

CSS

一、目录

二、选择器

```
p {  
  
}
```

2.2, 类选择器

- 单类选择器

```
.类名 {  
    .....  
}  
<li class="类名"></li>
```

- 多类选择器

2.3, id选择器

```
/*只能被调用一次*/  
#id名 {  
    .....  
}  
<div id="id名">
```

2.4, 通配符选择器

```
* {  
    //(所有标签都改变)  
}
```

2.5, 复合选择器

2.5.1, 后代选择器

```
.son li a {  
    .....  
}
```

2.5.2, 子选择器

```
.father > a {  
    .....  
}
```

2.5.3, 并集选择器

```
div,p {  
    .....  
}
```

2.5.4, 伪类选择器

- 链接伪类选择器

```
a:link {.....} /*选择未被访问*/  
a:visited{.....} /*已被访问*/  
a:hover{.....} /*鼠标指针位于上方*/  
a:active{.....} /*活动链接（鼠标按下未弹起）*/
```

- 表单伪类选择器

```
/*focus~~~~*/  
input :focus {  
    .....  
}
```

跳转到[目录](#)

三、 CSS字体属性

font-family	字体（多字体逗号隔开，加引号）
font-size	字体大小（PX：像素）
font-weight	字体粗细（数字）{默认400（normal），最大700（bold）}
font-style	字体风格{italic：倾斜 normal: 普通}
font复合写法	{-style -weight -size/line-height -family}{ 顺序不可颠倒 }

跳转到[目录](#)

四、 文本属性

4.1, 文本颜色

color

4.2, 水平对齐

text-align

4.3, 装饰文本

text-decoration

- underline : 下划线
- overline : 上划线
- line-through : 删除线
- none : 取消 (无装饰线)

4.4, 文本缩进

text-indent

4.5, 行间距

line-height

跳转到[目录](#)

五、 引入方式

5.1, 内部样式表

5.2, 行内样式表

5.3, 外部样式表

使用标签在引入文件

```
<link rel="stylesheet" href="xxx.css">
```

跳转到[目录](#)

六、 Emmet语法

6.1, 快速生成HTML语法

1. Emmet 语法

1.1 快速生成HTML结构语法

1. 生成标签 直接输入标签名 按tab键即可 比如 div 然后tab 键，就可以生成 <div></div>
2. 如果想要生成多个相同标签 加上 * 就可以了 比如 div*3 就可以快速生成3个div
3. 如果有父子级关系的标签，可以用 > 比如 ul > li就可以了
4. 如果有兄弟关系的标签，用 + 就可以了 比如 div+p
5. 如果生成带有类名或者id名字的，直接写 .demo 或者 #two tab 键就可以了
6. 如果生成的div类名是有顺序的，可以用自增符号 \$
7. 如果想要在生成的标签内部写内容可以用 {} 表示

6.3, 快速格式化代码

Shift + Alt + F

跳转到[目录](#)

七、CSS的元素显示模式

7.1, 元素

7.1.1, 块元素

```
<div>.....<p><ul><li>
```

1. : 独占一行
2. : 可控制
3. : 默认父元素
4. : 里面可以放行内/块元素
5. : 文字标签无法放块元素

7.1.2, 行内元素

```
<span>.....<a><strong>
```

1. : 一行可显示多个
2. : 不可控制高度
3. : 默认本身容量宽度
4. : 行内只能放行内元素
 - o 链接不能放链接
 - o 中不能放块元素

7.1.3, 行内块元素

```
<img>....<input><td>
```

1. : 设置大小
2. : 一行放多个

7.2, 元素显示模式的转换

7.2.1, 行内--->块

display : block

7.2.2, 块--->行内

display : inline

7.2.3, ----->行内块

display : inline-block

跳转到[目录](#)

八、CSS背景

8.1, 背景颜色

background-color :

8.2, 背景图片

background-image : url(.....);

8.3, 背景平铺

background-repeat:

1. repeat
2. no-repeat
3. repeat-x
4. repeat-y

8.4, 背景图片位置

background position : x y

8.4.1, 方位名词

(前后顺序无关)

x: left / center / right

y: top / center / botton

8.4.2, 精确单位

background-position : x y (前后位置不可变)

8.4.3, 混合单位

使用时顺序不可变

8.5, 图像固定

background-attachment

- scroll : 滚动
- fixed : 固定

8.6, 背景复合写法

顺序: 图片、地址、平铺、滚动、位置

```
background: .....
```

8.7, 背景色半透明

```
background : rgba(0,0,0,0.3)/*a:透明度 0~1*/
```

跳转到[目录](#)

九、 盒子模型

9.1, 盒子模型组成

9.2, 边框

9.3, 表格的边框

9.4, 圆角边框

9.5, 盒子阴影

跳转到[目录](#)

十、 浮动

浮动的元素不会压住底下的图片和文字

十一、 定位

top	顶部
bottom	底部
left	左偏移
right	右便宜

11.1, 相对定位

position : relative

- 拖标后继续占有原有位置 (不拖标)
- 相对自身原来的位置移sa
- 限制绝对定位

11.2, 绝对定位

position : absolute

1. **以父级为定位约束** (如果没有父级或者父级没有定位, 则已浏览器为父元素)
2. 以最近一级带有定位的父元素为父级定位
3. **脱标后不在占有原有位置**

子绝父相

- 子级绝对定位
- 父级相对定位

11.3, 固定定位

position : fixed

1. **父级为浏览器可视窗口**, 和父元素无关系。
2. 脱标后不在占有原有位置
3. 宽度计算 : $\text{calc}(100\% - 100\text{px})$

11.4, 粘性定位

position : sticky

(部分位置固定, 部分移动)

- 拖标后继续占有原有位置
- 以浏览器的可视窗口为参照点
- 必须添加至少一个偏移值

11.5, 定位叠放次序

z-index : (int?)

只有定位的盒子才有此属性

11.6、属性

添加定位后可以直接设置宽高

11.7、元素的显示与隐藏

1、display

- display : none ; 隐藏对象
- display : block; 显示对象

隐藏后位置也不再占有

2、visibility

- visibility : hidden; 元素隐藏
- visibility : visible;元素可视

隐藏后位置继续占有

3、overflow 溢出

- overflow :

visible	显示所有内容，不隐藏
auto	显示滚动条
scroll	自动滚动条
hidden	隐藏多出来的部分

十二、高级技巧
