# HTML+CSS基础

# 一、HTML概述

## 1.1 HTML无处不在

学习，游戏，生活，新闻，交友，购物，互联网时代只要是你叫的出来名字的东西，基本都能找到与之对应的网站。网站的类型多种多样。并且手机，电脑，平板，电视... 无处不在，而构成这些网页的东西，就是我们所要学习的东西，那我们先从能看到的部分开始，它就是 ：超文本标记语言（简称：HTML）。开发人员（web前端开发工程师）要和浏览器沟通，就要用到浏览器所能识别的语言(HTML超文本标记语言),所以HTML他是一门浏览器能够识别的语言。

## 1.2 HTML如何构建一个网页？

### 1.2.1 认识HTML的解构

一个网页的基本构成，可以分为三个部分，**结构（Tag）**，**样式（CSS）**，**行为（JavaScript）**。如果把HTML构建一个网页比喻成，建一栋大楼：

1、使用Tag标签搭建基本结构（设计楼层布局，房间布局等，构建大楼的基本骨架结构）

2、使用CSS样式进行装饰（贴瓷砖，抹白，过道，楼梯装修等，把基本骨架进行进一步的装饰）

3、使用JS控制网页行为（安装电梯，路灯，等功能性的设施）

也就是说：

1. 负责内容的HTML(超文本标记语言，骨架)
2. 负责外观的CSS (层叠样式表)/css3（衣服）
3. 负责行为的JavaScript(js/jquery应用)（行为动作）

HTML（HyperText Markup Language ）是一种由标签组成的**超文本标记语言**，而非编程语言, **一个html文档就是一个网页**。

### 1.2.2 HTML的优点

1、简易性，HTML版本升级采用超集方式，从而更加灵活方便。 　　  
2、[可扩展性](https://www.baidu.com/s?wd=%E5%8F%AF%E6%89%A9%E5%B1%95%E6%80%A7&from=1012015a&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d-rARsujn3nARknh7-Pj0s0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWm3Pjn1PHbd)，[HTML语言](https://www.baidu.com/s?wd=HTML%E8%AF%AD%E8%A8%80&from=1012015a&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1d-rARsujn3nARknh7-Pj0s0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWm3Pjn1PHbd)的广泛应用带来了加强功能，增加标识符等要求，HTML采取子类元素的方式，为系统扩展带来保证。  
3、平台无关性。虽然PC机大行其道，但使用MAC等其他机器的大有人在，HTML可以使用在广泛的平台上.

### 1.2.3开发工具

Editplus、sublime、webstrom ...

工具只是工具，不同的工具不会对代码产生任何的影响，用记事本都能写代码。

### 1.2.4 浏览器

Chrome（推荐使用），Firefox，IE ...

### 1.2.5快捷键

1、刷新：f5（刷新前一定要先在开发工具里保存一下）

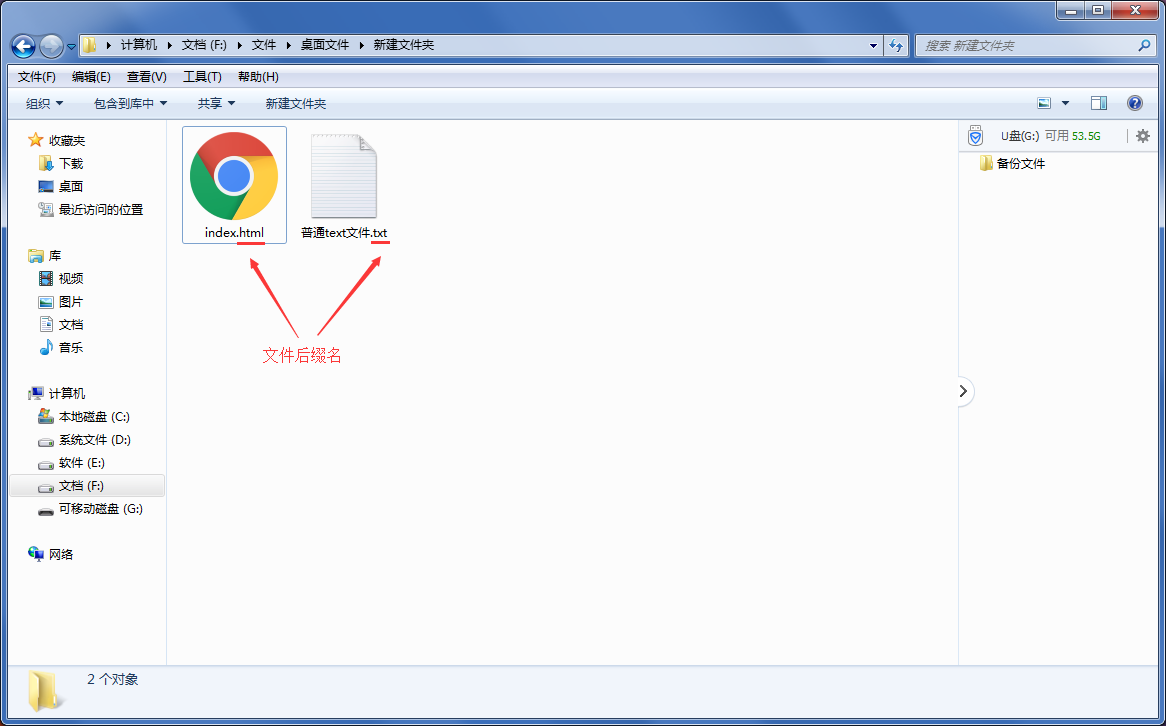
2、调试工具：f12

3、快速查看元素：ctrl+shift+c

**PS：**工具的下载与设置，通过百度都可以找到，今后的工作学习中肯定会重复的安装与设置，所以一定要亲自动手尝试安装与设置。

### 1.2.6 HTML的文件格式

HTML文件能在任何浏览器浏览，其文件格式后缀名为 **.html** 类似普通的记事本文件后缀名为 .txt



## 1.3 HTML标签

### 1.3.1 HTML基本结构

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>Document</title>  
</head>  
<body>  
  
</body>  
</html>

### 1.3.2 html标签

此元素可告知浏览器其自身是一个 HTML 文档。

<html> 与 </html> 标签限定了文档的开始点和结束点，在它们之间是文档的头部和主体。正如您所了解的那样，文档的头部由[**<head> 标签**](http://www.w3school.com.cn/tags/tag_head.asp)定义，而主体由 [**<body> 标签**](http://www.w3school.com.cn/tags/tag_body.asp)定义。

### 1.3.3 head标签

<head> 标签用于定义文档的头部，它是所有头部元素的容器。<head> 中的元素可以引用脚本、指示浏览器在哪里找到样式表、提供元信息等等。文档的头部描述了文档的各种属性和信息，包括文档的标题、在 Web 中的位置以及和其他文档的关系等。绝大多数文档头部包含的数据都不会真正作为内容显示给读者。

下面这些标签可用在 head 部分：[**<base>**](http://www.w3school.com.cn/tags/tag_base.asp), [**<link>**](http://www.w3school.com.cn/tags/tag_link.asp), [**<meta>**](http://www.w3school.com.cn/tags/tag_meta.asp), [**<script>**](http://www.w3school.com.cn/tags/tag_script.asp), [**<style>**](http://www.w3school.com.cn/tags/tag_style.asp), 以及 [**<title>**](http://www.w3school.com.cn/tags/tag_title.asp)。

<title> 定义文档的标题，它是 head 部分中唯一必需的元素。

### 1.3.4 meta标签

<meta> 元素可提供有关页面的元信息（meta-information），比如针对搜索引擎和更新频度的描述和关键词。

<meta> 标签位于文档的头部，不包含任何内容。<meta> 标签的属性定义了与文档相关联的名称/值对。

### 1.3.5 body标签

body 元素**定义文档的主体**。

body 元素**包含文档的所有内容**（比如文本、超链接、图像、表格和列表等等。），直接能够在页面中展示的部分。

# 二、基本结构标签

## 2.1 标签相关

### 2.1.1 HTML标签

* HTML 标记标签通常被称为 HTML 标签 (HTML tag)。
* html标签由尖括号组成的关键词,比如<html>
* html标签成对出现,第一个开始(开放)标签,第二个结束(闭合)标签。比如<head></head>标签必须闭合。
* html单标签加”/“闭合。如：<meta />，也可以不加，如：<meta>。

### 2.1.3 标签属性

标签的属性，能赋予标签一些作用。格式为：**属性名=”属性值”** 的格式写在开始标签内，同时一个标签可以有多个标签属性，如下 meta 标签的，charset、name、content，html标签的lang，都是标签属性，在这里meta标签通过不同的标签属性起到了不同的作用：

<!doctype html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport"  
 content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>Document</title>  
</head>  
<body>  
  
</body>  
</html>

### 2.1.4 注释

注释的作用类似于手机说明书，与主要内容的解析没有任何关系，注释中的任何内容也不会直接在浏览器中呈现，毕竟手机说明书写的再好，对手机的性能也没有任何一点影响。注释不会被浏览器解析。

##### html注释

<!--我是HTML注释-->

##### css注释

* 单行注释：//我是注释
* 多行注释：/\*我是注释\*/

## 2.2 基本结构标签

构建一个网页的基本结构也需要大量的标签，好比建一栋大楼需要大量的材料，这些标签没有太多特殊的作用，所以接下来就是去认识这些经常用来搭建结构的基本标。

### 2.2.1 p：段落标签

1. 表示文章或文本的一个段落。
2. 独占一行，支持宽高。

eg:

<p>我是p标签</p>

### 2.2.2 h1~h6：标题标签

1. 字体依次减小，权重依次降低。独占一行。
2. h标签有且只有6个，并且，一般情况下h1标签在网页中只允许出现一次。
3. 独占一行，支持宽高。

eg:

<h1>h1标题</h1>  
<h2>h2标题</h2>  
...  
<h6></h6>

### 2.2.3 div：盒子标签

1. <div> 可定义文档中的分区或节（division/section）。<div> 标签可以把文档分割为独立的、不同的部分。它可以用作严格的组织工具，并且不使用任何格式与其关联。如果用 id 或 class 来标记 <div>，那么该标签的作用会变得更加有效。
2. 独占一行，支持宽高。

eg:

<div>我是div标签</div>

### 2.2.4 span：普通文本标签

1. 请使用 <span> 来组合行内元素，以便通过样式来格式化它们。
2. span 没有固定的格式表现。当对它应用样式时，它才会产生视觉上的变化。
3. 一行展示，不支持宽高。

<span>我是span标签</span>

### 2.2.5 em与 i：斜体字体

1. em倾斜字体-轻微强调。
2. 倾斜字体-无强调语义-现常用于图标。
3. 一行展示，不支持宽高。

eg:

<em></em>  
<i></i>

### 2.2.6 strong 与 b：加粗字体

1. Strong加粗字体-强烈强调。
2. b加粗字体-无强调语义。
3. 一行展示，不支持宽高。

eg:

<strong > </strong >  
<b> </b>

### 2.2.7 ul：无序列表

#### 2.2.7.1 语法

1. 定义：无序列表是一个项目的列表，此列项目使用粗体圆点进行标记。无序列表始于 <ul> 标签 , 每个列表项始于 <li>。
2. 用途：通常用于导航，菜单，列表等布局
3. 结构：无序列表包含一个列表区域（ul）与数个列表项（li）
4. Ul和li都是独占一行。

eg:

<ul>  
 <li>1</li>  
 <li>2</li>  
</ul>

#### 2.2.7.2属性

* type="disc" 实心圆 <ul type="disc" ></ul>
* type="circle" 空心圆 <ul type=" circle " ></ul>
* type="square" 方块符 <ul type=" square " ></ul>

清除项目符号：

ul{list-style: none;}

#### 2.2.7.3嵌套

1. 只能在li里嵌套标签、图片、链接。
2. 标签不能交叉闭合。

eg:

<ul>  
 <li><a href="">我是a标签</a></li>  
 <li>  
 <ul>  
 <li></li>  
 </ul>  
 </li>  
</ul>

错误的写法：

<ul>  
 <li><a href="">我是a标签</li></a>  
 <p>我是p标签</p>  
</ul>

### 2.2.8 ol：有序列表

#### 2.2.8.1语法

1. 定义：有序列表也是一列项目，列表项目使用数字进行标记。有序列表始于 <ol> 标签。每个列表项始于 <li> 标签。
2. 用途：使用较少，可用于目录等。
3. ol和li都是独占一行。

eg:

<ol>  
 <li>1</li>  
 <li>2</li>  
</ol>

#### 2.2.8.2 属性

type：列表项样式

A / a：字母（大写 / 小写）

<ol type=”A”><li></li></ol>

I：罗马数字

<ol type=”I”><li></li></o--l>

#### 2.2.8.3嵌套

和无序列表规则一样

#### 2.2.9自定义列表

1. 语义：定义列表不仅仅是一列项目，而是项目及其注释的组合。定义列表以 <dl> 标签开始。每个定义列表项以 <dt> 开始。 每个定义列表项的定义以 <dd> 开始。
2. 用途：常用于上下（左右）的图文区域布局，标题-内容区块

<dl>  
 <dt></dt>  
 <dd></dd>  
</dl>

## 2.3功能标签

<br> <!--换行-->  
<hr> <!--水平线-->

### 2.3.1 a标签

#### 2.3.1.1基本属性

* <a> 标签定义超链接，用于从一张页面链接到另一张页面。
* <a> 元素最重要的属性是 href 属性，它指示链接的目标。
* target属性：\_self（在本窗口打开）
* target属性：\_blank（在空白窗口打开）
* target属性：\_new（在新窗口打开）

eg:

<a href="http://www.baidu.com" target="\_new">百度一下</a>

#### 2.3.1.2锚点

链接到同一个页面的不同的位置。

<a href="#daniel">点击跳转</a>  
<a href="" name="daniel">Daniel</a>

等价于：

<a href=**"#daniel"**>**点击跳转**</a>  
<div id=**"daniel"**></div>

#### 2.3.1.3 base

设置所有a标签的初始属性。

<base href="http://www.baidu.com" target="\_blank">

<a href="">百度一下</a>

这时候即使a没有设置地址，在点击的时候也会进行跳转

### 2.3.2 img标签

1. img 元素向网页中嵌入一幅图像。
2. 请注意，从技术上讲，<img> 标签并不会在网页中插入图像，而是从网页上链接图像。<img> 标签创建的是被引用图像的占位空间。
3. <img> 标签有两个必需的属性：[**src 属性**](http://www.w3school.com.cn/tags/att_img_src.asp) 和 [**alt 属性**](http://www.w3school.com.cn/tags/att_img_alt.asp)。

#### 2.3.2.1常用的图片格式

图片 = 图片名称 + 扩展名

扩展名常用的有三种：

* .jpg 常用格式
* .gif 常用格式 小动画图片（由多张图片快速循环播放）
* .png 常用格式 透明图片 （背景由灰白格子交错形成，但在页面中不会显示出来）

#### 2.3.2.2 属性

* src: <img src =”url” />（url：图片路径）
* width： <img width=“图片的宽度”>
* height： <img height=“图片的高度”>
* alt: <img alt=“图片加载失败显示的文字”>
* title: <img title=“鼠标放在图片上显示的文字”>

#### 2.3.2.3 三种路径

##### 1. 绝对路径：

从盘符开始的路径，如：

<img src=“D:\progfiles\web\images\2.jpg”>

PS：从自己的磁盘开始，在别人的电脑上路径就不一样了。因此，只能在自己的电脑上访问，一般不常用。

##### 2. 从域名出发：

从域名出发，只要有网就可以访问，如：

<img src=“http://bbs.moon.cn/static/image/common/logo.png”>

##### 3. 相对路径：

从当前HTML文档路径开始的路径（常用），如：

<img scr=“images/1.jpg”>

* 路径符号: / 下一级目录
* ../ 返回上一层目录
* ../../ 返加上一层的上一层目录

### 2.3.3 iframe 内联框架

#### 2.3.3.1概念

* 定义：iframe 元素会创建包含另外一个文档的内联框架（即行内框架）。
* 兼容：所有浏览器都支持 <iframe> 标签。

eg:

<iframe src=**"http://www.baidu.com"** frameborder=**"0"** scrolling=**"no"**></iframe>

#### 2.3.3.2 属性

|  |  |  |
| --- | --- | --- |
| **属性** | **值** | **描述** |
| [align](http://www.w3school.com.cn/tags/att_iframe_align.asp) | **left**  **right**  **top**  **middle**  **bottom** | **不赞成使用。请使用样式代替。**  **规定如何根据周围的元素来对齐此框架。** |
| [frameborder](http://www.w3school.com.cn/tags/att_iframe_frameborder.asp) | **1**  **0** | **规定是否显示框架周围的边框。** |
| [height](http://www.w3school.com.cn/tags/att_iframe_height.asp) | **pixels**  **%** | **规定 iframe 的高度。** |
| [longdesc](http://www.w3school.com.cn/tags/att_iframe_longdesc.asp) | **URL** | **规定一个页面，该页面包含了有关 iframe 的较长描述。** |
| [marginheight](http://www.w3school.com.cn/tags/att_iframe_marginheight.asp) | **pixels** | **定义 iframe 的顶部和底部的边距。** |
| [marginwidth](http://www.w3school.com.cn/tags/att_iframe_marginwidth.asp) | **pixels** | **定义 iframe 的左侧和右侧的边距。** |
| [name](http://www.w3school.com.cn/tags/att_iframe_name.asp) | **frame\_name** | **规定 iframe 的名称。** |
| [sandbox](http://www.w3school.com.cn/tags/att_iframe_sandbox.asp) | **""**  **allow-forms**  **allow-same-origin**  **allow-scripts**  **allow-top-navigation** | **启用一系列对 <iframe> 中内容的额外限制。** |
| [scrolling](http://www.w3school.com.cn/tags/att_iframe_scrolling.asp) | **yes**  **no**  **auto** | **规定是否在 iframe 中显示滚动条。** |
| [seamless](http://www.w3school.com.cn/tags/att_iframe_seamless.asp) | **seamless** | **规定 <iframe> 看上去像是包含文档的一部分。** |
| [src](http://www.w3school.com.cn/tags/att_iframe_src.asp) | **URL** | **规定在 iframe 中显示的文档的 URL。** |
| [srcdoc](http://www.w3school.com.cn/tags/att_iframe_srcdoc.asp) | **HTML\_code** | **规定在 <iframe> 中显示的页面的 HTML 内容。** |
| [width](http://www.w3school.com.cn/tags/att_iframe_width.asp) | **pixels**  **%** | **定义 iframe 的宽度。** |

#### 2.3.3.3 Link

* <link> 标签定义文档与外部资源的关系。
* <link> 标签最常见的用途是链接样式表。

## 2.4 简单的认识标签类型

### 2.4.1 标签类型的分类

所有的标签都有一个默认的display属性，标签按照这个属性值分为，**块级标签**（display: block），**行级**(display: inline)，**行块级**(display: inline-block)三种。

### 2.4.2 特征

#### 2.4.2.1 块级标签

* 排列特性：单独占据一行的位置
* 宽高特性：可以设置宽高，高度由内容或者字体行高撑开，宽度默认为auto
* 边距特性：支持设置内外边距，并且正常生效
* 默认为：display: block;
* 常见的块级标签：<div>、<p>、<ul>、<li>、<h>、<dl>、<dt>、<dd>....

#### 2.4.2.2 行级标签

* 排列特性：可以和其他标签在同一行排列
* 宽高特性：不支持设置宽高、宽高均由内容撑开。
* 换行间距：标签之间的换行会被解析为空格
* 默认为：display: inline;
* 常用的行级标签：<a>、<span>、<strong>、<em>、<i>、<b>....

#### 2.4.2.3 行块级标签

* 排列特性：可以和其他标签在同一行排列，一排无法放下的元素将在下一行开始排列
* 宽高特性：可以设置宽高，宽高均默认由内容或者字体行高撑开
* 换行间距：标签之间的换行会被解析为空格
* 默认值：display:inline-block;
* 常见的行块级标签：<img>**、**<input>**、**<select>**...**

#### 2.4.2.4 vertical-align

行级，行块级在垂直方向的对齐方式会受到vertical-align属性的影响。

<img src=**"timg.jpg"** alt=**""** width=**"200"**>  
<span>**我是span标签**</span>

我们可以发现图片和文字默认是没有办法底部对齐的，那么我们可以通过vertical-align使上中下对齐

* top 上对齐
* middle 居中对齐
* bottom 默认，底部对齐

**注意：**要给对齐的元素都加上这个属性，且只适用于行级和行块级标签。

## 2.5 标签的嵌套

### 2.5.1 标签正确的嵌套

嵌套是双标签才能有的结构关系，也就是把内容放到双标签的开始与结束标签之间，这里内容可以是标签，也可以是文字，但是遵循一定的**嵌套规范**，嵌套的结构在浏览器中的表现是**大的模块包含小的模块**。同样用建筑来类比的话，就是**大楼 > 楼层 > 套房 > 房间**。

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

或者

<div>  
 <p></p>  
 <ul>  
 <li></li>  
 </ul>  
</div>

### 2.5.2 嵌套注意点

##### 1. 标签不能交叉嵌套

如：

<div>  
 <p><a href=**""**></p></a>  
</div>

##### 2. p标签不能嵌套其它标签

如：

<p><span></span></p>

##### 3. 行级标签不能嵌套块级标签，a标签除外

<span><div></div></span>

# 三、CSS样式的基本使用

## 3.1 样式概述

### 3.1.1 什么是样式？

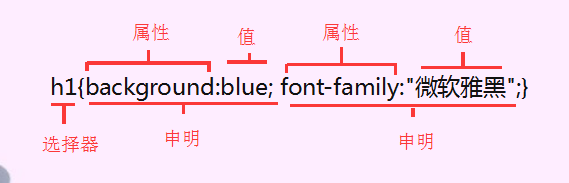
##### css（Cascading Style Sheets）：层叠样式表

* 对于元素（可视标签）进行装饰，如背景色，边框线，字体等等。
* 对元素的排版控制，如元素的宽高，元素之间的距离，位置等等。

### 3.1.2书写相关

#### 3.1.2.1 样式的书写包含以下结构：

* 定义：CSS 规则由两个主要的部分构成：选择器，以及一条或多条声明。
* 选择器：找到HTML结构中，需要被添加样式的元素
* 声明：具体要添加修改的样式，由属性，冒号和属性值和分号组成。

如：

div{  
 width: 100**px**; /\*元素内容宽度\*/  
 height: 100**px**; /\*元素内容高度\*/  
 background: **deeppink**; /\*元素背景\*/  
 color: **red**; /\*元素字体颜色\*/  
 font-size: 20**px**; /\*元素字体大小\*/  
}

#### 3.1.2.2 颜色的值得写法：

* 英文单词 如：red、green、deeppink
* 十六进制 如：#ffffff（#fff）、#000000（#000）十六进制颜色都是六位，如果六位的值都一样，可简写为三位
* rgb 如：rgb(0,0,0) 其中三个值的范围是0-255
* rgba 如：rgb(0,0,0,0.5) 最后一个参数（0.5）是透明度

## 3.2 样式的类型

### 3.2.1 行内样式

写在元素行内的style属性的属性值里，不利于代码优化。

<div style=**"**width: 100**px**;height: 100**px**; background: **deeppink"**></div>

#### 3.2.2 内联样式（内部样式）

写在head标签内的style标签内

<style>  
 div{  
 width: 100**px**;  
 height: 100**px**;  
 background: **deeppink**;  
 }  
</style>

#### 3.2.3 外联样式（外部样式)

写在单独的.css后缀的文件中，通过link标签引入。引入通常在最后一个meta标签后面，在大型项目经常使用。

<link rel="stylesheet" href="index.css">

# 四、选择器

## 4.1 选择器概述

css选择器，能对HTML结构中的一个或者一类元素通过选择器属性进行标记，然后可以通过这些标记找到对应的元素，为其添加样式

### 4.1.1 选择器的使用场景

HTML的CSS样式书写过程中，类名占据了绝大部分，因为具有复用的特性，而且使用非常灵活，建议优先使用。其次是标签名，适用于一个范围内的同一元素需要统一添加相同的样式的时候，比如列表标签。最后ID不建议在布局之中主要使用，因为其唯一的特性有很大的局限性，通常用于一个大的模块的最顶层使用一次。

## 4.2 选择器的结构关系

除了直接使用选择器选择标签之外，我们还可以通过标签所处的结构不同来进行元素选择的范围划分。也就是选择器的，包含关系，同级关系等等。

### 4.2.1 标签名选择器

通过标签名选择元素

<style>  
 p{  
 width: 100**px**;  
 }  
</style>

eg:

<body>  
 <p></p>  
</body>

### 4.2.2 类名选择器

通过类名选择元素，语法：”.” + 类名

<style>  
 .box{  
 width: 100**px**;  
 }  
</style>

<body>  
 <div class=**"box"**></div>  
</body>

### 4.2.3 ID名选择器

通过ID名选择元素，语法：”#” + ID名

<style>

#box{  
 width: 100**px**;  
 }  
</style>

eg:

<body>  
 <div id=**"box"**></div>  
</body>

### 4.2.4 通配符选择器

通常用来样式初始化

\*{ margin: 0; padding: 0;}

### 4.2.5后代选择器

父级元素 + 空格 + 子元素

<style>  
 #box p{  
 width: 100**px**;  
 }  
</style>

eg:

<body>  
 <div id=**"box"**>

<p></p>

</div>  
</body>

### 4.2.6 包含选择器（后代选择器）

**匹配规则：**包含选择器选择到所有的后代元素，无论是否是直接的子级，选择器之间以空格连接。

#### 选择器：ul li{ list-style: none; }

父级元素 + 空格 + 子元素 选中ul所有的li后代

<style>  
 ul li{  
 width: 100**px**;  
 }  
</style>

<body>  
 <ul>

<li></li>

</ul>  
</body>

### 4.2.7 子级选择器

子级选择器只会选择直接的子级元素，不会选择到所有的后代，选择器之间用大于符号（>）连接

#### 选择器：div > span{ color: red; }

只会选中div的亲儿子第一个span，而不会选中p里面的span

<div>  
 <span>1</span>  
 <p><span>2</span></p>  
</div>

### 4.2.8 群组选择器

群组选择器可以添加多组选择器，然后为这些选择器所找到的元素，添加相同的样式，多组选择器之间以逗号（,）隔开。

#### 4.2.3.1 语法

选择器：.nav , #box { width: 100px; height: 100px;}

同时选中box和nav

<div id=**"box"**></div>  
<div class=**"nav"**></div>

#### 4.2.3.2 注意：

.box1 ul li,.box2 ul li{ width: 100px; height: 100px;}

是同时选中box1和box2里面的li,但是**不能 .box1,.box2 ul li{ width: 100px; height: 100px;}**

<body>  
<div class=**"box1"**>  
 <ul>  
 <li></li>  
 </ul>  
</div>  
<div class=**"box2"**>  
 <ul>  
 <li></li>  
 </ul>  
</div>  
</body>

### 4.2.9 复合条件选择器

复合条件选择器又至少两个选择器同时选择，只有当元素满足所有的条件的时候才能被选择，比如类名是box的div标签，或者id名是wrap的p标签等等，缺一不可，多个选择器之间不需要任何符号的连接

##### 语法

div.box{ width: 100px; height: 100px; }

选中类名为box的div标签

<div class=**"box"**></div>

<p class=**"box"**></p>

### 4.2.10 毗邻选择器

毗邻选择器，找到接下来的第一个对应的亲兄弟元素，选择器之间用 + 号连接

##### 选择器：.box+div{ width: 100px; height: 100px; }

<div class=**"box"**></div>  
<div></div>

## 4.3 伪类选择器

伪类大多是**以选择器为依托来表现**的，是一种**特殊的选择器**（伪类选择器），或者**元素**（伪类元素）

### 4.3.1 伪类选择器

伪类选择器，依旧是一个选择器。跟普通选择器不同，它需要元素满足某些特殊条件的时候才能选择到元素，类似于给选择器加了一个开关。这些开关包括，鼠标停留在元素上（：hover），鼠标在元素上按下的时候（：active）等等：

##### 元素未被点击时

a:link{ color:**blue**; }

##### 2．鼠标在元素上悬停时

a:hover{ color:**purple**; }

##### 3. 鼠标在元素上按下时

a:active{ color:**yellow**; }

##### 4. 元素被点击过后

a:visited{ color:**green**; }

### 4.3.2 :hover伪类相关

hover 伪类可以不止使用于a标签，鼠标移动到div上，div的背景变成粉红色

div:hover{ background: **deeppink**; }

鼠标移动到div上，div的后代p的背景变成粉红色

div:hover p{ background: **deeppink**; }

鼠标移动到div上，div的亲兄弟p的背景变成粉红色

div:hover+p{ background: **deeppink**; }

### 4.3.3 伪类元素

伪类元素，与普通的标签元素类似，但是并不是直接写在body之中，而是依托于选择器添加，并且这些元素有特殊的特性

##### 1.元素前面插入内容

a::before{ content:'插入前面的文字'; }

##### 2.元素后面插入内容

a::after{ content:'插入后面的文字'; }

##### 3.特性：before 与 after的一些特性

* content 为必须的属性且格式必须完整 content:'';
* 其解析出的效果基本等同于写入一个标签，因此，可以设置宽高，但默认是行级，需要转换为块级。
* { }中的所有属性都是为伪类元素添加不影响使用伪类的元素
* after常用于清除浮动时使用

## 4.4 选择器的优先级

### 4.1.1 概述

当同一个元素被多个选择器同时选择，赋予不同属性的时候是可以同时实现的，但是赋予相同的属性时我们应该以谁的值为准？如下代码最终的背景颜色是？

<style>  
 div{  
 width: 100**px**;  
 height: 100**px**;  
 background: #ccc;  
 }  
 #box{  
 background: **deeppink**;  
 }  
 .c1{  
 background: **green**;  
 }  
</style>

<body>  
 <div id=**"box"** class=**"c1"**></div>  
</body>

因此，每个选择器的优先级是不相同的，而通过不同的选择器组合形成的选择器的优先级又是不同的，那么我们怎么去判定呢？接下来我们就来学习选择器的优先级！

### 4.1.2 影响优先级的因素

上边的案例，最终的颜色是由id选择器添加的蓝色，原因在于id选择器的优先级要高于其他选择器，这是选择器类别上造成的优先级差异。不仅如此，实际上影响选择器优先级的因素还有很多。

#### 4.1.2.1选择器类别

##### 1、类选择器 > 元素选择器

<style>  
 div{  
 width: 100**px**;  
 height: 100**px**;  
 background: #ccc;  
 }  
 .c1{  
 background: **green**;  
 }  
</style>

<body>  
 <div id=**"box"** class=**"c1"**></div>  
</body>

##### 2、id选择器 > 类选择器

<style>  
 #box{  
 background: **deeppink**;  
 }  
 .c1{  
 background: **green**;  
 }  
</style>

<body>  
 <div id=**"box"** class=**"c1"**></div>  
</body>

##### 3、元素选择器 > 通配符选择器

\*{  
 margin: 0  
}  
div{  
 width: 100**px**;  
 height: 100**px**;  
 margin: 100**px**;  
}

##### 4、然后通配符选择器又能清除标签的默认样式

\* {margin: 0;padding: 0;}

最终我们可以得出结论：**ID > class > 标签名 > 通配符 > 默认样式**

#### 5.1.2.1选择器数量

##### 1、ID个数不相等

ID个数多的优先级高，后面的不用比

#box1 #box2{ background: **deeppink**; }  
#box2{ background: **yellow**;}

因此，背景颜色为粉红色。

##### 2、ID选择器个数相等

则看class个数，class个数越多，优先级越高。

#box3 .box4 li{background: **purple**;}  
#box3 li{background: **blue**;}

因此，背景颜色为紫色。

##### 3、class 个数相等

再看tagName 个数，tagName个数越多优先级越高。

#box5 .box6 span{color:**green**;}  
#box5 .box6 p span{color:**deeppink**;}

因此，背景颜色为粉红色。

**PS:** 不分先后顺序，只看选择器类型和个数。

#### 5.1.2.3 执行顺序

由于HTML的代码是由上往下执行的，因此，当优先级相同时，后面的会覆盖前面的，同一个选择器下的相同属性也遵循该规则。

#### 5.1.2.4 样式的位置

1. 行内 > 内部 > 外部

2. 内部和外部的优先级又看选择器的优先级

3. 选择器优先级相同又看谁先加载。（看link标签引入的先后顺序）

#### 5.1.2.5 其他规则

!important > 行内 > 其他

**优先级总结：**选择器描述的越清晰范围越小优先级越高，执行顺序越靠后优先级越高

# 五、盒子模型

### 7.1 盒子模型概述

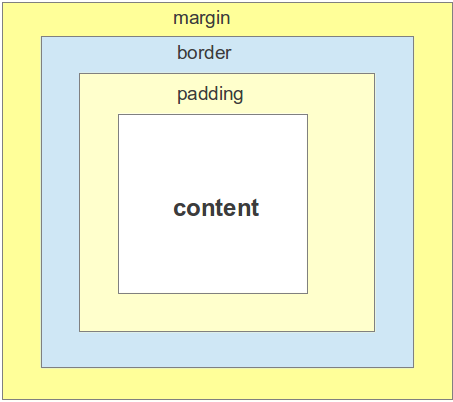
网页中构成布局的元素，可以想象成很多个放在地上的盒子，他们之间的位置关系，和自身特点有如下:

1. 这些盒子大小各异，
2. 盒子之间间隔有大有小，
3. 大盒子里放小盒子，小盒子里放更小的盒子，小盒子可以放在大盒子中间，也可以放在大盒子边缘
4. 盒子的厚度有厚有薄



所以实际上总的来说，网页元素在组成一个结构的时候最主要的东西，就是**元素内容区域的大小**，元素内容与元素边框的间隙，**元素的边框**宽度，**元素之间的距离**，也就是如下的四个样式：

1. content--元素的宽高(width/height) 元素内容区域的大小
2. padding--元素的内边距(padding) 元素内容区域和边框之间的间隙
3. border--元素的边框(border) 元素的边框
4. margin--元素的外边距(margin) 元素边框以外和其他元素之间的间隔



### 7.2 content内容区域

此区域是元素的实际放内容的区域，是一个包含宽度与高度的矩形区域

#### 4.3.2.1宽与高（width / height）

属性：

width：元素的宽度

height：元素的高度

值：

固定的数值：100px（px像素）

p{ width:100px; }

百分比：10%（基于父级的）

p{ height:10%; }

### 4.3.3 padding填充区域

填充区域是内容区域与边框之间的区域，一般用于元素的内容与元素边缘之间的距离设置

#### 4.3.3.1属性：

* padding-top：元素顶部（上）与内部元素的距离
* padding-right：元素右侧（右）与内部元素的距离
* padding-bottom：元素底部（下）与内部元素的距离
* padding-left：元素左侧（左）与内部元素的距离

##### 值：

固定的数值：100px（px像素）

p{padding-left:100px;}

百分比：10%（基于父级的宽度）

p{ padding-top:10%; }

##### 复合样式：

* padding:10px;（一个值） 上下左右--10px
* padding:10px 20px;（两个值） 上下--10px 左右--20px
* padding:10px 20px 30px;（三个值） 上--10px 左右--20px 下--30px
* padding:10px 20px 30px 40px;（四个值） 上--10px 右--20px 下--30px 左--40px （顺时针匹配）

### 4.3.4 border边框线区域

边框是元素的边缘，边框可以设置宽度，颜色，以及边框的风格

#### 4.3.4.1 border-width：边框的宽度 值：像素单位

<style>  
 p{  
 border-width: 1**px**;  
 }  
</style>

#### 4.3.4.2 border-style: 边框的样式

##### 值：

* none : 无边框（默认值）
* solid：实线
* dashed：虚线
* dotted：点状线
* double：双实线
* groove ：3D凹槽
* ridge：3D立槽
* inset：3D嵌入边框1
* outset：3D嵌入边框2

#### 4.3.4.3 border-color: 边框的颜色

<style>  
 p{  
 border-color: red;  
 }  
</style>

#### 4.3.4.4复合样式：

##### 所有边框的单个属性：

* border-width:10px;（一个值）
* border-width:10px 20px;（两个值）
* border-width:10px 20px 30px;（三个值）
* border-width:10px 20px 30px 40px;（四个值）
* border-color: red;（一个值）
* ......
* border-style: solid;（一个值）
* .....

##### 所有边框线的所有属性：

border:1px solid red;

##### 单条边框线的所有属性：

* border-left: 1px solid red;
* border-right: 1px solid red;
* border-top: 1px solid red;
* border-bottom: 1px solid red;

##### 单条边框线的单个属性：

* border-left-width:10px;
* border-left-style:solid;
* border-left-color:red;
* ......

### 4.3.5 margin外边距区域

外边距区域是盒子模型最外层的区域，主要用于不同的盒子之间的间距设置

#### 4.3.5.1 单个属性：

* margin-top：元素顶部（上）与外部元素的距离
* margin-right：元素右侧（右）与外部元素的距离
* margin-bottom：元素底部（下）与外部元素的距离
* margin-left：元素左侧（左）与外部元素的距离

##### 值：

固定的数值：100px（px像素） p{margin-left:100px;}

百分比：10%（基于父级的宽度） p{ margin-top:10%; }

关键字：auto（自适应，可以使元素左右居中显示） p{ margin:0 auto; }

#### 4.3.5.2 复合样式：

* margin:10px;（一个值） 上下左右--10px
* margin:10px 20px;（两个值） 上下--10px 左右--20px
* margin:10px 20px 30px;（三个值） 上--10px 左右--20px 下--30px
* margin:10px 20px 30px 40px;（四个值） 上--10px 右--20px 下--30px 左--40px （顺时针匹配）

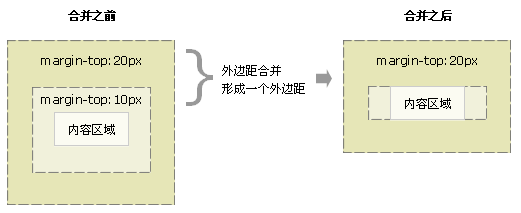
### 4.3.6 边距的合并与溢出

外边距在使用的时候有如下的一些特殊情况需要注意

#### 4.3.6.1内包含外边距合并

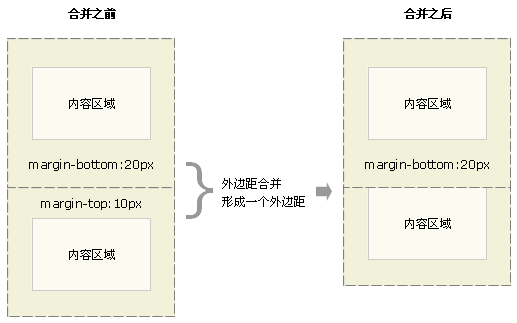
1. 当父子关系的两个元素，子级拥有上下外边距时，其外边距不会作用于父级，而是会传递给父级

2. 当父元素和子元素都有上下外边距时，合并后的外边距会取其中较大者



#### 4.3.6.2上下外边距的合并

当上下相邻的两个元素，在相邻的面同时拥有外边距时，其外边距值只会取其中较大的值



# 六、清除默认样式

## 5.1为什么要清除默认样式？

1. 避免重复定义他们，提高样式代码的重用。
2. 设置自己的风格，字体，颜色等。
3. 减少代码->减少维护成本
4. 可以提供完全空白的画布，然后你就可以自己定义相应的样式了。
5. 开发更加有逻辑性：你只是添加样式而不是移除和修改。
6. 浏览器的兼容性问题可以降到最小。（每个浏览器默认样式不一样，比如行高，某个浏览器是1，另外一个浏览器可能是1.1，这样在开发网页写样式表的时候，处理起来比较麻烦，所以直接给重置统一，这样就能很方便的处理兼容性。）

## 5.2需要清除的默认样式

##### 1. 以下标签有默认外边距

body,dl,dd,dt,p,h1,h2,h3,h4,h5,h6{ margin:0; }

##### 2. 以下标签有内外边距，有项目符号

ol,ul{ margin:0;padding:0;list-style:none; }

##### 3. a标签去除下划线和默认字体颜色

a {text-decoration:none; color: inherit;}

##### 4. 直接使用通配符去掉所有标签的内外边距

\*{ margin:0;padding:0; }

# 七、元素类型转换

我们可以发现并不是所有的标签我们都能为其添加宽度高度等等一些特定样式。因为实际上通过这些标签的特性，除了宽高特性之外，在界面上默认的排列方式，内外边距的作用情况等等的特性我们可以把标签大致分为**行级**、**块级**、**行块级**。

元素类型(display)：行级元素(display:inline) 块级元素(display:block) 行块级元素(display: inline-block)

## 7.1 行级标签与块级标签的特性

### 7.1.1 行级标签的特性

#### 7.1.1.1特性

* 排列特性：可以和其他标签在同一行排列，一排无法放下的元素将在下一行开始排列
* 宽高特性：不支持设置宽高、宽高均由内容撑开。
* 边距特性：左右外边距生效，上下外边距不生效。上下内可以设置生效，但是无法作用于其他元素。margin:auto失效
* 换行间距：标签之间的换行会被解析为空格
* 默认为：display: inline;
* 常用的行级标签：<a>、<span>、<strong>、<em>、<i>、<b>....

### 7.1.2 块级标签

#### 7.1.2.1 特性

* 排列特性：单独占据一行的位置
* 宽高特性：可以设置宽高，高度由内容或者字体行高撑开，宽度默认为auto
* 边距特性：支持设置内外边距，并且正常生效
* 默认为：display: block;
* 常见的块级标签：<div>、<p>、<ul>、<li>、<h>、<dl>、<dt>、<dd>....

### 7.2.3 行块级标签

#### 7.2.3.1 特性

* 排列特性：可以和其他标签在同一行排列，一排无法放下的元素将在下一行开始排列
* 宽高特性：可以设置宽高，宽高均默认由内容或者字体行高撑开
* 边距特性：支持设置内外边距，并且正常生效，margin:auto失效
* 换行间距：标签之间的换行会被解析为空格
* 对其方式：行块元素的位置会受到text-align文本对其方式的影响
* 默认值：display:inline-block;
* 常见的行块级标签：<img>**、**<input>**、**<select>**...**

### 7.2.4 text-align文本对齐方式

##### 值：

* left 默认，左对齐
* right 右对齐
* center 居中

text-align 对行级，行块级也有用 最常用就是让img图片居中对齐 注意：要给父级这个属性

## 7.2 元素类型转换

通过CSS中的display属性，我们可以把元素的特性进行转换，比如将span标签转换成块级标签的特性，那么这个span标签的特性将与普通的块级标签没有区别，但是需要注意的是这时候的span标签依旧是一个行内标签。

### 7.2.1 常用的特性

##### 1、块级特性：display:block;

a{ display: **block**; }

##### 2、行内特性：display:inline;

div{ display: **inline**; }

##### 3、行块级特性：display:inline-blcok;

div{ display: **inline-block** }

##### 4、不显示元素：display:none;

隐藏元素，并且隐藏的元素在页面中不占位置。

div{ display: **none;** }

### 7.2.2 行级、行块级元素的垂直对齐方式

通过vertical-align 属性我们能规定，行内元素以及文字或者行内块元素在垂直方向的对其方式

#### 值：

1. baseline：默认值。元素以基线对齐
2. top：元素以顶线对齐
3. middle: 居中对齐
4. bottom：元素以底线对齐

如：

span{  
 background: **deeppink**;  
 vertical-align: **middle**;  
}  
img{ vertical-align: **middle**; }

<span>**我是span标签**</span>  
<img src=**" images/1.jpg "** alt=**""** width=**"200"**>

这个时候span标签和img标签在垂直方向就能按照指定的方式对齐，要注意的是。要以什么方式对齐，两个标签都要设置这个属性，且值也要一样。

### 7.2.3 text-align属性

#### 1、值

1. text-align: left; 设置文本或者行级、行块级元素左对齐显示
2. text-align: center; 设置文本或者行级、行块级元素居中对齐显示
3. text-align: right; 设置文本或者行级、行块级元素右对齐显示
4. text-align: justify; 设置文本两端对齐显示

PS：在设置行级、行块级元素对齐方式的时候需要给他的父级元素这个属性

# 八、background背景样式

背景显示在元素最底层的属性，不会影响到元素内的子级或者文字的正常排列，背景通常包含背景颜色，和背景图片两种形式。

### 8.1.1 背景颜色（background-color）

规定要使用的背景颜色

p{ background-color: **deeppink**; }

### 8.1.2 背景图片（background-image）

规定要使用的背景图像。PS：背景图片只会在设置这个背景图片的盒子里显示，超出的是不会被显示的

p{ background-image: url("images/1.jpg"); }

### 8.1.3 背景图片重复方式（background-repeat）

规定如何重复背景图像。

p{ background-repeat: **repeat**; } /\*平铺\*/  
p{ background-repeat: **no-repeat**; } /\*不平铺\*/  
p{ background-repeat: **repeat-x**; } /\*x轴平铺\*/  
p{ background-repeat: **repeat-y**; } /\*y轴平铺\*/

### 8.1.4 背景位置（background-position）

规定背景图像的位置。默认0,0，相对于盒子元素。背景定位的值包含两个参数分别对应为x轴y轴的坐标值之间用空格隔开。

##### 值：

* 关键字：center、top、right、bottom、left
* 固定的数值、百分比（计算公式： (元素的宽度 – 图片的宽度) \* 10%）

p{ background-position: **left top**; } /\*左上\*/

p{ background-position: **100px 100px**; }

p{ background-position: **10% 10%**; } /\*左上\*/

注意：如果我们只写一个值的时候

1. 如果使用的是px或者百分比，这个值默认为x轴的值，y默认为center

2. 如果使用的是关键字，要看这个关键字是那一个轴的，另外一个轴的值默认为center

### 8.1.6背景大小（background-size）

规定背景图片的尺寸。背景大小的值包含两个参数分别对应为宽、高之间用空格隔开

##### 关键字：设置时只需一个参数

cover：等比缩放铺满整个元素

contain：等比缩放到遇到第一条元素边缘停止缩放

p{ background-size: **cover**; }  
p{ background-size: **contain**; }

##### 固定的数值、百分比

p{ background-size: 200**px** 200px; }

p{ background-size: 200**px**; }

p{ background-size: 200**px** auto; }

百分比是相对于盒子元素。

p{ background-size: 10% 10%; }

注：当只写一个值时该值为x轴的值，y轴的值默认等比缩放（auto）

### 8.1.7 背景依附（background-attachment）

规定背景图像是否固定或者随着页面的其余部分滚动

p{ background-attachment: **scroll**; } /\*随界面的滚动而移动（默认值）\*/

此时，背景图片此时的定位0,0就是是相对于可视化窗口进行定位的。

p{ background-attachment: **fixed**; } /\*不会随界面的滚动而移动\*/

### 8.1.8复合属性

p{ background:deeppink url(“images/1.jpg”) no-repeat center top/100px 100% fixe; /\*background: 颜色 图片 是否平铺 定位/背景尺寸 是否滚动\*/

# 九、浮动

## 9.1 浮动概述

### 9.1.1、定义：

使元素**脱离文档流**，按照**指定方向**（左右）发生移动，遇到**父边界**或者**相邻的浮动元素**停了下来。

##### 文档流：

是文档中可显示对象在排列时所占用的位置/空间(在页面中占位置)。

##### 脱离文档流：

在页面中不占位置 ，CSS 中通过 float 属性实现元素的浮动

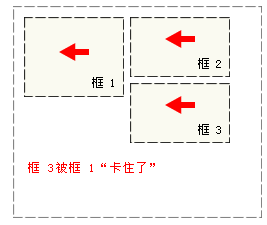
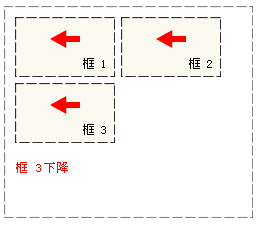
#### 9.1.2、属性、值：

p{ float: **left**; } /\*左浮动\*/  
p{ float: **right**; } /\*右浮动\*/  
p{ float: **none**; } /\*不浮动\*/

## 9.2 浮动的特性

### 9.2.1 排列规则

1. 第一个浮动元素遇到父级边缘的时候停止浮动。
2. 其它元素在横向遇见相邻浮动元素的时候停止
3. 当父元素宽度不够时在下一排排列。但是，在换行的过程中有元素高度比**换行前相邻元素**的高的时候，元素就会被“卡”住。



### 9.2.2 宽高特性

a、无论什么类型的元素，只要浮动就支持设置宽高，但是当未设置宽高时由内容撑开

b、支持margin，但不支持 margin:auto;

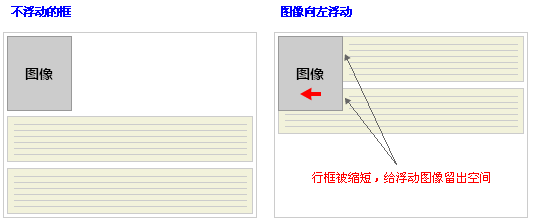
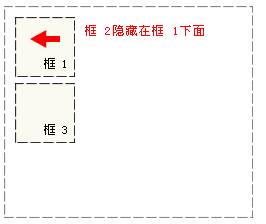
c、可以解决垂直外边距合并问题

### 9.2.3 层级特性

a、浮动元素之后未浮动元素会在浮动元素的下方显示，但是浮动元素之前的未浮动元素不会被影响，因为浮动元素只能对后面的元素产生影响。

b、文字不会出现在浮动的元素的下方而是围绕浮动元素显示，而块级元素却会在下方显示，但是要注意的是，行级、行块级不会。

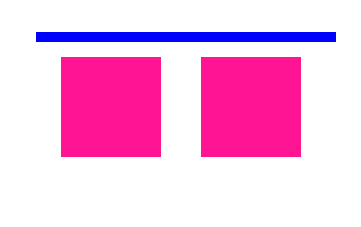
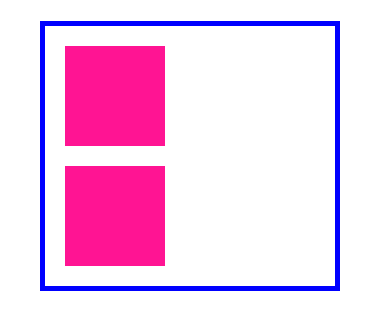
c、层级提升半层。



## 9.3 清除浮动

### 9.3.1 什么是清除浮动

清除浮动是指解决如，父级高度塌陷不能占据当前位置等等的一些问题，而不是说让浮动失效。由于浮动元素脱离文档流，因此，造成父级坍塌。



### 9.4.2 清楚浮动的方法

#### 1、紧邻兄弟法

给紧邻兄弟加 clear属性，值（left、right、both）

缺点：平白无故的多了一个盒子元素。

#### 2、父级元素法

1、加高 扩展性不好

2、display：inline-block margin：auto失效，有间隙

3、overflow：hidden 比如你需要滚动条，或者有绝对定位的子元素在外边。（原理不需要知道）

#### 3、after伪类（时下主流）

好比给此元素添加了一个空标签，然后使用了浮动属性清除浮动。像这种常用的我们通常可以写在一个类里面，在需要用到的时候可以直接使用类就可以了。

.clearfix:after{  
 content: '';  
 display: **block**;  
 clear: **both**;  
}

# 十、定位

## 10.1 什么是定位？

定位属性（position）允许元素相对于**其自身所在的位置**、**父级定位元素**、**浏览器窗口**（取决于属性值）进行移动。

## 10.2 定位的偏移量属性

### 10.2.1 定位偏移量的作用

元素添加定位属性与属性值后，只会出现在默认的位置，如果想要移动它，还需要添加与定位配合使用的**偏移量属性**。

### 10.2.2 属性与值

#### 属性：

* left：元素左边与参照物左边的距离
* right：元素右边与参照物右边的距离
* top：元素上边与参照物上边的距离
* bottom：元素下边与参照物下边的距离

#### 值：

* 固定数值：div{ position:relative; left:100px; top:100px; }
* 百分比：div{ position:absolute; right:50%; bottom:50%; } 相对于父级盒子宽高

PS：一般同轴向（X、Y）的偏移量属性通常只会出现一个。

## 10.3 定位的属性与值

### 12.3.1相对定位：relative

#### 定义：

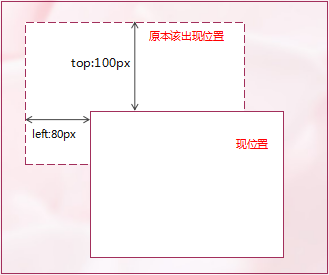
相对于最初应该出现的位置定位。

#### 特点：

元素框最初位置仍然保留,**不脱离文档流**（阻碍布局，因此，一般不用其定位，但不给偏移量不影响元素）

#### 语法：

div{ position:**relative**;}



### 12.3.2绝对定位：absolute

#### 定义：

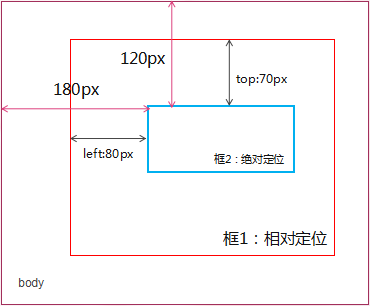
相对于相对于**最近的父级定位元素**定位，如没有定位父级元素，则相对**body**定位（注意：在使用bottom的时候是相对于可视化窗口进行定位）。使用偏移量后会相对定位父级移动，**脱离文档流**。

#### 特点：

1. 使元素脱离文档流
2. 所有类型元素使用绝对定位就支持宽高
3. 支持margin，但是margin：auto失效

#### 写法：

position:**absolute**;  
left:100**px**;  
top:100**px**;



注意:一般我们用relative作为父级定位元素，用absolute定位

注意：元素设置定位（除了相对定位）之后块级元素宽度不再默认为父级的百分百。

### 12.3.3固定定位：fixed

#### 定义：

相对于**可视化窗口**（浏览器中我们能看见的区域）定位，**脱离文档流**

#### 特点：

元素不会随浏览器滚动而滚动，**脱离文档流**

#### 写法：

position:**fixed**;  
left:100**px**;  
top:100**px**;

## 10.4 定位的层级

### 10.4.1 层级概念

定位元素与定位元素之间有时候会遇到相互层叠的情况，通过层级我们可以决定他们之间层叠的顺序。

### 10.4.2 层级规则

1、默认情况下html结构中，后边的定位元素层级会比前边的层级高

2、通过层级属性的设置值越大层级越高，就显示在上面，没有上限，也没有下限。（在计算机不溢出情况下，计算机溢出就会以计算器最大值为准），值只能为整数

3层级相同，后面加载的元素会在上层。

position:**absolute**;  
left:0;  
top:0;  
z-index:10;

注意：只有使用的定位的元素（除了相对定位）层级属性才有意义

## 10.5 定位技巧

### 10.5.1万能居中

#### 方法1：

我们都知道，定位使margin：auto失效，但是如果在x轴left和right值都为0的时候，在x轴方向就可以实现margin:auto，

如果y轴方向top和bottom都为0则可以实现y轴方向绝对居中。

注意的是：要和margin:auto;搭配使用

div{  
 position:**absolute**;  
 width:100**px**;  
 height:100**px**;  
 left:0;  
 top:0;  
 right:0;  
 bottom:0;  
 margin:**auto**;  
}

#### 方法2：

我们也可以先给left和top值50%，然后用margin拉回自身宽高的一半，也可以实现绝对居中。

div{  
 position:**absolute**;  
 width:100**px**;  
 height:100**px**;  
 left:50%;  
 top:50%;  
 margin-left:-50**px**;  
 margin-top:-50**px**;;  
}

#### 方法3：

和方法2的原理一样，只是我们可以简写，如下：

div{  
 position:**absolute**;  
 width:100**px**;  
 height:100**px**;  
 left: calc(50% - 50**px**);  
 top: calc(50% - 50**px**);  
}

# 十一、常用CSS2，CSS3样式

## 11.1 CSS常用样式

### 11.1.1 内容超出处理

属性：overflow

##### 值：

* hidden 溢出隐藏
* auto 溢出出现滚动条
* scroll 一直出现滚动条
* visible 超出显示（默认）

##### 不同溢出轴向处理：

* x轴溢出时（overflow-x ）
* y轴溢出时（overflow-y ）

### 11.1.2 字体样式(font)

#### 11.1.2.1 字体颜色（color）

##### 颜色的值得写法：

1. 英文单词 如：red、green、deeppink
2. 十六进制 如：#ffffff（#fff）、#000000（#000）十六进制颜色都是六位，如果六位的值都一样，可简写为三位
3. rgb 如：rgb(0,0,0) 其中三个值的范围是0-255
4. rgba 如：rgb(0,0,0,0.5) 最后一个参数（0.5）是透明度

#### 11.1.2.2 字体大小（font-size）

##### 单位：

1. px 单位： 可以通过px我单位设置字体大小，浏览器默认字体大小是16px，最小12px，最大无上限。
2. em 单位： 默认字体的大小（16px）的倍数，如 2em = 32px
3. rem 单位： html字体大小的倍数，默认的html字体大小也是16px

#### 11.1.2.3 字体风格（font-style）

##### 值：

1. normal：默认
2. italic：斜体字体
3. oblique：倾斜字体（和前者的区别与em和i标签的区别一样）

#### 11.1.2.4 字体类型（font-family）

##### font-family 规定元素的字体系列

p{ font-family: '华文行楷'; }

font-family 可以把多个字体名称作为一个“回退”系统来保存。如果浏览器不支持第一个字体，则会尝试下一个。也就是说，font-family 属性的值是用于某个元素的字体族名称或/及类族名称的一个优先表。浏览器会使用它可识别的第一个值。如：‘华文行楷’'微软雅黑'; 如果系统没有华文行楷，则会选择微软雅黑，一次往后推，如果到最后一个都没有，则会选择浏览器默认字体类型

p{ font-family: '华文行楷','华文中宋','微软雅黑'; }

#### 11.1.2.5 字体粗细（font-weight）

##### 值：

* bold: 加粗
* normal： 不加粗
* 数字：100-900 （700 = bold）

#### 11.1.2.6 字体行高（line-height）

line-height 属性设置行间的距离（行高），不允许使用负值。

##### 值：

* px 22px（具体的数值）
* 数值： 1.5（基于字体大小的倍数）、
* 注：字体行高可以影响到元素的默认高度（行内元素除外,对行内元素没有作用）

#### 11.1.2.7复合样式

p{ font: 20px '华文行楷'; }  
p{ font: 20px/100px '华文行楷'; }  
p{ font: bold 20px/100px '华文行楷'; }  
p{ font: bold italic 20px/100px '华文行楷'; }

注意：属性值可以缺省，但是最少要保留 **字体大小** 与 **字体类型**

#### 11.1.2.8 @font-face

导入字体，其中font-family是对字体的命名，src是字体文件的路径

@font-face {  
 font-family: "xiaowei";  
 src: url("../font/小微logo体.otf");  
}  
@font-face {  
 font-family: "kuhei";  
 src: url("../font/站酷酷黑.ttf");  
}

font-family: "xiaowei";

### 11.1.3 文本样式

文本样式是针对文字一些排列相关的样式，如字体间距，对齐方式等等。

#### 11.1.3.1文本水平对齐方式（text-align）

##### 值：

* left（左对齐，默认值）
* right（右对齐）
* center（居中对齐）
* 注意：还可使行级、行块级元素对齐，但一定要给 其父元素。

#### 11.1.3.2 文本修饰（text-decoration）

##### 值：

1. none（无修饰）
2. underline（下划线）
3. overline（上划线）
4. line-through（删除线）

#### 11.1.3.3 文本首行缩进（text-indent）

##### 单位：

* px text-indent: 20px; 允许使用负值
* em text-indent: 2em; 1em等于一个字的宽度

#### 11.1.3.4文本溢出处理（text-overflow）

##### 值：

* clip（超出裁剪）
* ellipsis（超出出现省略号）

##### 单行超出省略号：

overflow:**hidden**; /\*超出隐藏\*/  
text-overflow:**ellipsis**; /\*文本超出用省略号代替\*/  
white-space:**nowrap**; /\*不允许换行\*/

#### 11.1.3.5 文本字母处理（text-transform）

##### 值：

p{ text-transform: **none**;} /\*正常显示\*/  
p{ text-transform: **capitalize**;} /\*单词首字母大写\*/  
p{ text-transform: **uppercase**;} /\*单词全部大写\*/  
p{ text-transform: **lowercase**;} /\*单词全部小写\*/

#### 11.1.3.6 字间距（letter-spacing）

p{ letter-spacing: 10**px**; }

#### 11.1.3.7 词间距（word-spacing）

有空格就算作一个词

p{ word-spacing: 10**px**; }

#### 11.1.3.8 文本换行方式（white-space）

##### 值：

1. nowrap（只有遇到<br />才会换行）
2. normal（默认换行方式）
3. pre （保留空格、缩进、换行，但是文本不会自动换行）
4. pre-line （合并空格、缩进、换行，文本自动换行）
5. pre-wrap（保留空格、缩进、换行、并且文本自动换行）

#### 11.1.3.9 单词换行方式（word-break）

##### 值：

1. break-all（允许在单词内换行）
2. normal（默认换行方式）
3. keep-all( 不允 许在单词内换行 )

### 11.1.4 字体转义

html界面中有一些特殊符号不可以直接显示在文档当中如< > ，因为会被浏览器当做源码解析掉 ...... 这时候需要对它们进行转义才能在界面中正常的显示，方式是通过使用该字符的编码进行转义之后再界面中显示该字符，但是又不会被浏览器识别为特殊字符

#### 11.1.4.1 部分字符转义对照表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **字符** | **十进制** | **转义字符** | **字符** | **十进制** | **转义字符** | **字符** | **十进制** | **转义字符** |
| > | &#60; | &gt; | Á | &#193; | &Aacute; | á | &#225; | &aacute; |
| < | &#62; | &lt; | Â | &#194; | &circ; | â | &#226 | &acirc; |
| ￡ | &#163; | &pound; | Ã | &#195; | &Atilde; | ã | &#227; | &atilde; |
| ¤ | &#164; | &curren; | Ä | &#196; | &Auml | ä | &#228; | &auml; |
| ￥ | &#165; | &yen; | Å | &#197; | &ring; | å | &#229; | &aring; |
| | | &#166; | &brvbar; | Æ | &#198; | &AElig; | æ | &#230; | &aelig; |
| § | &#167; | &sect; | Ç | &#199; | &Ccedil; | ç | &#231; | &ccedil; |
| ¨ | &#168; | &uml; | È | &#200; | &Egrave; | è | &#232; | &egrave; |
| © | &#169; | &copy; | É | &#201; | &Eacute; | é | &#233; | &eacute; |
| a | &#170; | &ordf; | Ê | &#202; | &Ecirc; | ê | &#234; | &ecirc; |
| ? | &#171; | &laquo; | Ë | &#203; | &Euml; | ë | &#235; | &euml; |
| ? | &#172; | &not; | Ì | &#204; | &Igrave; | ì | &#236; | &igrave; |
| /x7f | &#173; | &shy; | Í | &#205; | &Iacute; | í | &#237; | &iacute; |
| ® | &#174; | &reg; | Î | &#206; | &Icirc; | î | &#238; | &icirc; |
| ˉ | &#175; | &macr; | Ï | &#207; | &Iuml; | ï | &#239; | &iuml; |
| ° | &#176; | &deg; | Ð | &#208; | &ETH; | ð | &#240; | &ieth; |
| ± | &#177; | &plusmn; | Ñ | &#209; | &Ntilde; | ñ | &#241; | &ntilde; |
| 2 | &#178; | &sup2; | Ò | &#210; | &Ograve; | ò | &#242; | &ograve; |
| 3 | &#179; | &sup3; | Ó | &#211; | &Oacute; | ó | &#243; | &oacute; |
| ′ | &#180; | &acute; | Ô | &#212; | &Ocirc; | ô | &#244; | &ocirc; |
| μ | &#181; | &micro; | Õ | &#213; | &Otilde; | õ | &#245; | &otilde; |
| ? | &#182; | &para; | Ö | &#214; | &Ouml; | ö | &#246; | &ouml; |
| · | &#183; | &middot; | &times; | &#215; | &times; | ÷ | &#247; | &divide; |
| ? | &#184; | &cedil; | Ø | &#216; | &Oslash; | ø | &#248; | &oslash; |
| 1 | &#185; | &sup1; | Ù | &#217; | &Ugrave; | ù | &#249; | &ugrave; |
| o | &#186; | &ordm; | Ú | &#218; | &Uacute; | ú | &#250; | &uacute; |
| ? | &#187; | &raquo; | Û | &#219; | &Ucirc; | û | &#251; | &ucirc; |
| ? | &#188; | &frac14; | Ü | &#220; | &Uuml; | ü | &#252; | &uuml; |
| ? | &#189; | &frac12; | Ý | &#221; | &Yacute; | ý | &#253; | &yacute; |
| ? | &#190; | &frac34; | Þ | &#222; | &THORN; | þ | &#254; | &thorn; |
| ? | &#191; | &iquest; | ß | &#223; | &szlig; | ÿ | &#255; | &yuml; |
| À | &#192; | &Agrave; | à | &#224; | &agrave; |  |  |  |

### 11.1.5 cursor鼠标指针

规定鼠标放置到元素上时显示的指针样式

**属性：**cursor

##### 值：

1. auto 默认。浏览器设置的光标。
2. pointer 光标呈现为指示链接的指针（一只手）
3. move 此光标指示某对象可被移动。
4. text 光标指示文本。
5. wait 光标指示正忙（通常是一只表或沙漏）
6. help 光标指示帮助（通常是一个问号或一个气球）

### 11.1.6 元素透明度

设置元素的透明程度，可以从完全透明到完全透明之间

#### 11.1.6.1 主流浏览器（谷歌、火狐、IE8以上）下的透明度

**属性：**opacity

##### 值：

0-1 0位完全透明，1为完全不透明

p{ opacity: 0.5; }

#### 11.1.6.2 IE8及以下版本的透明度

**属性：**filter

##### 值：

0-100 0完全透明，100完全不透明

p{ filter: alpha(**opacity**=50); }

**注意：**元素的透明度，作用于这个元素以及其里面的子级元素和文本内容等等，而rgba()背景透明只是一个颜色值，只是刚好这个颜色带有透明的属性，对于元素的子级文本等等，没有任何的影响

### 11.1.7 样式的继承

某些特定的css样式会被其后代元素继承，从而间接的继承生效，这种样式大部分为**字体文本样式**

#### 11.1.7.1部分常见继承样式

color、font、text-align、list-style、word-spacing、......

#### 11.1.7.2 继承注意点

* a标签由于自身拥有字体颜色，所以无法继承字体颜色
* 继承属性的优先级较低

#### 11.1.7.3 强制继承（inherit）

我们可以通过给样式设置一个通用值（inherit），让元素强制继承父级的该属性，哪怕这个属性是默认不能被继承的，列如：

* height: inherit;
* background-color: inherit;

## 11.2 常用css3

### 11.2.1 兼容性

1、部分css3须加兼容前缀：

-webkit- 常用于兼容chrome浏览器，

-moz- 常用于兼容火狐，

-o- 常用于兼容opera，

-ms- 常用于兼容IE。

2、兼容性查询网站：http://caniuse.com/

### 11.2.2多行文本超出显示省略号

.main p:nth-child(2){  
 display: **-webkit-box**; /\*继承block的属性\*/  
 -webkit-box-orient:**vertical**; /\*元素垂直显示\*/  
 -webkit-line-clamp:2; /\*设置文本显示的行数\*/  
 overflow: **hidden**; /\*（不能使用padding）\*/  
}

### 11.2.3 渐变色

#### 11.2.3.1线性渐变 linear-gradient()

渐变方向：left、right、top、bottom可单独，也可两两组合，注意，没有center

.main div:nth-of-type(15){ background: -webkit-linear-gradient(**left**,**deeppink**,**yellow**,#153170);}

.main div:nth-of-type(16){ background: -webkit-linear-gradient(**left**,**deeppink** 10%,**yellow** 60%,#153170);}

使用百分比的时候表示：从10%以前都是粉红色的没有渐变，而10%-60%就会和黄色搭配产生渐变，60%以后就是黄色域蓝色搭配产生渐变。

#### 11.2.3.2 径向渐变 radial-gradient()

渐变方向：right、left、top、bottom两两组合，可center，但必须单独使用。

.main div:nth-of-type(17){ background: -webkit-radial-gradient(**center**,**deeppink**,**yellow**,#153170);}

### 11.2.4 border-radius圆角

值：px %（相对于自身的宽高）

#### 11.2.4.1 圆角复合属性

.box{

*width*: 100px;

*height*: 100px;

*background*: hotpink;

*border-radius*: 10px;

*border-radius*: 10px 20px;

*border-radius*: 10px 20px 30px;

*border-radius*: 10px 10px 20px 30px;

}

#### 11.2.4.2 圆与椭圆

当元素为正方形的时候，可以实现圆：

##### 使用px实现圆

border-radius: 元素的半径;

##### 使用百分比实现圆

border-radius:50%;

当元素为长方形的时候，只有百分比能够实现椭圆：

##### 使用px

##### 使用百分比



注意：当你只需要给小的圆角的时候通常用px值，如果要圆或者椭圆的时候通常使用50%

### 11.2.5 transition过渡

#### 11.2.5.1 单个属性

1. transition-property：要过渡的属性名称（如width、height）
2. transition-duration：过渡效果持续时间
3. transition-delay：延迟过渡时间（可不写）
4. transition-timing-function: 过渡效果运行曲线

##### Transition-timing-function：运动曲线

1. linear: 匀速
2. ease：慢快慢
3. ease-in：匀加速
4. ease-out：匀减速

#### 11.2.5.2 复合属性

transition: 2s; /\* 过渡时间 \*/

transition: width 2s; /\* 过渡属性 过渡时间 \*/

transition: 2s linear; /\* 过渡时间 过渡曲线 \*/

transition: width 2s linear; /\* 过渡属性 过渡时间 过渡曲线 \*/

transition: width 2s 3s linear; /\* 过渡属性 过渡时间 延时时间 过渡曲线 \*/

### 11.2.6 transform变换

#### 11.2.6.1 rotate（180deg）（旋转）

##### 单位：

1. deg 角度（180°）
2. turn 圈

##### 写法：

1. transform： rotate(180deg);（默认值，绕Z轴转）
2. transform： rotateX(180deg);（绕X轴顺时针转）
3. transform： rotateY(180deg);（绕Y轴逆时针转）
4. transform： rotateZ(180deg);（绕Z轴顺时针转）

#### 11.2.6.2 scale（0.5） （缩放）

1. 大于1表示放大，小于1表示缩小
2. 一个值 既代表X轴，也代表Y轴
3. 两个值 第一个代表X轴，第二个代表Y轴，逗号隔开

#### 11.2.6.3 translate（X，Y） （位移）

位移后元素初始位置依然占据， 且移动时不会影响其他元素的布局

##### 一个值

transform：translate（100px）; 代表X轴方向

transform：translateX（100px）; 代表X轴方向

transform：translateY（100px）; 代表Y轴方向

transform：translateZ（100px）; 代表Z轴方向，需要3D环境

##### 两个值

第一个值代表X轴方向，第二个值代表Y轴方向

transform：translate（100px,100px）;

#### 11.2.6.5 变换基点transform-origin

变换的基点（即参考点），默认的基点：绝对中心点，该属性提供两个参数值：

1. 如果设置两个，第一个为X轴，第二个为Y轴（两值以空格隔开）
2. 如果设置一个，该值为X轴，第二个默认Y轴50%（给定值时）

##### 取值：

百分比

百分比指定坐标值，可以为负值

transform-origin：50% 50%;

px

长度指定坐标值，可以为负值。

transform-origin：50px 50px;

left、right、center、top、bottom两两组合

left、right、center X轴

top、center、bottom Y轴

只给一个值，另一个值默认center

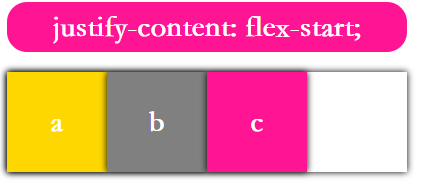
### 11.2.7 弹性盒子布局

前提：父级元素添加display: flex属性

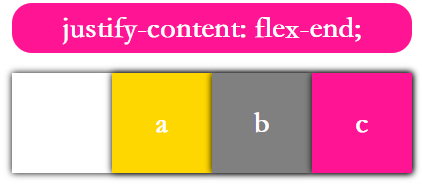
#### 11.2.7.1 Justify-content

定义： 定义了项目在主轴（通常是x轴）上的对其方式

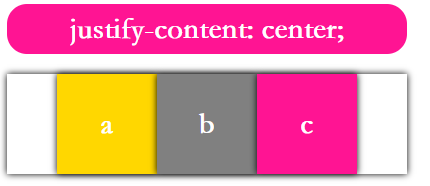
##### 1、justify-content：flex-start;（左对齐）



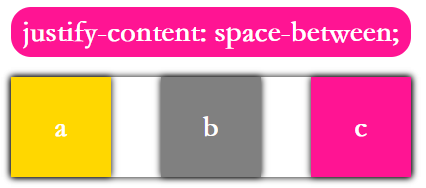
##### 2、justify-content: flex-end；（右对齐）



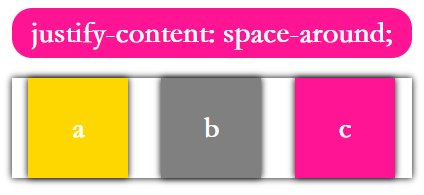
##### 3、justify-content: center；（居中对齐）



##### 4、justify-content: space-between；（两端对齐，项目之间间隔相等）



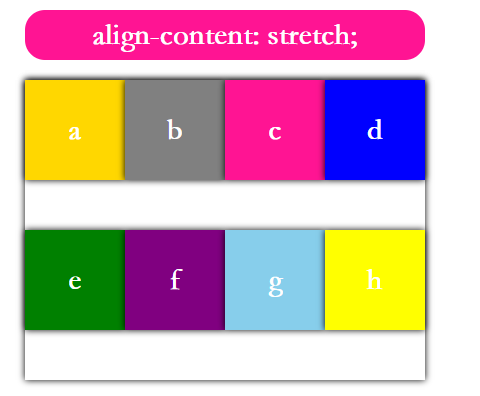
##### 5、justify-content: space-around；（每个项目两侧间隔相等）



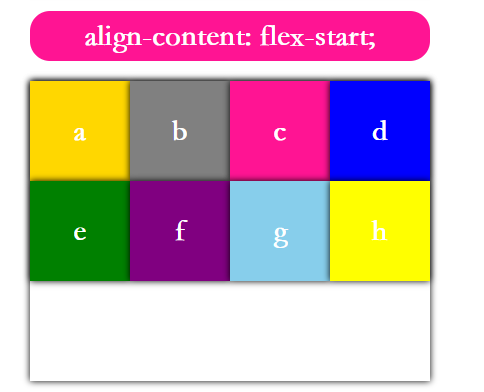
#### 1.2.7.2 align-content

定义：定义了多根主轴对齐方式，如果项目只有一根轴线，则不起作用

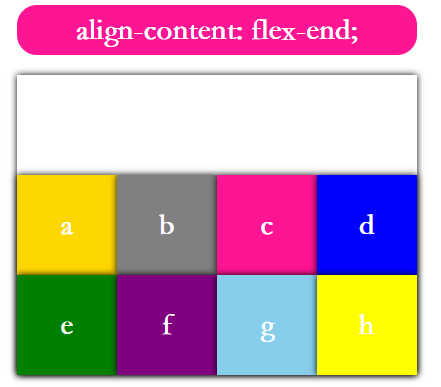
##### 1、align-content: stretch（默认值，轴线占满整个交叉轴）



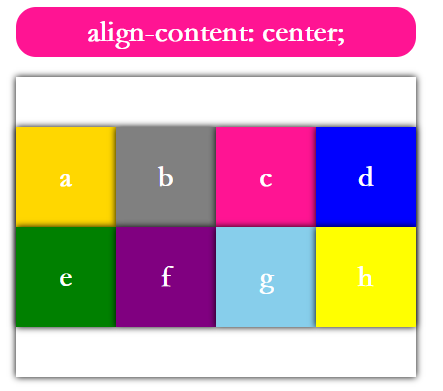
##### 2、align-content: flex-star（与交叉轴上沿对齐）



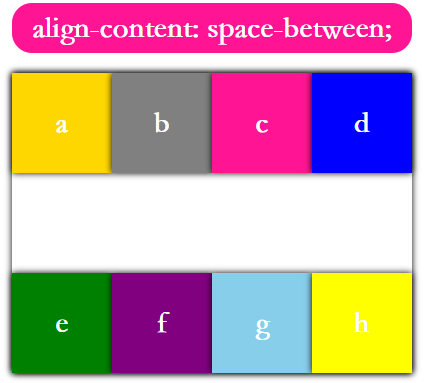
##### 3、align-content: flex-end（与交叉轴下沿对齐）



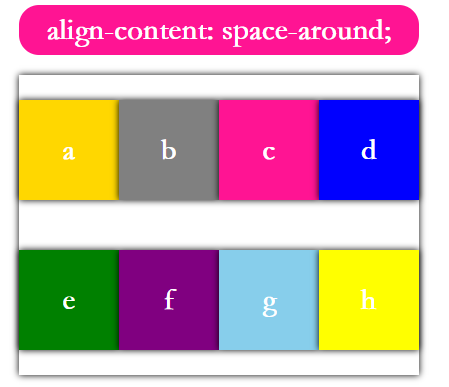
##### 4、align-content: center（与交叉轴中部对齐）



##### 5、align-content: space-between（与交叉轴两端对齐，中间主轴宽度平均分配）



##### 6、align-content: space-around（交叉轴两侧间隔相等）



# 十三、表单表格

## 13.1 表单

### 13.1.1 表单概述

#### 13.1.1.1 定义：

表单是一个包含**表单元素**的区域，表单元素是允许用户在表单中（比如：文本域、下拉列表、单选框、复选框等等）**输入信息**的元素。

1. 获取焦点：输入框光标闪烁的时候 有外轮廓、
2. 失去焦点：没有光标了，没有外轮廓了
3. 外轮廓：获取焦点的时候表单会有外轮廓。outline: none; 去除获取焦点时的外轮廓

#### 13.1.1.2 构成

表单由一个**表单区域**（**form**）和若干**表单元素**（如**input**）构成。input、select是行块级元素

<form action=**' http://www.baidu.com '** method=**'get'**>  
 <input type=**"text"**>  
</form>

#### 13.1.1.3 属性

1. action="http://www.baidu.com" 提交信息的目标地址
2. method="" 值：get post 方式
3. type="" 每个表单元素都有一个type属性，值不同 功能也不同
4. name="" 指定表单元素的名字，对表单元素进行标识，与value属性的值进行对应，后台获取提交的值，就需要通过name
5. value="yajie" 表单默认值, 或者表单中输入的值
6. placeholder="请输入用户名" 提示信息（css3）
7. disabled 禁用输入框
8. checked 单选、多选框默认选中
9. selected 下拉列表默认选中

### 13.1.2 input

input是form表单里的一个表单项，可以通过不同的type值成为不同作用的表单元素

#### 13.1.2.1 text 普通文本框

**账户名：**<input type=**"text"** name=**"user"** value=**"yajie"** placeholder=**"请输入用户名"**>

#### 13.1.2.2 password 密码框

输入的文本会用小圆点显示

密码：<input type="password" name="pwd" >

#### 13.1.2.3 radio 单选框

只能选中一项，但是所有的name值必须一致

请输入您的性别：

<input type="radio" name="sex" id="sex1"><label for="sex1">男</label>  
<label><input type="radio" name="sex" checked>女</label>  
<label><input type="radio" name="sex">人妖</label>

##### lable标签

通过lable标签的for属性可以关联标记某些表单元素，使用户在使用鼠标操作时更加方便

示例代码结构：

**请输入您的性别：**<input type=**"radio"** name=**"sex"** id=**"sex1"**><label for=**"sex1"**>**男**</label>  
<label><input type=**"radio"** name=**"sex"** checked>**女**</label>  
<label><input type=**"radio"** name=**"sex"**>**人妖**</label>

两种方法皆可以

#### 13.1.2.4 checkbox 多选框

可以选中多项，即使name一致也不行。

**请输入您喜欢吃的水果：**<label><input type=**"checkbox"** name=**"apple"**>**苹果**</label>  
<label><input type=**"checkbox"** name=**"watermelon"** checked>**西瓜**</label>  
<label><input type=**"checkbox"** name=**"banana"**>**香蕉**</label>  
<label><input type=**"checkbox"** name=**"orange"**>**橘子**</label>

#### 13.1.2.5 reset 重置

清空输入的所有信息

<input type=**"reset"** value=**"重置"**>

#### 13.1.2.6 hidden 隐藏输入框

<input type=**"hidden"**>

#### 13.1.2.7 file 文件域

上传文件

<input type=**"file"**>

#### 13.1.2.8 submit 提交 <button></button>标签默认为submit类型

输入信息完成，点击提交按钮提交到指定地址（action）

<input type=**"submit"** value=**"提交"**>  
<button>**百度一下**</button>

### 13.1.3 文本域

textarea 多文本输入框（文本域）

<textarea name=**""** id=**""** cols=**"50"** rows=**"10"**></textarea>

cols是宽度，rows是高度。但是我们一般使用css样式去控制宽高，resize:none使用户不能拖动文本域

textarea{  
 width: 200**px**;  
 height: 100**px**;  
 resize: **none**; /\*使用户不能拖动文本域\*/  
}

### 13.1.4下拉列表

下拉列表：由select标签包裹的一项或者多项option

##### 可选属性：

selected 默认选中

disabled 禁用表单元素

multiple 按住Ctrl就可以选中多个

size 使下拉列表显示多个

<select name=**""** id=**""** multiple size=**"3"**>  
 <option value=**""**>**静心**</option>  
 <option value=**""** selected>**小武**</option>  
 <option value=**""**>**云子**</option>  
 <option value=**""**>**小王子**</option>  
 <option value=**""**>**蚊子**</option>  
</select>

### 13.1.5 定义域

定义域表单：由fieldset包裹，里面由legend定义域标题和一个或者多个表单元素组成。

<fieldset>  
 <legend>**快递单**</legend>  
 <label><input type=**"text"**>**姓名：**</label> <br><br>  
 <label><input type=**"text"**>**电话：**</label> <br><br>  
 <label><input type=**"text"**>**地址：**</label> <br><br>  
</fieldset>

## 13.2 表格

表格的构成通常由一个表格主体（table）然后包含若干行-单元行（tr）表格行内包含的若干列-单元格（td、th）构成，也可以通过（thead、tbody、tfoot）将表格进行表头、表体、表脚区域语义化的划分。

### 13.2.1 表格的结构

<table border=**"1"**>  
 <thead>  
 <tr><th colspan=**"4"**>**食物表**</th></tr>  
 </thead>  
 <tbody>  
 <tr><th>**水果**</th><th>**蔬菜**</th><th>**主食**</th><th>**调料**</th></tr>  
 <tr><td>**苹果**</td><td>**白菜**</td><td>**大米**</td><td>**辣椒**</td></tr>  
 <tr><td>**香蕉**</td><td>**青菜**</td><td>**小麦**</td><td>**大葱**</td></tr>  
 </tbody>  
 <tfoot>  
 <tr><th>**备注**</th><td colspan=**"3"**></td></tr>  
 </tfoot>  
</table>

### 13.2.2 表格的属性

#### 13.2.2.1 表格边框属性

给table标签属性：border="1" 等价 给 table th td 边框css样式

合并边框 border-collapse: collapse;

#### 13.2.2.2 css属性

1. 给table宽度,高度。默认表格每一项会平均分配宽，除去thead和tfoot，tbody里面的每一行平均分配高  
2. 可以给th,td单独的宽高. 如果有冲突，会取最大值。  
3. 字体居中：th默认上下左右居中，td默认上下居中。可以通过text-align: center; 设置文本对齐方式  
4. table支持margin，th、td不支持margin  
5. 给table和th,td背景是等价的，color也可设置  
6. colspan 合并列 不能合并表头和表尾  
7. rowspan 合并行 不能合并表头和表尾  
8. table可以浮动

#### 13.2.2.3 合并行和列

<table class=**"fl-L"**>  
 <tr>  
 <th colspan=**"2"**>**姓名**</th>  
 <!--<th>年龄</th>-->  
 <th>**成绩**</th>  
 </tr>  
  
 <tr>  
 <td rowspan=**"2"**>**拉闸**</td>  
 <td>**18**</td>  
 <td>**59**</td>  
 </tr>  
  
 <tr>  
 <!--<td>剑来</td>-->  
 <td>**30**</td>  
 <td>**58**</td>  
 </tr>  
</table>

PS：注意，在合并行和列的时候需要将多余的项删除

### 13.2.3 表格的特性

##### 表格的独特类型：display: table

1. 表格的独特类型：display: table  
2. 支持margin: auto;  
3. 支持宽高，不给宽高内容撑开宽高  
4. 独占一行

# 十四、阿里图标

阿里矢量图标库，是阿里旗下开发的一系列的图标产品，可以使用在网页，手机APP等多端，阿里图标相对于图片图标体积更小加载更快的优点。

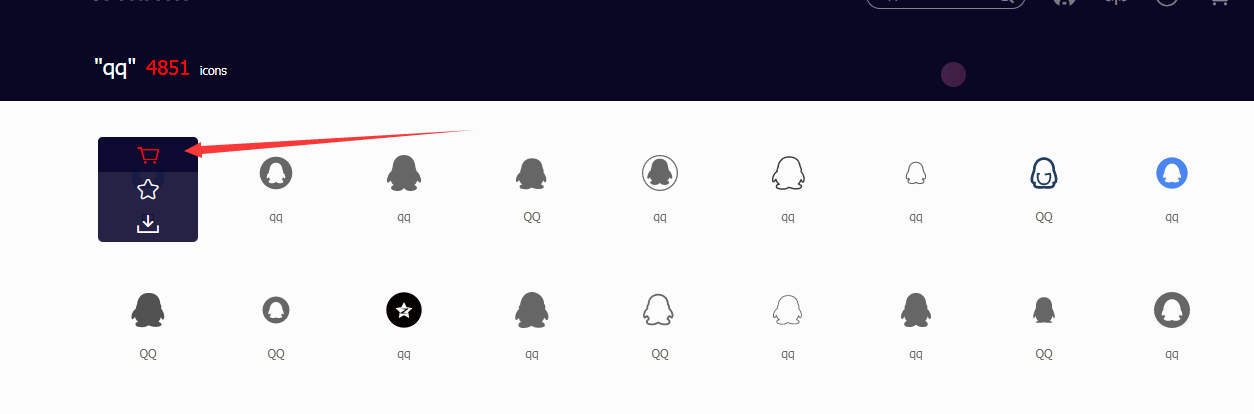
## 14.1添加图标

1. 登陆注册阿里图标官网：<http://www.iconfont.cn/plus>

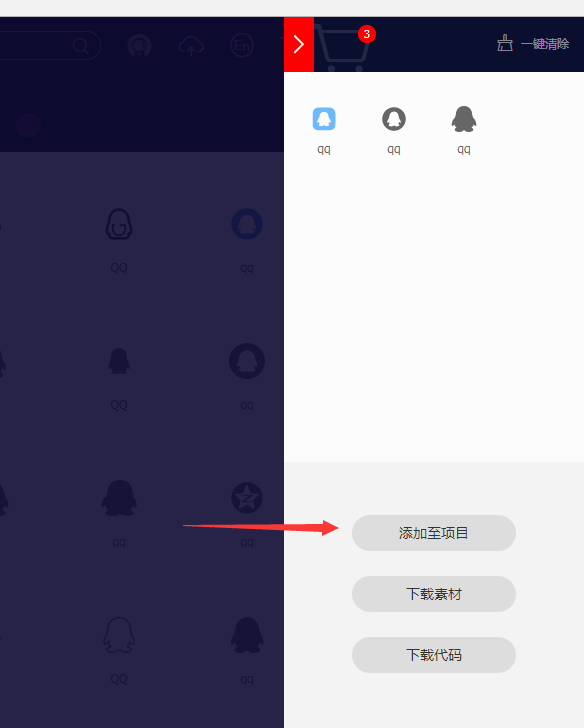
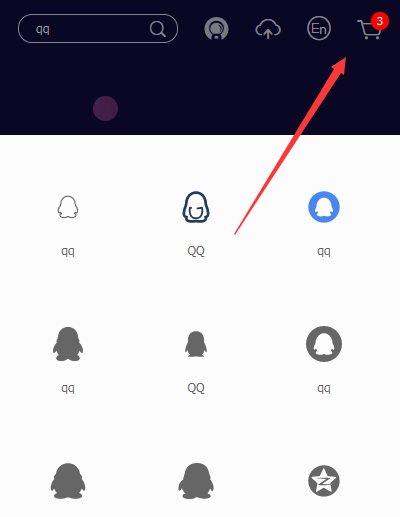
2. 搜索想要的图标

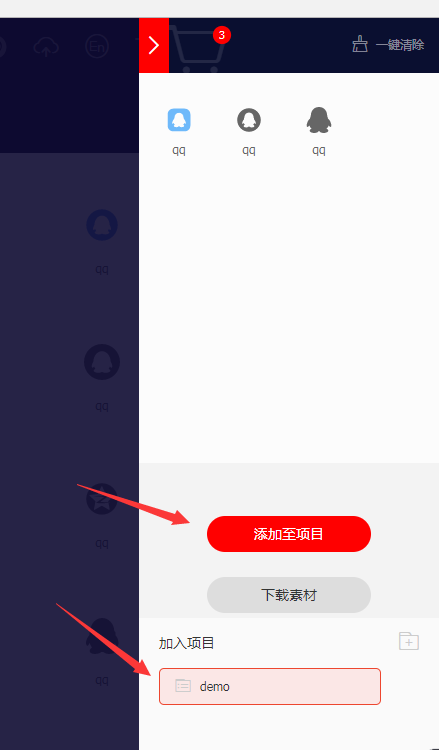


3. 选择想要的图标点击添加到库

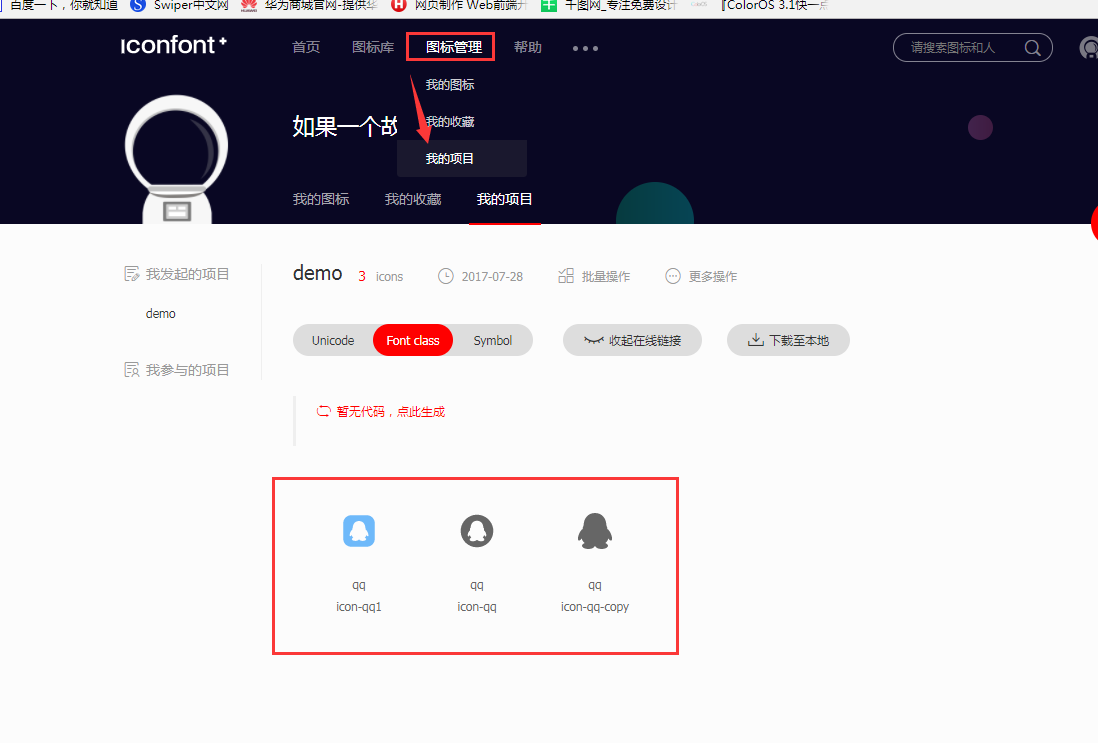


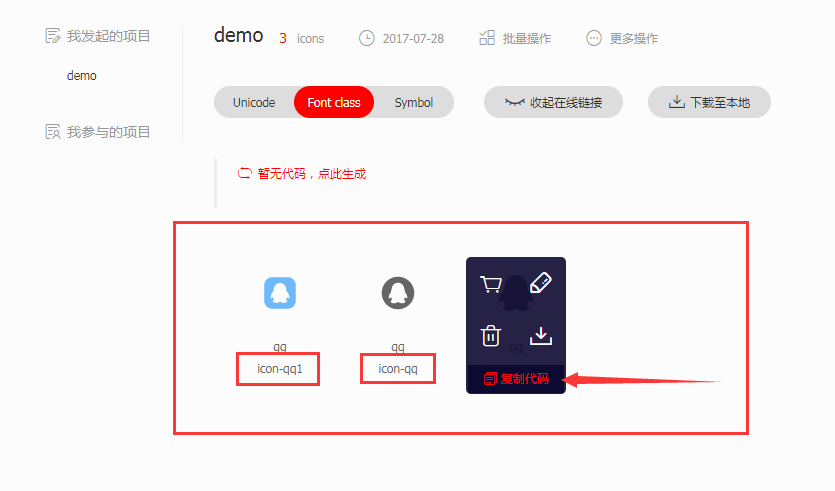
4.选择完想要的图标过后点击购物车，添加至项目





5. 在我的项目可以找到添加好的图标，与图标类名了

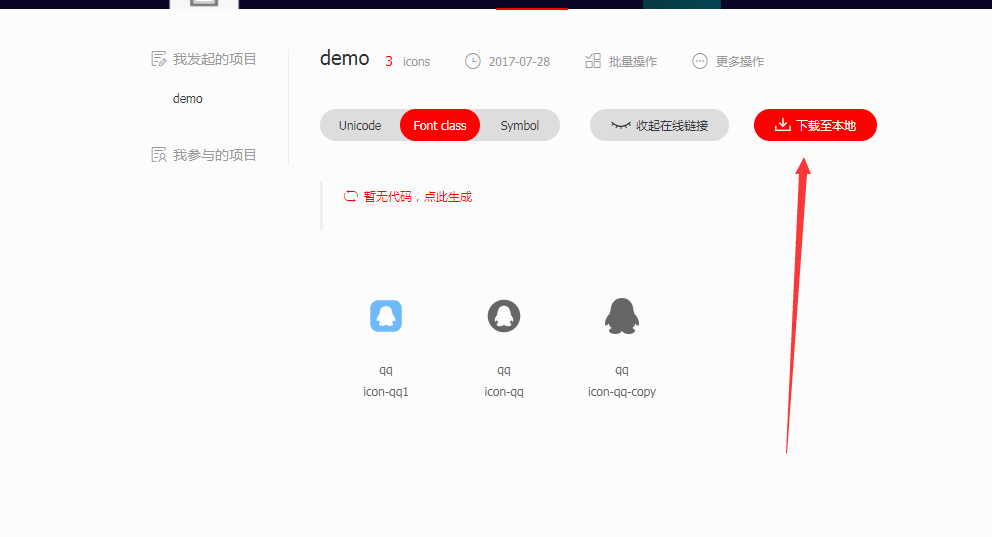


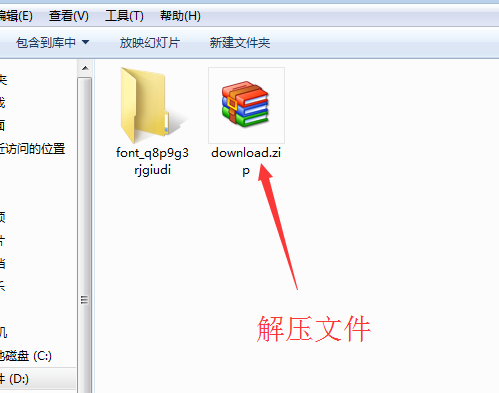


## 14.2 本地引用

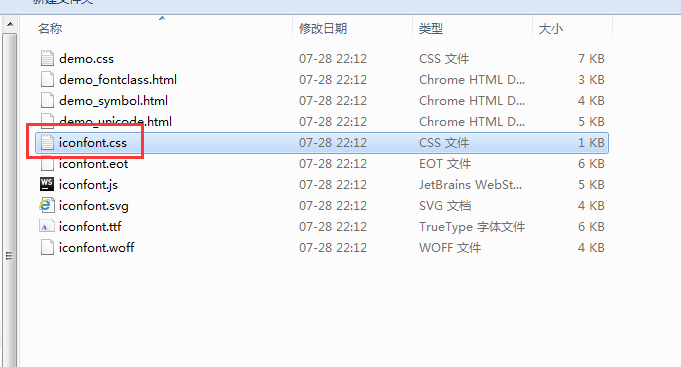
### 14.2.1 下载本地引用图标

1、当整个网页调试完成后，可以下载图标到本地解压引用

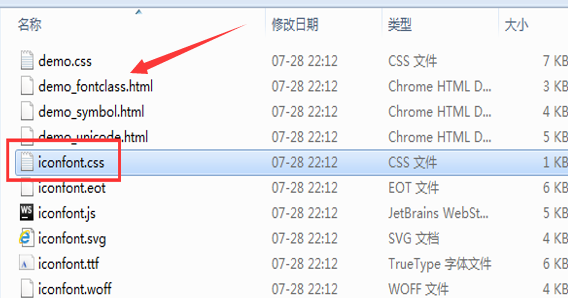




2、通过link标签引用解压后的文件夹里指定的图标css样式文件



3、打开fontclass.html获取每个图标的类名



4、引入iconfont的css

<link rel="stylesheet" type="text/css“ href="font/iconfont.css">

5. 设置class属性

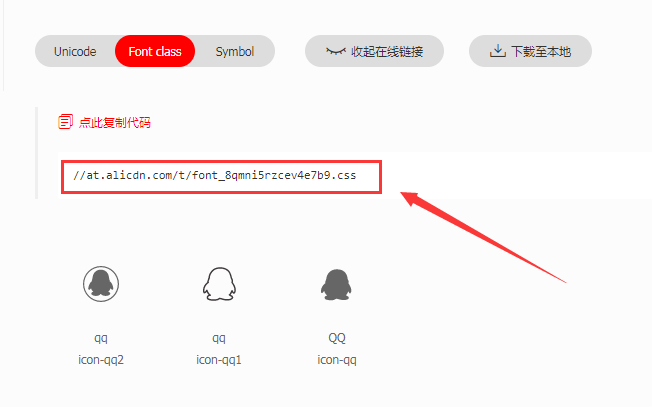
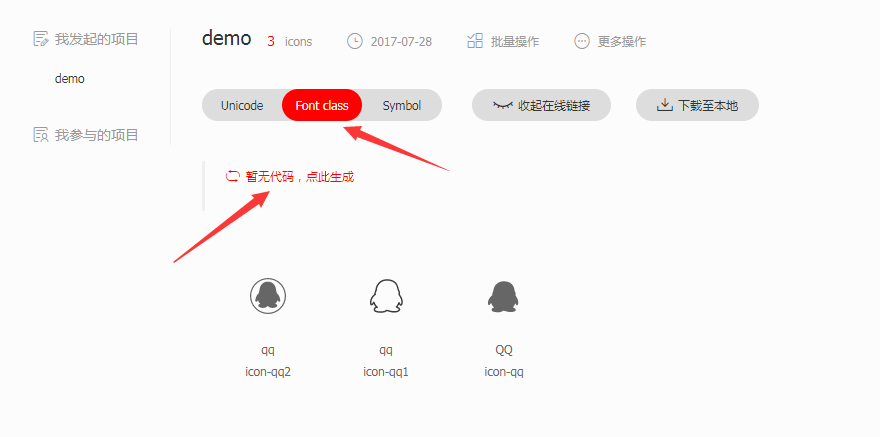
<i class="iconfont icon-weixin1"></i>

6. 设置样式

<i class=**"iconfont icon-weixin1"** style=**"**font-size: 30**px**; color: **red**;**"**></i>

## 14. 3 在线引用

1. 选择font class 并且点击生成代码，看到在线引用地址



2、复制在线地址，通过link标签引用地址（加上 http: 前缀）

3、最后在标签里，添加固定类名与图标类名引用图标

# HTML-CSS企业命名规范

## 1.1 HTML规范

### 1.1.1 代码规范

1. p, dt, h标签里面不能嵌套块属性标签；
2. 内联元素不能嵌套块；（a除外）
3. a标签不能嵌套a；

### 1.1.2 html文件命名

1. 文件名中只可由英文字母a~z、排序数字0~9或间隔符-组成。
2. 文件名中禁止包含特殊符号，比如空格、$等
3. 文件名统一使用小写字母，文件名使用英文名词命名，或英文简写

### 1.1.3 图片

1. 所有img元素必须加上width height alt属性, **修饰性图片**alt值留空

2. 图片命名：

a. 图片后缀命名一律小写。

b. 使用间隔符 - 进行连接。

c. 一般背景图片以bg-开头，

d. 按钮图片以btn-开头，

e. 图标图片以icon-开头，

f. 聚合图以spr-开头，后跟英文单词，如果名称过长，适当使用缩写

### 1.1.4 注释

1. 单行注释：

<!-- 注释内容 -->

2. 多行注释：

<!--

\* 注释内容一

\* 注释内容二

\* 注释内容三

-->

### 1.1.5 HTML命名规范

#### 1.1.5.1 命名的两种方式：

a. ID命名： 具有唯一性(身份证号) 相同的ID名只能在页面中出现一次， 并且ID只能有一个名字

b. class类名：class的类名可以在页面中重复出现，并且一个元素可以有多个类名,(中间以空格分开)

#### 1.1.5.2 HTML命名规范：

a. 用小写字母

b. 以英文开头 可以包含（英文字母 - \_ 数字），不能以数字开头

c. 见名知意

d. 拓展名写法:

e. 加 - 或 \_ （腾讯：下划线 ' \_ ' 禁止出现在class命名中，统一使用 '-' 连字符）

#### 1.1.5.3 常用命名

常用类/ID命名举列，常用类的命名应尽量以常见英文单词为准，做到通俗易懂，并在适当的地方加以注释。 部分命名解释约定：

页头： header

页面主体： main

页脚： footer

登陆： login

标志： logo

广告： banner

导航： nav

菜单： menu

搜索： search

滚动： scroll

功能区：　　shop(如购物车，收银台)

标签页： tab

文章列表： list

提示信息： msg

小技巧： tips

栏目标题： title

子导航： sub-nav

顶导航： top-nav

登录条： login-bar

边导航： side-bar

左导航： l-s-bar

右导航： r-s-bar

子菜单： sub-menu

加入： joinus

指南： guide

服务： service

热点： hot

新闻： news

下载： download

注册： regsiter

状态： status

投票： vote

合作伙伴： partner

友情链接： friendLink

版权： copyRight

商　标： label

搜索： search

注释：　　　 note

面包屑： breadCrumb(即页面所处位置导航提示)

容器： container

外　套： wrap

内容： content

菜单内容1： menu1Content

菜单容量：　 menuContainer

按钮： btn

当前： current

## 1.2 CSS规范

### 1.2.1代码风格

##### 1.腾讯：css属性值需要用到引号时，统一使用单引号

selectors{ font-family:'\5FAE\8F6F\96C5\9ED1';}

##### 2.为单个css选择器或新申明开启新行

.home-count .hd,

.content-title,

.select-game-title .title{ }

##### 3. css属性书写顺序

建议遵循：**布局定位属性** **自身属性** **文本属性** **其他属性** **CSS3属性**

布局定位属性

display position（相应的 top,right,bottom,left） float clear visibility /overflow

自身属性

width height margin padding border background

文本属性

color font text-decoration text-align vertical-align white- space break-word

其他（CSS3）

content cursor border-radius box-shadow text-shadow background:linear-gradient ...

##### 4.删除css属性值为0的单位

/\* 不推荐 \*/

selectors{ margin:0px; padding:0px;}

/\* 推荐 \*/

selectors{ margin:0; padding:0;}

### 1.2.2 注释

##### 1.行间注释

.search{background: url('') no-repeat; /\*定义搜索框的背景\*/}

##### 2.整段注释

/\*===== 搜索条 =====\*/

.search {background: url(../img/search.gif) no-repeat;}

/\*===== 搜索条结束 =====\*/