

层叠样式表 CSS

1. **内联样式（只在一句话用）** 将样式直接编写到 style 中（style="color=; font-size: px; "）
不方便复用，后期维护，尽量避免内联样式

2. **内部样式表（只在一种界面用）** 在<head>里面加上<style type="text/css ">

```
P{  
    color=;      这叫声明  
    font-size: px 这叫声明  
}
```

这全部叫做声明块

3. **把样式表写到外部 css 文件**中，然后用 link 标签了来引入

用<link rel="stylesheet " type="text/css " href="style.css ">

复用增加 利用浏览器的缓冲加快用户访问的速度 增长用户体验

Webstorm 的快捷键

! +Tab 快速生成标准模式

标签+Tab 快速生成标签组

Atl+/ 跳出提示

Ctrl+D 复制当行

Ctrl+/ 开启注释

Ctrl+Shife+/ 注释取消

ctrl + X: 删除行

ctrl + Y: 删除一行

ctrl + W: 选中单词

Ctrl+回车 无论在哪里都直接换行

迅速加</br> S+Enter

Ctrl+R 运行

上一个选择卡 Ctrl+Tab

Shift+Enter 重新开始新一行(换行)

F2 高亮错误或警告快速定位

Div.box\$*3+T 就出来三个 div box1-3

H200+Tb 就是 heigth:200px;

Alt+鼠标移动 多行修改

元素分两大类:

块元素和内联元素

块元素（独占一行的元素）：div h1-h6 p ul ol

div 没有任何语义，就是纯粹的块元素，并且没有任何的默认样式用于布局，默认透明色

行内元素（显示文本）：a img span

span 是一种行内元素，只占自身位置，不占一整行,专门用来选中文字，设置样式不能故意手动控制宽高

重点:

- 块元素用来做页面布局，行内元素用来选中文本样式，用块元素包含内联元素
- a 元素可以包含任意元素，但是除了本身不能嵌套
- p 元素不能包含任何块元素

px: 像素·点: 多少个小方格，大多数情况用像素值设置
百分比;自适应界面用%

常用选择器 1

● 元素选中器

用选择器可以设置界面指定元素的样式

语法: 元素 { } p{}/h{}/span{ }

● Id 选择器 (命名开头不能加数字)

用元素的 id 属性值选中唯一的一元素，不建议用复合选择器

语法: #名字{ } #id{ }

● 类选中器 (class 里面可以有多个选择器 用空格隔开)

用元素的 class 属性值选中一组元素

语法: .名字{ } .name{ }

引用: class=

● 选中器分组 (并集选择器)

选中器分组可以同时选中多种选择器

语法: 选中器 1, 选择器 2, 选择器 3 p,#id,.name{ }

● 通配选择器

可以用来选中页面所有元素

语法: *{ }

● 复合选择器

可以选择多种选择器的元素，满足交叉部分的,同时满足的

语法: 选择器 1 选择器 2 选择器 3 (标签写前面，一般元素以及类选择器放后面)

常用选择器 2

元素之间的关系

父元素: 直接包含元素的元素

子元素: 直接被父元素包含的元素

祖先元素: 直接或者间接包含后代元素的元素

后代元素: 直接或者间接被祖先后代元素包含的元素

● 后代元素选择器

选中指定元素的后代元素

语法: 祖先元素 后代元素{ }

例: 选中 id 为 d1 的 div 中的 p 元素中的 span 元素

#id p span{ }

● 子元素选中器

选中指定父元素的子元素

语法: 父元素>子元素

常用选择器 3

● 伪类选择器

专门用来表示元素的一种特殊状态

比如：访问过的超链接，普通的超链接，获取焦点的文本框

当我们需要为处在这些特殊状态的元素设置样式时可用伪类选择器

a 的伪类 (:link :visited :hover :active)

- 正常链接: a: link
- 已经访问的链接: a :visited(只能定义字体颜色)

其他标签也可以用

- 鼠标接触的连接: a:hover
- 鼠标接触正在点击的连接: a:active

重点:

- 涉及到 a 的四种伪类优先级是一样的，靠后的优先
- hover 不能写在 active 后面
- link 与 visited 顺序如何都不冲突，但是不能写在 hover 和 active 的后面
- 所以顺序要是 lvha 才避免出错。

Input 的伪类 (:focus ::selection)

使用 input 可以创建输入的文本输入框

语法: <input type="" name="" id="" value="" />

伪类样式:

:focus(文本框获取焦点后，修改背景颜色为红色)

```
input:focus{ background-color: red; }
```

::selection(为 p 标签选中之后变的颜色，就像用鼠标选中会黑色的那样子)

```
p::selection{ background-color: #FF0000; } (在火狐不显示)
```

```
p::-moz-selection{ background-color: #FF0000; } (moz 是火狐专用)
```

◆ 一般用 selection 在样式中写以上俩来避免浏览器不显示与不兼容

否定伪类

为所有p元素中除了 hello这值以外的其他元素

P:not(.hello){ }()里面是值，除了这值以外的其他样式

伪元素选择器

使用伪元素表示元素中的特殊位置

P:first-letter{ }在 p 标签中 首字母的样式设置

P:first-line{ }在 p 标签中 这一行的话,以句号为结束，样式设置

P:before{ content: " 最前面的" } 在文章最前面的话

P: after{ content: " 最前面的" } 在文章最前面的话

属性选择器

属性选择器

➤ 例子: title 属性, 任何标签都可以, 鼠标放在这上面会出现一小框, 来显示 title 的内容



可以依据元素中的属性或者属性值来选取指定的元素

语法:

```
[属性名] 选取含有指定属性的元素  
[属性名="属性值"] 选取含有指定属性值的元素  
[属性名^="属性值"] 选取属性值以指定内容开头的元素  
[属性名$="属性值"] 选取属性值以指定内容结尾的元素  
[属性名*="属性值"] 选取属性值以包含指定内容的元素
```

元素[属性名=值]{ }选取含有指定属性值的元素

```
p[title]{ background-color: yellow;}
```

p[title=hello]{ background-color: yellow;} 这标签的title, 并且要是hello才能用这样式

子元素选择器

在全部里面排序

:first-child{ } 可选择的第一只子元素

:first-child{ } 可选择的最好一只子元素

:nth-child (1-9) { } 可选择的第 1-9 只子元素 (括号里还可以写 even 表示偶数位置的子元素 odd 表示奇数的位置)

在标签里面排序

P:first-of-type{ } 是 P 中的第一只

:last-of-type{ } 是 P 中的最后一只

:nth-of-type () { } 是 P 中的第几只

以上和: first-of-child很相似, 只不过child是在所有的子元素第一只

而type是当前标签的第一只

后一只兄弟元素选择器

在span后面紧挨着的p span+p{ }

选择span后面的所有p span~p{ }

继承

利用继承可以将一些基本样式设置祖先元素, 所有后代元素自动继承祖先元素

并不是所有元素都会继承被子元素继承, 比如: 背景相关、边块、定位的都不能继承 •

选择器的优先级 (**style>id>class><标签>>*>继承**)

内联样式	优先级	1000、	(<code><p style="color=; font-size: px;"> </p></code>)
Id 选择器	优先级	100	<code>#id{ }</code>
类和伪类	优先级	10	<code>.name{}</code> class 和伪类
元素选择器	优先级	1	<code>p{ }/h{ }/span{ }</code>
通配*,	优先级	0	<code>*名字{ }</code>
继承的样式, 没有优先级			<code>asd sdf{ }</code>

- 当选择器中包含多种选择器时, 需要将多种选择器先相加, 再比较
- 如果选择器相同, 则选择靠后的选择器
- 并集选择器的, 是单独计算的, 选最大为此选择器的贡献值。
- 可以在样式最后加 (**! important**) 此时这样式是最高 的优先级

`background-color: yellow ! important;`

- 语义越明确优先级越高

属性

Li

列表的属性: `ul li ol`都是块元素, 列表都可以嵌套

通过**type**属性可以修改无序列表的项目符号

可选值:**disc** (圆点) **square** (实心的方块) **circle** (空心的圆)

一般不使用, 如果需要设置项目符号, 则可以选择**li**设置背景图片来设置

font;

font-family: 微软雅黑; 用来设置字体模式, 多个字体用 “, ” 隔开 前面优先使用

```
font-family: serif;">衬线字体: 我是一段文字, ABCDEFGabcdefg</p>
font-family: sans-serif;">非衬线字体: 我是一段文字, IHABCDEFGabcdefg<
font-family: monospace;">等宽字体: 我是一段文字, ABCDEFGabcdefg</p>
font-family: cursive;">草书字体: 我是一段文字, ABCDEFGabcdefg</p>
font-family: fantasy;">虚幻字体: 我是一段文字, ABCDEFGabcdefg</p>
```

衬线字体: 我是一段文字, ABCDEFGabcdefg

非衬线字体: 我是一段文字, ABCDEFGabcdefg

等宽字体: 我是一段文字, IHABCDEFGabcdefg

草书字体: 我是一段文字, ABCDEFGabcdefg

虚幻字体: 我是一段文字, **ABCDEFGabcdefg**

font-style: Italic (斜体)
font-variant: bold 字体加粗
font-variant: small-caps
font-color: ;
font-size::;

font可以直接设置全部样式（用空格隔开）

✧ 语法: font: italic bold 30px "微软雅黑";

注意:

- 使用 font 时: 斜体 加粗 大小写字母 没有顺序 可写可不写
- 如果不写则使用默认的值, 但是要求文字的大小和字体必须写
- 字体必须是最后一个样式, 大小必须是第二个, 否则不成立

行间距

Line-height (单线本子一样 没一行的空间都一样, 字体居中显示)

- 行间距=行高-字体大小

可接收的值:

- 可直接就收一个大小 (line-height: 11px)
- 可接收百分数, 相对于字体去计算行高
- 可接收一个数值, 是字体的几倍 (line-height: 2) 就是字体的两倍

✧ 小技巧:

对于单文本来讲, 可以将行高设置为与父元素高度一致

```
width: 200px; <!--父元素 -->
```

```
height: 200px;
```

```
background-color: lightcoral;
```

```
line-height: 200px; <!--行高 -->
```

这样可以使行文本在父元素垂直居中, 如图



在字体后也可以设置行高

```
font: 100px/200px "微软雅黑"; /* 字体 100 行高 200 行间距 100 */
```

注意: line-height 有字体大小时不能写在 font 的前面会覆盖

文本:

```
text-transform: capitalize; /*通过空格来识别字母 首字母都大写*/
```

```
text-transform: uppercase; /*所有字母都大写*/
```

```
text-transform: lowercase; /*所有字母都小写*/
```

```
text-decoration: underline; /*下划线*/
```

```
text-decoration: overline; /*上划线*/
```

```
text-decoration: line-through; /*中划线*/
```

```
text-align: center; /*文本居中对齐显示*/
```

```
text-align: left; /*文本靠右对齐显示*/
```

```
text-align: right; /*文本居左对齐显示*/
```

```
text-align: justify; /*文本两边对齐显示*/
```

text-indent:10 px /*缩进 10 个字符*/
spacing 控制字符的空格来控制字符的简介
letter-spacing: 1px; /*字符间距*/
word-spacing: 5px; /*单词间的距离*/