### Webstorm快捷键使用

End 快速回到句末

Home 快速回到句头

Ctrl + D 快速复制光标所在行

Ctrl + X 快速删除光标所在行

Ctrl + Alt + T 快速在选中文本前后加<>

Ctrl + / 注释

Ctrl + shift + 方向键 快速移动选中代码

Ctrl + -/+ 快速展开或合并

Shift+回车 新起一行

# HTML基础

作用只有**标记**

# 1.浏览器请求过程

请求报文：

请求行+请求头+空行+请求体

响应报文：

相应行+响应头+响应体

# 2.URL简述

URL协议类型+服务器IP地址+服务器端口号+资源名称

# 3.HTML的作用

使用标记语言显示html文件的语义

标签：用于描述其它文本语义的文本，在浏览器中不会显示出来，所以叫超文本

注：

虽然使用HTML使得网页显示格式变化，但他实际作用是用来**显示语义**的

# 4.HTML基本格式

## 基本框架：

**<!DOCTYPE html>** **--html5的DTD声明**

<html>

<head lang = “en”>

<title></title>

</head>

<body>

</body>

</html>

注：标签都成对出现，不带/的称为开始标签，带/的称为结束标签

## **HTML标签：**

使浏览器识别为HTML文件，其他所有标签都必须在其内部

## **Head标签：**

添加配置信息

1. 网站标题，网站小图标
2. SEO相关信息（指定网站的关键字、指定网站的描述信息）
3. 外挂一些外部的CSS/Js文件
4. 添加一些浏览器适配的信息

注意点：

一般情况下，写在head标签内部的内容都不会显示给用户查看

## **Title标签：**

指定网站的标题，并且这个指定的标题会作为用户保存网站的默认标题

注：title标签必须写在head标签内

## Body标签：

填写需要用户查看的内容（文字、图片、音频等）

注：一对HTML标签中只能有一对body标签

## Link标签：

<link rel="icon" href="地址 " sizes="宽X高">

**（rel=‘icon’设置网页小图标）**

## DTD文档声明：

由于HTML版本众多，此申明用于让浏览器按照DTD声明正确执行网页

注：

1. DTD必须写在文档第一行
2. 不区分大小写
3. 它不是个标签

DTD分类：（分html和xhtml，共六种）

1. Strict ：（严格的）对使用非描述标签进行禁用，且不能使用标签，因为html语言本身只是显示句意而非格式，如font等不可用
2. Transitional：（普通的）无特殊要求，但不能使用标签
3. Frameset：（带框架的，后续再补充）

# 5.字符集

乱码产生：没有指定字符集，或者文件保存的字符集与解析的字符集不同

Meta标签：指定字符集（设置网页编码格式），**放在head标签内**

**例：<meta charset = “GBK” />**

常用字符集：GB2312 存储汉字及常用外文，体积小，访问速度快（推荐纯汉字网页使用）

**UTF-8** 储存全部语言，体积大（推荐含有外文的网页使用）（推荐）

**注：编写代码时的字符集应和用于解析的字符集相同**

# 6.标签分类

双标签：开始标签与结束标签成对出现

单标签：只有开始标签

并列关系：同级别

嵌套关系：父子关系

# 7.基础标签

## 7.1 H系列标签（H1~H6）【双标签】

例：<H1>XXXX</H1>

作用：用于给文本添加标题语义

注：1. 范围只能1~6，超过6则无效

2. 用于标记语义而非改变格式

3. 显示的内容会独占一行

4. **一个网页中只能又一个H1标签**（与SEO有关）

## 7.2 P标签（段落标签）【双标签】

例：<p></p>

作用：标记段落

注：显示在一行

## 7.3 hr标签【单标签】

例：<hr>

作用：划分线

注：显示在一行

## 7.4 img标签【单标签】

例：<img src=”” Alt=”” width=”” hight=”” title=””>

作用：显示图片

属性： src 资源位置

Width 宽度

Height 高度

Title 告诉浏览器图片内容（用户鼠标悬停时会显示title值）

Alt （alternate交换）告诉浏览器找不到图片时显示什么

注：单个设置width或height时等比例缩放

显示的图片不在一行

## *7. 5 a标签【双标签】*

<a href=”指定需要跳转的目标界面” >被链接对象</a>

作用：设置超链接

属性：target 控制是否跳转到新的界面（\_self / \_blank）

Title 悬停显示内容

注：1. 可以设置文字，图片

2. 网址前要加http://

### 7.5.1 假链接

假链接 — 对象还未真正设置超链接

假链接格式：href=“#” 点击后会回到网页顶部（可用于功能：返回顶部）

Href=“JavaScript:” 点击后页面不会有任何变化

### 7.5.2 锚点

\*每个标签都可以设置id ，可以通过假链接**<a href=”#id号”>文本</a>**跳转

注：1. 这样实现的跳转没有过渡动画

2. a标签除了可以跳转到当前界面的指定位置外，还可以跳转到其他界面的时候直接跳转到其他界面的指定位置 例：**<a href=”测试.html#1000”>测试</a>**

不用过渡动画的可以用a标签，需要有***过渡动画***的不要用a标签做

## 7.6 base标签【单标签】

例：<base target = ”\_blank”>

作用：来统一所有超链接打开窗口方式

注：可在单个a标签中单独修改

## 7.7 列表标签

Webstrom快捷写法 如：ul>li\*3 +tab

作用：给一堆数据添加列表语义

列表分类：

1. 无序列表（unordered list--最多）

2. 有序列表（ordered list-- 最少）

3. 定义列表（其次）

注：1. 单纯用于添加语义而非格式

2. ul和li标签都是成对出现的

3. 由于ul和li是一个组合，使用不推荐包含其他标签

例：**无序列表格式**

<ul> --unordered list

<li></li> --list item

</ul>

应用场景：

1. 新闻列表
2. 商品标签
3. 导航条

**定义列表**

<dl>

<dt></dt>

<dd></dd>

</dl>

注：

应用场景：

1. 做网页对尾部信息
2. 做图文混排

## 7.8 table标签

<table>

<caption></caption> --标题标签，要紧跟table标签后面

<tr> --表示表中一行（table row）

<td></td> --一对td表示一行中的一个单元格

</tr>

</table>

Th标签 – 专门用于显示标题，自动加粗居中（用法：把<td>换成<th>）

属性（可在tr和td使用 ）：

1. border 边框（默认为0）
2. bgcolor 设置背景颜色
3. width和height 宽高
4. colspan 设置水平方向合并单元格
5. rowspan 设置垂直方向合并单元格
6. align 水平方向对齐方式（left，center，right）
7. valign 垂直方向对齐方式（top， ）
8. cellspacing 外边距大小，默认为2（单元格间的间隙）
9. cellpadding 内边距大小，默认为1（内容距离边框的距离）

注：

1. **注意不要使用属性更改格式，html语言仅用于描述**
2. 先要细线边框的话要先设置背景颜色，靠与单元格颜色的反差，外边距就成为了线
3. 表格默认边框值为0，不显示边框
4. 设置合并单元格后需要删除多余的单元格
5. Table和tr和td是个组合
6. Table的width和hight属性

## 7.7 br标签【单标签】

<br>

作用：换行

注：企业开发中很少用换行，**基本都用<p></p>**

# 8.路径

路径（同级，下级，上级【../】）

1. 一定要用/
2. 一定要用相对路径（涉及转移问题）

# 9.表单

作用：收集信息

在浏览器中所有的表单标签都有特殊的外观和默认的功能

格式：  
<form>

<表单元素>

</form>

常见的表单元素：

## 9.1 input标签

前缀<input **type**=”text”>

注：默认情况下文字和输入框没有关联关系（点击文字不会聚焦到输入框）---使用lable标签实现

属性：

1.1 Text 明文输入框

1.2 Password 暗文输入框

1.3 Value 默认值设置

1.4 name 表示将被提交到服务器

2.1 Redio 单选框（性别：< input type =“redio” >男）

2.2 name 多个redio的name值相同时，只能单选中一个

2.3 checked （单多选框都可用）值为chacked（可省略但不建议） 设置默认单选

3.1 checkbox 多选框

* 1. Checked

4.1 Button 普通按钮可配合js完成一些操作（< input type =” button”）

4.2 value 设置值

4.2 Image 图片标签（< input type = “image” src = “地址”>）

5.1 reset 重置按钮，有默认标题“重置”

5,2 value 修改标题

6.1 submi 提交按钮（将表单数据提交到服务器）

6.2 服务器设置在<form>标签的 action 属性设置

7.1 hidden 不会显示出来，用于私下收集信息

Type值 以下了解即可（兼容差）：

1. Email 输入邮箱
2. url 输入网址
3. number 输入数字
4. date 选择时间
5. color 用取色版选取颜色

## 9.2 lable标签

作用：使文字与输入框绑定

格式1：**<label for="001">id</label><input type="text" id="001"> --推荐（可非对应）**

格式2：**< label >**

**<input type="text" >**

**</label> --有局限**

## 9.3 select标签

作用：用于定义下拉列表,无需绑定

<select >  
 <option>待选项</option>  
</select>

属性：

1.selected 默认选择

2. optgroup 选项分组

<select>  
 <optgroup label="num">  
 <option>1</option>  
 <option>2</option>  
 </optgroup>  
</select>

## 9.4 textarea标签

作用：可多行文本框

格式：

<textarea>

</textarea>

属性：

1. cols 宽度（一行显示多少字）
2. rows 高度（显示多少行）

注：

1.双标签中间不填默认为空

2.cols和row只是控制文本框大小，不能限制输入的文字多少

3.默认情况下，文本框可以手动调整大小

## 9.5 datalist表签 (了解)

注：众多浏览器不支持

作用：给输入框绑定待选项

格式：

<input type=”text” list=”001”>

<datalist id=”001”>  
 <option>待选项</option>  
</datalist>

# 10.多媒体标签

## 10.1 video标签

作用：播放视频

格式1：<video src=””></video>

格式2：<video>

<source src=”URL” type=”video/mp4”></source>

<source src=”” type=””></source>

</video>

属性：

1. autoplay=“autoplay” 设置自动播放

2. controls = “controls” 显示控制条

3. poster=“url” 设置视频开始前的预览图

4.loop=“loop” 控制是否循环，一般做广告视频

5.preload 预加载视频（与autoplay相斥，autoplay会失效）

6.muted=“muted” 设置静音

7.width

8.height

## 10.2 audio标签

作用：播放音频

格式同video ，两者使用基本一样

只有三个属性不能使用：width，height，poster

## 10.3详情和概要标签

作用：用summary标签来描述概要信息，用details标签来描述详细信息

格式：

<details>

<summary>概要信息</summary>

详细信息

</details>

## 10.4 marquee标签

注：Marquee标签不是W3C推荐的标签，在W3C官方文档中也无法查询这个标签，但在各大浏览器对这个标签的支持非常好

作用：跑马灯

格式：

<marquee>内容</marquee>

属性：

1.direction 方向（up，down，left，right）

2.scrollamount 速度（数字越大越快）

3.loop 循环次数

4.behavior 滚动方式（slide到边界就停止，alternate边界来回滚动）

注：也可让图片滚动

## 10.5 部分被废弃的标签

原因：html只用标记语义

无语意 例：

1.Br 换行

2.hr 分割线

3.font 修改字体 有语义的替换

4.b 加粗 strong：定义重要性强调的文字

5.u 加下划线 ins：（insered）定义插入的文字

6.I 文本倾斜 em：（emphasizd）定义强调的文字

7.s 删除线 del：（deleted）定义被删除的文字

## 10.6 字符实体（类似转义字符）

在html中，对空格、回车、tab不敏感，会把其多个当做一个处理

字符实体（后面一定要加分号）：

&nbsp; 空格

&lt; < (less than)

&gt; > (greater than)

&copy ©

其他的查文档

# CSS基础

1.选择器

2.属性

# 1. CSS固定格式

注：

1. <style>标签必须写在<head>内

2. type属性默认为text/css

3. 必须按固定的格式

格式：

<style type="text/css">

h1{

text-align: center;

color: #a52a2a;

font-family: "微软雅黑 Light";

font-size: 80px;

}

标签名称{

属性：属性值

……

}

</style>

# 2.文字属性

## 2.1 font-style 样式

取值：

1、normal 正常

2、italic 倾斜

快捷键：输入（fs）+tab（或回车） 默认为italic

输入（fsn）+tab（或回车）则为norma

## 2.2 font-weight 粗细

取值：

1.设置数字（100为细线，900为加粗）

2.bold 加粗

3.bolder 加粗

4.light

快捷键：fwl，fwb

## 2.3 font-size 大小

取值：

像素大小

注：设置像素大小要加px

快捷键：fz + 大小

## 2.4 font-family 字体

取值：

字体类型

注：

1. 若使用的字体用户未安装，则默认使用宋体，（可设置为其他字体）

· 格式：（font.family = “未安装的字体”，“其次使用的字体”）

2. 想要将英文和中文字体分开设置，要先设置英文的字体，再设置中文的字体

原因：中文字体能处理英文字体而英文字体不能处理英文字体，使用当先英文字体处理不了中文时，就会应用后面的中文字体

格式：同上

常用字体：

中文：宋体、黑体、微软雅黑

英文；times news roman /arial

注：

1. 默认宋体

2. 字体是中文要加引号

3. 设置的字体必须是已安装的

快捷键：ff

## 2.5 简写格式：font：粗细 斜体 大小 字体

注：

1. 大小 字体 一定要写，其他的粗细和斜体可省略

2. 大小和字体的顺序不能换，且一定要放在最后面

## 2.6 text-decoration 装饰（下划线，删除线）

值：

1. none 取消装饰 （常见用途：去掉超链接自带的下划线）

2. underline 下划线

3. line-through 删除线

4. overline 上划线

快捷键：td 、 tdl 、tdn 、tdo

## 2.7 text-align 文本对齐方式

值：

1. center

2. right

3. left

快捷键：

## 2.8 text-indent 文本缩进

值:

1. 2em （推荐）一个em代表一个字的缩进

2. 2px 缩进两个像素

快捷键：ti ，ti2e ，ti2

## 2.9 color 颜色

值：

1. 英文单词

2. rgb（00,00,00）

3. rgba (00,00,00,0) 第四位代表透明度

4.十六进制

5.十六进制缩写

# 3. 选择器

## 3.1 标签选择器

注：

1. 标签选择器选中的是当前界面中所有的标签，不能单独选中一个标签

2. 标签选择器无论标签藏的多深都能被选中

3. 只要是HTML中的标签就可以被选中

## 3.2 id选择器

使用格式：

<style>

#id号{

}

</style>

注：

1. 每个标签都有id属性

2. id不可重复

3. id名称只能由字母下划线和数字组成，**不能由数字开头**

4. id不能用标签名

5.CSS使用时要前面加**#**

**6. id尽量不要使用，id后期会留给js使用**

## 3.3 类选择器 class

格式：<标签 class=”类名1 类名2 。。。。”>

注：

1. 每个标签都有class属性

2. class可重复

3. class名称只能由字母下划线和数字组成，**不能由数字开头**

4. class不能用标签名

5.CSS使用时要前面加 **. (点)**

**6.类名是专门用来设置样式的**

**7.一个表签可以绑定多个类名的**

## 3.4 id选择器 和 calss选择器 区别

1.1

id 不可重复

class可重复

2.

id选择器以#开头

class选择器以 . 开头

3.

一般开发中。id是留给js使用的，而calss是留给css的

## 3.5后代选择器

作用：找到指定标签所有的后代标签，设置属性

格式：标签名称1 标签名称2{

属性：值

}

注：

1. 后代选择器必须用空格隔开

2. 后代不仅包括后面一级，还**包括后面所有级别**

3. 标签名称可以使用其他的选择器

4. 使用**空格**隔开

3.6 子元素选择器

作用：找到指定标签中的直接子元素，设置属性

格式：

标签名称1>标签名称2{

属性：值

}

注：

1. **只会查找子一层**

2. 使用**>**进行连接

3. 标签名称可以使用其他的选择器

3.6交集选择器

作用：选中两个标签中相交的那部分

格式：

选择器1选择器2{

属性：值；

}

注：

1. 两个选择器间**没有任何符号**

2. 交集选择器仅作为了解，开发用得不多

3.7 并集选择器

作用：各选中的所有标签设置属性

格式：

选择器1,选择器2{

属性：值；

}

注：

1. 使用**，**进行连接

2. 标签名称可以使用其他的选择器

## 3.8 兄弟选择器

作用：用于选中**同级**关系的标签

3.8.1 相邻兄弟选择器

作用：选中紧跟后面同级的标签

格式：

选择器1+选择器2:{

属性：值；

}

注: 选中的是选择器2，他一定要紧跟选择器1后面

3.8.2 通用兄弟选择器

作用: 选中选择器1后面所有的选择器2的内容

格式：

选择器1~选择器2{

属性：值；

}

注：是选择器1后所有的选择器2

## 3.9序选择器

### 3.9.1 选中同级别的第几个

格式：

标签名：nth-child（n）{ -同级别第n个(括号内换为odd则选中奇数的位置,even为偶数，里面也可用表达式，如（2n+1）

}

标签名：nth-last-child(n){ --------同级别倒数第n个

}

标签名 ：first-child{

}

标签名：last-child{

}

标签名：only-child{ -------同级别中唯一的标签（一个级别中有多个标签时是失效）

}

### 3.9.2 选中同类型的第几个

作用：选中同级别中同类型的第几个

格式：

标签名：nth-of-type（n）{ 同类型第n个(括号内换为odd则选中奇数的位置,even为偶数）

}

标签名：nth-last-of-type（n）{

}

标签名:first-of-type{

}

标签名;last-of-type {

}

标签名：only-of-type{

}

## 3.10 属性选择器

作用：根据指定的属性名找到对应的标签，然后设置属性（最常用于input标签的选择 ）

格式：

1. [attribute]

2 . [attribute == value]

例：

p[id]{

属性：值；

}

3.值以什么开头

[attribute |= value] CSS2

[attribute ^= value] CSS3

4. 值以什么结尾

[attribute $= value] CSS3

5. 值包含什么的

[attribute ~= value] CSS2

[attribute \*= value] CSS3

3和5中的css2要求是被隔开的，

3.11 通配符选择器

作用：给界面上所以的标签设置属性

注：由于设置所有的标签，标签过多时性能会非常低，

# 4. CSS三大特性

## 4.1 继承性

作用：父元素设置的属性，子元素也可用使用

注：

1. 不是所有的属性都可以继承，只有以color/text-/font-/line-开头的属性才可以继承

2. 只要是后代都可以继承

3. 继承的特殊性

3.1 a标签的文字颜色和下划线是不能继承的

3.2 h标签的文字大小是不能继承的

应用场景：

一般设置网页上的公用信息，例如网页上的颜色、字体大小、字体等内容，一般直接给body设置属性

## 4.2 层叠性

作用：当同一个标签被设置多个同样的属性，CSS

注： 只有多个选择器选中“同一个标签”，设置同一个属行时，才会启用层叠性

## 4.3 优先级

作用：确定层叠性的优先顺序

优先级判断的三种方式：

1. 是否是直接选中（间接选中是指继承）

如果是间接选中，那么谁离目标标签比较近就听谁的

2. 是否是相同的选择器（谁在后面则听谁）

3. 不同选择器，并且不是相同类型的选择器

**id>类>标签>通配符(\*)>继承>浏览器默认**

### 4.3.1 !important

作用：用于提升某个直接选中标签的选择器中的某个属性的优先级的，可以将被指定的属性的优先级提升为**最高**

格式：

p{

color: green !important；

}

注：

1. important **只能用于直接选中**，不能用于间接选中

2. 通配符选择器选中的标签也是直接选中的

3. **!important 只能提升被指定的属性**，其他的属性的优先级不会被提升

4. ！important 必须写在属性值的分号前面

5. ！important 前面的！不能省略

### 4.3.2 权重问题

1.什么是优先级的权重？  
作用：当多个选择器混合在一起使用时，我们可以通过计算权重来判断谁的优先级最高。

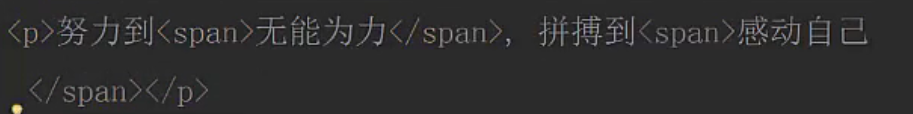
2权重的计算规则  
2.1首先，计算选择器中有多少个id，**id多的选择器优先级最高**2.2如果id的个数一样，那么**再看类名的个数**，类名个数多的优先级高。   
2.3如果类名的个数一样，那么**再看标签名称的个数**，标签个数多的优先级最高。  
2.4**如果id、类名、标签都是一样多那么不会再往下计算，谁在后面听谁的。**

注意点：  
1.只有选择器是直接选中标签的才需要计算权重。

# 5. CSS显示模式

## 5.1容器级及文本级

5.1.1 什么是div？  
作用：一般用于配合css完成网页的基本布局  
5.1.2 什么是span？  
作用：一般用于配合css修改网页中的一些局部信息

  
  
5.1.3 div和span有什么区别？  
 3.1.div会单独占领一行，而span不会单独占领一行  
 3.2.div是一个容器级别的标签，而span是一个文本级的标签  
  
5.1.4 容器级的标签和文本级的标签的区别？  
 容器级的标签中可以嵌套其他素有的标签  
 文本级的标签中只能嵌套文字/图片/超链接

不用记：

容器级： div h ul ol dl li …

文本级： span p buis stong em ins del …

**注: 企业开发一般用div或者按照组标签进行嵌套**

## 5.2块级与行内

在HTML中将所以标签分为两类，分别是容器级和文本级

在css中css也将所有的标签分为两类，分别是块级元素和行内元素

5.2.1 什么是块级元素，什么是行内元素？

***块级元素会独占一行***

***行内元素不会独占一行***

5.2.2块级元素和行内元素的区别？

2.1块级元素

独占一行，

如果没有设置宽度，默认和父元素一样宽。

如果设置了宽高，就按照设置的来。

2.2行内元素

不会独占一行

如果没有设置宽度，默认和内容一样宽

行内元素是不可以设置宽度和高度的

2.3行内块级元素

为了能够让元素既能够不独占一行，又能够设置宽度和高度

那么就出现了行内块级元素。

常见块级元素:

p div h ul ol dl li dt dd

常见行内元素:

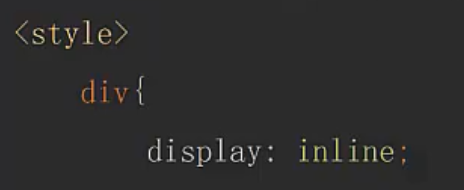
span buis strong em ins del

### 转换方法

1.如何转换css元素的显示模式？

设置元素的display属性

2.display取值



block 块级

inline 行内

inline-block行内块级

3快捷键

di

db

dib

# 6.CSS背景和精灵图

## 6.1 background属性

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **CSS** |
| [background](https://www.w3school.com.cn/cssref/pr_background.asp) | 在一个声明中设置所有的背景属性。 | 1 |
| [background-attachment](https://www.w3school.com.cn/cssref/pr_background-attachment.asp) | 设置背景图像是否固定或者随着页面的其余部分滚动。 | 1 |
| [background-color](https://www.w3school.com.cn/cssref/pr_background-color.asp) | 设置元素的背景颜色。 | 1 |
| [background-image](https://www.w3school.com.cn/cssref/pr_background-image.asp) | 设置元素的背景图像。 | 1 |
| [background-position](https://www.w3school.com.cn/cssref/pr_background-position.asp) | 设置背景图像的开始位置。 | 1 |
| [background-repeat](https://www.w3school.com.cn/cssref/pr_background-repeat.asp) | 设置是否及如何重复背景图像。 | 1 |
| [background-clip](https://www.w3school.com.cn/cssref/pr_background-clip.asp) | 规定背景的绘制区域。 | 3 |
| [background-origin](https://www.w3school.com.cn/cssref/pr_background-origin.asp) | 规定背景图片的定位区域。 | 3 |
| [background-size](https://www.w3school.com.cn/cssref/pr_background-size.asp) | 规定背景图片的尺寸。 | 3 |

### background: 背景颜色 背景图片 平铺方式 关联方式 定位方式;

（任何一个属性都可以被省略）

## 6.2 background-image: url()

6.2.1 如果图片的大小没有标签的大小大, 那么会自动在水平和垂直方向平铺来填充

6.2.2 如果网页上出现了图片, 那么浏览器会再次发送请求获取图片

## 6.3 background-repeat

**repeat 默认, 在水平和垂直都需要平铺**

**no-repeat 在水平和垂直都不需要平铺**

**repeat-x 只在水平方向平铺**

**repeat-y 只在垂直方向平铺**

场景：

可以通过背景图片的平铺来降低图片的大小, 提升网页的访问速度

## 6.4 background-position: 水平方向 垂直方向;

水平方向: **left center right**

垂直方向: **top center bottom**

也可设置像素值（可负）

应用：可以通过定位属性保证图片永远居中显示

## 6.5 background-attachment 背景关联方式

scroll ：默认值， 会随着滚动条的滚动而滚动

fixed ：不会随着滚动条的滚动而滚动，一直静态显示

## 6.6 插入图片和背景图片的区别

1

背景图片仅仅是一个装饰, **不会占用位置**

插入图片会占用位置

2

背景图片有定位属性, 所以可以很方便的控制图片的位置

**插入图片没有定位属性**, 所有控制图片的位置不太方便

3

**插入图片**的语义比背景图片的**语义要强**, 所以在企业开发中如果你的图片想被搜索引擎收录, 那么推荐使用插入图片

**一个画面由多个图片构成的情况，大多是用”div+图片背景“的嵌套构成的**

## 6.7 CSS精灵图

CSS精灵图作用：

可以减少请求的次数, 以及可以降低服务器处理压力

如何使用CSS精灵图

CSS的精灵图需要配合背景图片和背景定位来使用

**通过设置div的宽高，通过背景图片定位，是局部的图片内容显示出来**

# 7.盒子模型

## 7.1 边框

**border: 边框的宽度 边框的样式 边框的颜色;**

<https://www.w3school.com.cn/jsref/prop_style_border.asp>

7.1.1样式：

solid：实线

dashed：虚线

7.1.2 border-width: 上 右 下 左;

注意点:

1. 这三个属性的取值是按照顺时针来赋值, 也就是按照**上右下左**来赋值
2. 这三个属性的取值省略时的规律

上 右 下 左 > 上 右 下 > 左边的取值和右边的一样

上 右 下 左 > 上 右 > 左边的取值和右边的一样 下边的取值和上边一样

上 右 下 左 > 上 > 右下左边取值和上边一样

## 7.2 边距

内边距 padding: 上 右 下 左;

外边框 margin: 上 右 下 左;

### 外边距合并现象

默认布局的垂直方向上外边距是不会叠加的, 会出现合并现象, **谁的垂直外边距比较大就听谁的**

### margin-top问题

1. 如果两个盒子是嵌套关系, 那么设置了里面一个盒子顶部的外边距, 外面一个盒子也会被顶下来
2. 如果外面的盒子不想被一起定下来,那么可以给外面的盒子添加一个**边框属性**
3. 在企业开发中, 一般情况下如果需要**控制嵌套关系盒子之间的距离, 应该首先考虑padding,** 其次再考虑margin(margin本质上是用于控制兄弟关系之间的间隙的)

### box-sizing

1. 作用：这个属性可以保证我们给盒子新增padding和border之后, 盒子元素的宽度和高度不变
2. box-sizing取值：

content-box：(默认取值，增加内边距，元素宽高会改变)

border-box：（增加内边距，元素宽高不会改变）

1. **增加padding和border之后要想保证盒子元素的宽高不变, 系统会自动减去一部分内容的宽度和高度**

## 7.3其他

注：

1. 定位流中margin 0 auto不再生效。
2. 行高可以撑起盒子的高度。
3. 使用ul可以设置菜单栏，要想把元素水平排版，需要转换为行内块级元素，浮动，设置边框，行高。

### 1.设置盒子在另一个盒子中居中

margin: 0 auto 只对水平方向有用，垂直方向无效。

margin: auto;可以设置一个盒子在另一个盒子当中的水平居中。

### 2.盒子

### 居中和内容居中

**margin: 0 auto;和 text-align：center;的区别：**

text-align：center;作用：设置盒子中的文字和图片的水平居中。

margin: 0 auto;作用：让盒子自己居中。

**如果子元素的宽高大于了父元素的宽高**，使用这些设置将会无效。因此只能设置定位流。

除了定位流之外也可以用 margin：0 -100%；但是需要父元素设置text-aligin：center;

### 清空默认边距

blockquote, body, button, dd, dl, dt, fieldset, form, h1, h2, h3, h4, h5, h6, hr, input, legend, li, ol, p, pre, td, textarea, th, ul {

margin: 0;

padding: 0;

}

### 行高和字号

1. **注意点：**

1.1 盒子的高和行高不是一个概念。文字默认是在行高中垂直居中的。

1.2 在企业开发中，经常**把盒子的高度和行高设置为一样，以使文字在盒子中居中。**

1.3如果一个盒子中有多行文字，就不能设置盒子高度与行高一样，这是需要设置padding属性。

1. **设置行高：**

line-height

# 8.浮动

## 浮动流排版方式：

浮动流是一种半脱离的排版方式。

告诉浏览器不按照标准流进行排版，而是按照浮动流的排版方式排版，在父元素中的左部分显示元素，与父元素左边对齐，right则表示与父元素右边对齐。

**注意：**

**浮动流是没有居中对齐方式的，没有center这个属性值的。**

**在浮动流中是不能使用margin:0 auto；的。**

在浮动流中不区分块级元素，行内元素或者行内块级元素，并且这些元素都可以设置宽高。

## 浮动元素的脱标：

脱标：

**浮动元素脱离标准流。**

如果一个元素浮动流，那么这个元素相当于从标准流中删除了，之后的元素排版时不考虑脱标的元素。脱标的元素会覆盖后面的没有脱标的元素。

规则：

1. 相同方向上的浮动元素，先浮动的元素会显示在前面，后浮动的元素会显示在后面。
2. 不同方向上的浮动元素，左浮动会找左浮动，右浮动会找右浮动。
3. 浮动元素在浮动之后的位置，由浮动元素在浮动之前在标准流中的位置来确定。
4. 如果浮动元素前面的那个元素没有浮动，则该浮动元素会在上个未浮动的元素的下面进行浮动
5. 当父元素足够宽能够放下所有浮动元素的时候，这些浮动元素会并列显示，当不能放下所有浮动元素的时候，会从最后一个能放下的浮动元素开始在下一行向前贴近。
6. 浮动元素字围现象

浮动元素不会盖住未浮动元素的文字，文字会围绕在浮动元素的旁边而不会被盖住。可以用此来实现**图文混排效果**

1. 其他

一般左水平方向布局时使用浮动流，垂直方向布局时一般使用标准流，**布局网页时，先从上往下，再从左往右，从外向内。**

## 盒子高度问题

1.在标准流中内容的高度可以撑起盒子的高度

2. 在浮动流中浮动元素内容的高不可以撑起盒子的高

### 清除浮动：overflow:hidden;

**注意点:**

由于overflow:hidden可以撑起盒子的高度, 所以可以直接设置margin属性

IE8以前不支持利用overflow:hidden来清除浮动, 所以需要加上一个\*zoom:1;

实际上\*zoom:1能够触发IE8之前IE浏览器的hasLayout机制

优点可以不用添加额外的标签又可以撑起父元素的高度, 缺点和定位结合在一起使用时会有冲突

**\*zoom:1;和\_zoom:1的区别**

这个是hack写法，用来识别不同版本的IE浏览器

\_后面的属性只有IE6能识别

\*后面的属性 IE6 IE7能识别

**不推荐：**

给前面的父盒子添加高度（在企业开发中能不写高度就不写高度, 所以这种方式不常用

）

利用clear:both;属性清除前面浮动元素

使用clear:both之后margin属性会失效, 所以不常用

内墙法，外墙法

伪元素选择器

给前面的盒子添加伪元素来清除浮动

*给前面的盒子添加双伪元素来清除浮动注意点:*

*添加伪元素后可以撑起盒子的高度, 所以可以直接设置margin属性*

*先知道有这些方式, 原理需要学习到BFC和hasLayout才能明白*

*支持BFC的浏览器（IE8+，firefox，chrome，safari）通过创建新的BFC闭合浮动；*

*不支持 BFC的浏览器 （IE5-7），通过触发 hasLayout 闭合浮动。*

# 9.定位

## 9.1 相对定位

**相对定位就是相对于自己以前在标准流中的位置来移动**

格式:

position: relative;

**1. 相对定位是不脱离标准流的, 会继续在标准流中占用一份空间**

**2. 由于相对定位是不脱离标准流的, 所以在相对定位中区分块级元素/行内元素/行内块级元素**

**3. 由于相对定位是不脱离标准流的, 并且相对定位的元素会占用标准流中的位置, 所以当给相对定位的元素设置margin/padding等属性的时会影响到标准流的布局**

## 9.2绝对定位

**绝对定位就是相对于body或者某个定位流中的祖先元素来定位**

格式:

position: absolute;

**1. 绝对定位的元素是脱离标准流的, 不会占用标准流中的位置**

**2. 由于绝对定位的元素是脱离标准流的, 所以绝对定位的元素不区分块级元素/行内元素/行内块级元素**

**3. 如果一个绝对定位的元素是以body作为参考点, 那么其实是以网页首屏的宽度和高度作为参考点, 而不是以整个网页的宽度和高度作为参考点，相对于body定位会随着页面的滚动而滚动**

**4. 一个绝对定位的元素会忽略祖先元素的padding**

### 绝对定位水平居中

**1.注意当一个盒子绝对定位之后不能使用margin: 0 auto;让盒子自身居中**

**2.如果想让过一个绝对定位的盒子自身居中, 可以使用left: 50%; margin-left:-元素宽度一半px;**

## 子绝父相

子元素用绝对定位**，**父元素用相对定位

企业开发中一般相对定位和绝对定位都是一起出现, 很少单独使用

## 固定定位

什么是固定定位?

固定定位和前面学习的背景关联方式很像, 背景关联方式可以让某个图片不随着滚动条的滚动而滚动, 而固定定位可以让某个盒子不随着滚动条的滚动而滚动

格式:

**position: fixed;**

固定定位注意点:

固定定位的元素是**脱离标准流**的, 不会占用标准流中的位置

由于固定定位的元素是脱离标准流的, 所以绝对定位的元素**不区分**块级元素/行内元素/行内块级元素

IE6不支持固定定位

固定定位**应用场景:**

网页对联广告

网页头部通栏(穿透效果)

## 静态定位

什么是静态定位?

默认情况下标准流中的元素position属性就等于static, 所以静态定位其实就是默认的标准流

静态定位应用场景:

一般用于配合JS清除定位属性

## z-index属性

z-index**应用场景**

控制界面上的定位元素的覆盖关系, 例如网页中后面的定位元素不能覆盖前面的导航条通栏

什么是z-index值?

**用于指定定位的元素的覆盖关系**

定位元素的覆盖关系:

默认情况下定位的元素一定会盖住没有定位的元素

默认情况下写在后面的定位元素会盖住前面的定位元素

默认情况下所有元素的z-index值都是0, 如果设置了元素的z-index值, 那么谁比较大谁就显示在前面

定位元素的从父现象

父元素没有z-index值, 那么子元素谁的z-index大谁盖住谁

父元素z-index值不一样, 那么父元素谁的z-index大谁盖住谁，谁的父亲牛逼谁牛逼，拼爹。

# 10.图片和文字水平居中

**<div class="funbtnitem">  
 <img src="../../assets/img/shenhe.png" alt="">  
 <span>修改</span>  
</div>**

**一、通过vertical-align:middle实现现图片与文字水平对齐，需要区分html是行内元素，还是块状元素；例如：标签img、span是行内元素；标签p是块状元素则需要将标签p通过diaplay:inline-block;转化为行内元素；**

**1、实现img、p及div水平居中对齐，则只需要在img的css中的vertical-align：middle；及将p和div（display：inline-block）转化为行内元素；**

**2、实现img、span水平居中对齐，由于这两者都是行内元素；则只需要在img的css中的vertical-align：middle；span标签什么都不需要添加；**

**.funbtnitme {  
 margin-right: 10px;  
 font-size: 13px;  
}  
.funbtnitme img {  
 width: 15px;  
 height: 15px;  
 vertical-align:middle;  
}**

**二、通过flex布局实现图片与文字水平对齐**

**只需要在父级元素中css添加：**

**display:flex;**

**flex-direction:row;**

**align-items:center;**

**子级元素则不需要像上面那样添加vertical-align：middle；**

**.funbtnitme {  
 margin-right: 10px;  
 font-size: 13px;  
 text-align: center;  
 display: flex;  
 flex-direction: row;  
 align-items: center;  
}  
.funbtnitme img {  
 width: 15px;  
 height: 15px;  
}**

***来自 <***[***https://www.cnblogs.com/lyt0207/p/12761360.html***](https://www.cnblogs.com/lyt0207/p/12761360.html)***>***

# 11.a标签的伪类选择器

**专门用来修改a标签不同状态的样式。**

**标签(选择器)：状态{**

**属性。。。**

**}**

**a标签的状态：**

**link：未访问过**

**visited：已经访问过**

**hover：鼠标悬停（可以用在任何标签上）**

**active：长按**

**注意点：**

**1. a标签的伪类选择器可以单一出现也可以一起出现**

**2. 如果伪类选择器一起出现，有一个严格的顺序要求，必须先编写link，再编写visited，其次是hover，最后是active。如果顺序颠倒可能会没有效果。**

**3. 如果未访问和访问过的样式一样，伪类选择器可以缩写，即不写link和visited，直接对a标签设置属性。**

**4. 在企业开发中，伪类a标签的伪类选择器最好写在a标签选择器之面，*标签的盒子属性（显示模式，宽高，内外边距）直接写在a标签的选择器中*，a标签的背景，文字都写在a标签的伪类选择器中。**

# 12.过渡模块

**12.1 作用：用于样式变化时进行过渡**

### 12.2 transition：对象 时长（动画 延时）---【后面用，隔开】

**12.3 属性： transition-property:变化的属性名 ;**

**transition-duration:过渡时长 ;**

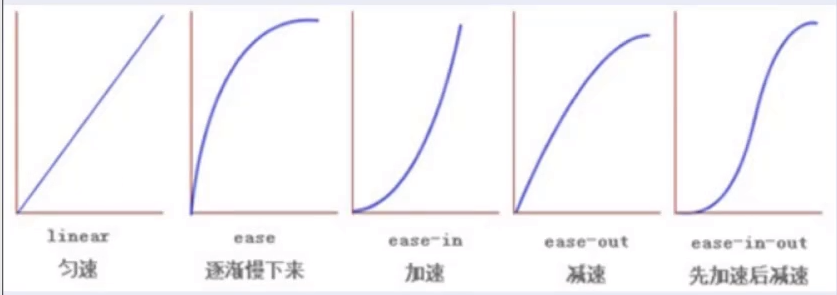
**transition-timing-function：动画**

**transition-delay：延迟**

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **CSS** |
| [**transition**](https://www.w3school.com.cn/cssref/pr_transition.asp) | **简写属性，用于在一个属性中设置四个过渡属性。** | **3** |
| [**transition-property**](https://www.w3school.com.cn/cssref/pr_transition-property.asp) | **规定应用过渡的 CSS 属性的名称。** | **3** |
| [**transition-duration**](https://www.w3school.com.cn/cssref/pr_transition-duration.asp) | **定义过渡效果花费的时间。默认是 0。** | **3** |
| [**transition-timing-function**](https://www.w3school.com.cn/cssref/pr_transition-timing-function.asp) | **规定过渡效果的时间曲线。默认是 "ease"。** | **3** |
| [**transition-delay**](https://www.w3school.com.cn/cssref/pr_transition-delay.asp) | **规定过渡效果何时开始。默认是 0。** | **3** |

|  |  |
| --- | --- |
| **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(*n*,*n*,*n*,*n*)** | **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。** |

**不同属性值对应的的变换速度：**

****

# 13.动画

## 13.1 过渡和动画之间的异同

**不同点：**

**过渡需要人为的触发，动画不需要人为触发就会自动执行**

**相同点：**

**过度和动画都是给元素添加动画的，并且都是系统新增属性，并且都需要满足三要素。**

## 13.2 animation-name属性

**格式：**

**animation：动画名称 动画时长 （动画运动速度 延迟时间 执行次数 往返动画）；**

**属性值的顺序与动画属性表中的属性顺序一样。**

**要素一：**

**animation-name：动画名称；**

**告诉系统我们需要创建一个叫做“动画名称的动画”，并且需要在其他地方写动画内容。**

**要素二：**

**@keyframes 动画名称{**

**from{**

**属性设置**

**}**

**to{**

**属性设置**

**}**

**}**

**或**

**用百分比替换from和to，0%代表from，100%代表to**

**中间过程可以是1%，25%，50%等等。**

**要素三：**

**animation-duration属性，设置动画持续的时间。**

**animation-delay属性，设置动画延迟多久开始**

|  |  |  |
| --- | --- | --- |
| **属性** | **描述** | **CSS** |
| [**@keyframes**](https://www.w3school.com.cn/cssref/pr_keyframes.asp) | **规定动画。** | **3** |
| [**animation**](https://www.w3school.com.cn/cssref/pr_animation.asp) | **所有动画属性的简写属性，除了 animation-play-state 属性。** | **3** |
| [**animation-name**](https://www.w3school.com.cn/cssref/pr_animation-name.asp) | **规定 @keyframes 动画的名称。** | **3** |
| [**animation-duration**](https://www.w3school.com.cn/cssref/pr_animation-duration.asp) | **规定动画完成一个周期所花费的秒或毫秒。默认是 0。** | **3** |
| [**animation-timing-function**](https://www.w3school.com.cn/cssref/pr_animation-timing-function.asp) | **规定动画的速度曲线。默认是 "ease"。** | **3** |
| [**animation-delay**](https://www.w3school.com.cn/cssref/pr_animation-delay.asp) | **规定动画何时开始。默认是 0。** | **3** |
| [**animation-iteration-count**](https://www.w3school.com.cn/cssref/pr_animation-iteration-count.asp) | **规定动画被播放的次数。默认是 1。** | **3** |
| [**animation-direction**](https://www.w3school.com.cn/cssref/pr_animation-direction.asp) | **规定动画是否在下一周期逆向地播放。默认是 "normal"。** | **3** |
| [**animation-play-state**](https://www.w3school.com.cn/cssref/pr_animation-play-state.asp) | **规定动画是否正在运行或暂停。默认是 "running"。** | **3** |
| [**animation-fill-mode**](https://www.w3school.com.cn/cssref/pr_animation-fill-mode.asp) | **规定对象动画时间之外的状态。** |  |

***来自 <***[***https://www.w3school.com.cn/css3/css3\_animation.asp***](https://www.w3school.com.cn/css3/css3_animation.asp)***>***

**animation-fill-mode属性值：**

|  |  |
| --- | --- |
| **值** | **描述** |
| **none** | **不改变默认行为。** |
| **forwards** | **当动画完成后，保持最后一个属性值（在最后一个关键帧中定义）。** |
| **backwards** | **在 animation-delay 所指定的一段时间内，在动画显示之前，应用开始属性值（在第一个关键帧中定义）。** |
| **both** | **向前和向后填充模式都被应用。** |

***来自 <***[***https://www.w3school.com.cn/cssref/pr\_animation-fill-mode.asp***](https://www.w3school.com.cn/cssref/pr_animation-fill-mode.asp)***>***

# 14.Transform （CSS 2D）

**14.1 作用:**对元素进行移动、缩放、转动、拉长或拉伸。

## 14.2 translate（X,Y）：移动

从其当前位置移动，向left（x 坐标） 和 top（y 坐标）移动

## 14.3 rotate(Xdeg)：旋转

默认顺时针旋转，X为度数，deg为单位

## 14.4 scale()：缩放

1. 指定（宽，高）

2. 倍数

## 14.5 skew()：翻转

通过 skew() 方法，元素翻转给定的角度，根据给定的水平线（X 轴）和垂直线（Y 轴）参数值 skew(30deg,20deg) 围绕 X 轴把元素翻转 30 度，围绕 Y 轴翻转 20 度。

## 14.6 matrix( )

matrix() 方法把所有 2D 转换方法组合在一起。

matrix() 方法需要六个参数，包含数学函数，允许您：旋转、缩放、移动以及倾斜元素。

*来自 <*[*https://www.w3school.com.cn/css3/css3\_2dtransform.asp*](https://www.w3school.com.cn/css3/css3_2dtransform.asp)*>*

## 注意：

如果要进行多个转换，需要用空格将值隔开，例如：

**Transform：rotate（30deg） translate（100px，0px）**

## 14.7 transform-origin形变中心点

**默认情况下所有的元素都是以自己的中心点为参考进行旋转的。**

**例：**

**transform-origin: 0px 0px;**

**修改形变中心为左上角，第一个参数代表水平方向，第二个为垂直方向。取值除了像素值之外，还有百分比为值和关键字，例如：center，top left等。**

## 14.8 旋转的轴向

**默认元素是按照z轴旋转的，看起来只是在平面上进行了旋转。**

**rotateX()：**

**以X轴为轴向进行旋转，同理还有rotateY()，rotateZ()。**

**透视：**

**在旋转时表现出近大远小的效果。由perspective属性控制，单位是像素。在设置x，y轴向旋转时必须添加此属性到被旋转元素的父元素中。**

## 14.9 阴影

**盒子阴影：**

**CSS3 边框阴影**

**在 CSS3 中，box-shadow 用于向方框添加阴影**

**div{  
box-shadow: 10px 10px 5px #888888;  
}**

***来自 <***[***https://www.w3school.com.cn/css3/css3\_border.asp***](https://www.w3school.com.cn/css3/css3_border.asp)***>***

**box-shadow属性值：水平偏移 垂直偏移 模糊度 阴影扩展 阴影颜色**

**注意：**

**1. 盒子阴影有内外之分，默认是外阴影。如果要设置内阴影需在属性值后面添加值 inset**

**2. 盒子阴影的默认颜色是由盒子内部内容决定的，阴影颜色默认跟随盒子中内容颜色。**

**3. 快速添加阴影只需设置：水平偏移 垂直偏移 模糊度**

**4. 文字阴影：**

**对比盒子阴影，*文字阴影只有 水平偏移，垂直偏移，模糊度，颜色四个属性值*，属性名与盒子阴影一样。同样的，颜色属性也可以省略，默认跟随文字颜色。**

## 14.10 元素的透视效果

**perspective属性，取值为数字+单位px;**

**给元素设置旋转时，如果没有设置透视，不会出现近大远小的效果。**

**若要实现此效果，需要给旋转元素的 *父元素* 设置透视属性.**

## 14.11 元素的不透明度

**opacity属性，在0到1之间取值，值为0时表示完全透明，为1时表示完全不透明。**

# 15.3D转换

15.1 元素转换为3D

给元素添加tranform-style属性，并设置值为preserve-3d即可将一个元素转换为3D元素。设置时需要将tranform-style属性设置在父元素。

15.2 设置立体图形

设置立体图形需要结合transform进行三维平移和旋转。

动画和默认样式中如果有同名的属性，动画中的属性会覆盖默认样式中的属性。

在编写动画中，不变的写在前面，需要变化的写在后面。

元素旋转后，转轴以外的轴向也会跟着旋转，就是说坐标系也会跟着旋转。

# 16.背景属性

## 1. 宽高属性

背景宽高由background-size属性控制，取值如下：

1.1 默认

默认情况下，背景图片会x,y两个方向平铺，图片有多大就显示多大。

1.2 具体像素

可以设置背景图片的具体宽高像素，如果宽高比与原图不一样会导致图片变形。

1.3 百分比

根据父元素的高宽的百分比设置背景图片的宽高

1.4 等比拉伸

在宽高中只设置一个属性的值，另一个属性会根据等比的规则系统自动设置，或者给另一个属性设置auto。

1.5 覆盖

给background设置值为cover，告诉系统图片需要等比拉伸，同时高度和宽度都填满整个元素为止。

**1.6 contain**

给background设置值为contain，告诉系统图片需要等比拉伸，同时高度或宽度其中一个填满整个元素为止。

## 2. 背景图片定位区域

显示区域由属性background-origin来设置，值有border-box（从外边框开始），padding-box（默认），centent-box（从内容区域开始）

默认情况下，背景图片是从padding区域开始显示的

## 3. 规定背景绘制区域

背景绘制区域由属性background-clip进行设置，同理，值有border-box，padding-box，centent-box三个

## 4. 多重背景图片

可以给背景设置多张图片。

格式：

background：url(xxx)，url(xxx)；

多张图片用逗号隔开即可

注意：

后添加的图片会盖住先添加的图片。

定位：

background：url(xxx) no-repeat left top，url(xxx) right top；

位置取值放在url之后并且在逗号之前即可定位背景图片。

也可以不同属性分开写，但是顺序要一致，同样要用空格隔开。

## 5. 设置背景图片居中

background-position: 50% 50%;