li>
        <li class="dropdown">
          <a href="#" class="dropdown-toggle" data-toggle="dropdown">下拉列表
            <span class="caret"></span>
          </a>
          <ul class="dropdown-menu">
```



```
        <li><a href="#">列表项</a></li>
        <li><a href="#">列表项</a></li>
        <li class="divider"></li>
        <li><a href="#">列表项</a></li>
    </ul>
</li>
</ul>
<form action="" class="navbar-form navbar-right">
    <a href="#" class="navbar-link">登录</a>
    <a href="#" class="navbar-link">注册</a>
    <input type="text" class="form-control" placeholder="关键词"/>
    <input type="submit" class="btn btn-primary" value="搜索"/>
</form>
</div>
</div>
</nav>

<div class="container">
    <div class="row">
        <div class="col-md-6">
            <div class="dropdown">
                <button type="button" class="btn btn-default dropdown-toggle" data-toggle="dropdown">
                    点击触发下拉列表
```

```
<span class="caret"></span>
</button>
<ul class="dropdown-menu" >
  <li><a href="#">Action</a></li>
  <li><a href="#">Another action</a></li>
  <li class="divider"></li>
  <li class="disabled"><a href="#">Something else here</a></li>
  <li><a href="#">Separated link</a></li>
</ul>
</div>

</div>
<div class="col-md-6"></div>
</div>
<div class="row">
  <div class="col-md-4">
    <div class="text-center">
      <ul class="pagination">
        <li><a href="#">&laquo;</a></li>
        <li><a href="#">1</a></li>
        <li><a href="#">2</a></li>
        <li class="active"><a href="#">3</a></li>
        <li><a href="#">4</a></li>
        <li><a href="#">5</a></li>
        <li><a href="#">&raquo;</a></li>
      </ul>
```

```
</div>
</div>
<div class="col-md-4">
  <ul class="breadcrumb">
    <li><a href="#">网站首页</a></li>
    <li><a href="#">前端学习</a></li>
    <li><a href="#" class="text-muted">bootstrap</a></li>
  </ul>

  <span class="badge">20</span>
  <button class="btn btn-success">
    未读消息
    <span class="badge" style="background:red; color: #fff;">99+</span>
  </button>
</div>
<div class="col-md-4">
  <div class="thumbnail">
    
    <div class="caption">
      <h3><a href="#">框架课程</a></h3>
      <p>第二天的 bootstrap 课程</p>
    </div>
  </div>
</div>
</div>
```

```
</div>
<div class="row">
    <div class="col-md-6">

        <div>
            <ul class="nav nav-tabs">
                <li class="active"><a href="#xw1" data-toggle="tab">新闻一</a></li>
                <li><a href="#xw2" data-toggle="tab">新闻二</a></li>
            </ul>
            <div class="tab-content">
                <div class="tab-pane active in fade" id="xw1">
                    <p>
                        <a href="#">鲁大师最新官方版下载_百度软件中心</a>
                        <span class="pull-right">2016-3-7</span>
                    </p>
                    <p>
                        <a href="#">鲁大师最新官方版下载_百度软件中心</a>
                        <span class="pull-right">2016-3-7</span>
                    </p>
                    <p>
                        <a href="#">鲁大师最新官方版下载_百度软件中心</a>
                        <span class="pull-right">2016-3-7</span>
                    </p>
                    <p>
                        <a href="#">鲁大师最新官方版下载_百度软件中心</a>
```

```

        <span class="pull-right">2016-3-7</span>
    </p>
</div>
<div class="tab-pane fade" id="xw2">
    这是新闻二二二二二二
</div>
</div>
</div>
<div class="col-md-6">
    <div class="list-group">
        <div class="list-group-item no-border">
            <h3><a href="#">bootstrap 学习笔记</a></h3>
            <p class="text-muted"><span> 发 布 时 间 ： 2016-6-14</span> <span class="pull-right"> 点 击 量 ： <span class="badge badge-bg">20</span></span></p>
            <p class="text-muted">文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容</p>
            <p>
                <span class="badge">前端</span>
                <span class="badge">bootstrap</span>
                <span class="badge">北京尚学堂</span>
            </p>
        </div>
        <div class="list-group-item no-border">

```

```
<h3><a href="#">bootstrap 学习笔记</a></h3>
<p class="text-muted"><span> 发 布 时 间 ： 2016-6-14</span> <span class="pull-right"> 点 击 量 ： <span class="badge
badge-bg">20</span></span></p>
<p class="text-muted">文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容
文章内容文章内容文章内容文章内容文章内容文章内容</p>
<p>
  <span class="badge">前端</span>
  <span class="badge">bootstrap</span>
  <span class="badge">北京尚学堂</span>
</p>
</div>
<div class="list-group-item no-border">
  <h3><a href="#">bootstrap 学习笔记</a></h3>
  <p class="text-muted"><span> 发 布 时 间 ： 2016-6-14</span> <span class="pull-right"> 点 击 量 ： <span class="badge
badge-bg">20</span></span></p>
  <p class="text-muted">文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容文章内容
  文章内容文章内容文章内容文章内容文章内容文章内容</p>
  <p>
    <span class="badge">前端</span>
    <span class="badge">bootstrap</span>
    <span class="badge">北京尚学堂</span>
  </p>
</div>
</div>
```

```
        </div>
    </div>
</div>

<!-- jQuery (necessary for Bootstrap's JavaScript plugins) -->
<script src="http://apps.bdimg.com/libs/jquery/2.1.4/jquery.min.js"></script>
<!-- Include all compiled plugins (below), or include individual files as needed -->
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

常见分辨率

128*128 128*160 240*320（常见 2.0 2.2 2.4 2.8 寸屏手机使用）

320*240（常见 2.4 寸屏手机使用）

240*400（常见 3.0 寸屏手机使用）

320*480（主流屏常见 3.2 3.5 寸屏手机）

360*640（诺基亚常见）

480*800（常见 4.0 寸屏使用）

480*854（常见 3.7 寸屏使用）

960*540（QHD 常见 4.0 4.3 4.5 寸使用）

960*640（常见苹果机使用）

1136*640（常见苹果 5 使用）

1280*720（HD 主流屏常见 4.3 4.5 4.7 5.0 5.5 5.7 6.0 寸屏使用）

1280*768（常见 4.5 寸屏使用）

1280*800（常见三星 5.3 寸及平板电脑使用）

1024*600（常见 7.0 寸屏平板电脑使用）

1024*768（常见 LG5.0 寸屏及 7.9 寸 9.7 寸平板电脑使用）

1920*1080（FHD 主流屏常见 4.7 5.0 5.0 5.7 6.0 寸手机及电视使用）

1920*1280（常见 9.0 寸平板电脑使用）

1920*1200（常见 10.1 寸平板电脑及酷派大神 7.0 使用）

2048*1539（常见苹果 iPad 及国产一些平板使用）

2560×1440（2k 屏）

4096×2160（4K 屏常见电视使用）