**CSS层叠样式表（二）**

# 1. Emmet语法

Emmet语法前身是Zen coding,它使用缩写来提高html/css的编写速度，Vscode内部已经集成该语法。

## 1.1 快速生成HTML结构语法

* 生成标签 直接输入标签名 按tab键即可(下方适用)
* 如果想生成多个相同标签 加上\*即可 例如 div\*3
* 如果有父子级关系的标签，可以用> 比如 ul > li
* 如果是兄弟关系的标签，用+ ，例如 div+p
* 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。
* 如果生成的div类名是有顺序的，可以用自增符合$
* 如果想要在生成的标签内部写内容可以用{}表示。

## 1.2 快速生成CSS样式语法

CSS基本采取简写形式即可

* 比如w200 按tab 可以生成 width: 200px;
* 比如lh26 按tab 可以生成line-height: 26px;

## 1.3 格式化代码

Vscode快速格式化代码: shift+alt+f

也可以设置当我们保存页面的适合自动格式化代码。

1）文件------> 首选项 ------> 设置

2） 搜索emmet.include;

3） 在settings.json下的[用户]加入以下语句

“editor.formatOnType”:true,

“editor.formatOnSave”:true

# 2. CSS的复合选择器

## 1.1 css复合选择器介绍

在CSS中，可以根据选择器的类型把选择器分为基础选择器和复合选择器，复合选择器是建立在基础选择器之上，对基本选择器进行组合形成的。

* 复合选择器可以更准确，更高效的选择目标元素（标签）
* 复合选择器是由两个或多个基础选择器，通过不同的方式组合而成的。
* 常用的复合选择器包括： 后代选择器，子选择器，并集选择器，伪类选择器等等。

## 1.2 后代选择器（重要）

后代选择器又称为包含选择器，可以选择父元素里面的子元素，其写法就是把外层标签写在前面，内层标签写在后面，中间用空格隔开。当标签发生嵌套式，内层标签就称为外层标签的后代。

语法：

|  |
| --- |
| 元素1 元素2 { 样式声明 } |

上述语法表示选择元素1里面的所有元素2（后代标签）可以多层嵌套使用

文本

描述已自动生成

## 1.3 子选择器（重要）

子元素选择器（子选择器）只能选择作为某元素的最近一级子元素。简单理解就是选亲儿子元素。

语法：

|  |
| --- |
| 元素1 > 元素2 { 样式声明 } |

上述语法表示选择元素1里面所有直接后代（子元素）元素2.

例如：

|  |
| --- |
| div > p { 样式声明 } |

* 元素1和元素2中间用大于号隔开
* 元素1是父级，元素2是子级，最终元素是元素2
* 元素2必须是亲儿子，其孙子、重孙之类都不归他管，你也可以叫他亲儿子选择器。

文本

描述已自动生成

## 1.4 并集选择器（重要）

并集选择器可以选择多组标签，同时为他们定义相同的样式，通常用于集体声明。

并集选择器是各选择器通过英文逗号（,）连接而成，任何形式的选择器都可以作为并集选择器的一部分。

语法：

|  |
| --- |
| 元素1, 元素2 { 样式声明 } |

上述语法表示选择元素1和元素2

例如：

|  |
| --- |
| ul,  div { 样式声明 } /\* 选择ul和div标签元素 \*/ |

* 元素1和元素2中间用逗号隔开。
* 按照语法约定，并集选择器喜欢竖着写。
* 逗号可以理解为和的意思
* 并集选择器通常用于集体声明。

## 1.5 伪类选择器

伪类选择器用于向某些选择器添加特殊的效果，比如给链接添加特殊效果，或选择第一个，第n个元素。

伪类选择器书写的最大特点是用冒号（：）表示，比如：hower: first-child.

因为伪类选择器很多，比如有链接伪类，结构伪类等，所以这里先给大家讲解常用的链接伪类选择器。

## 1.6 链接伪类选择器

|  |
| --- |
| a: link /\* 选择所有未被访问的链接\*/  a: visited /\* 选择所有已被访问的链接 \*/  a: hover /\* 选择鼠标指针位于其上的链接 \*/  a: active /\* 选择活动链接（鼠标按下未弹起的链接）\*/ |

文本

描述已自动生成

伪类选择器注意事项

1. 为了确保生效，请按照LVHA的顺序声明：link -> visited->hover -> active

2. 记忆法：love hate

3. 因为a链接在浏览器中具有默认样式，所以我们实际工作中都需要给链接单独指定样式。

* 链接伪类选择器实际工作开发中的写法：

|  |
| --- |
| /\* a是标签选择器 所有的链接 \*/  a {  color: gray;  }  /\* :hover 是链接伪类选择器 鼠标经过 \*/  a: hover {  color: red; /鼠标经过是，由原来的灰色变成了红色  } |

## 1.7 :focus伪类选择器

:focus 伪类选择器用于选取获取焦点的表单元素。

焦点就是光标，一般情况下<input>类表单元素才能获取，因此这个选择器也主要针对于表单元素来说。

|  |
| --- |
| input: focus {  background-color: yellow;  } |

## 1.8 复合选择器总结

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 选择器 | 作用 | 特征 | 使用情况 | 隔开符号及用法 |
| 后代选择器 | 用来选择后代元素 | 可以是子孙后代 | 较多 | 符号是**空格**  .nav a |
| 子代选择器 | 选择最近的一级元素 | 只能选亲儿子 | 较多 | 符号是**大于**  .nav > p |
| 并集选择器 | 选择某些同样式样的元素 | 可以用于集体声明 | 较多 | 符号是**逗号**  .nav,.header |
| 链接伪类选择器 | 选择不同状态的链接 | 跟链接相关 | 较多 | 重点记住a{}和a:hover实际开发的写法 |
| :focus选择器 | 选择获得光标的表单 | 跟表单相关 | 较少 | input:focus记住这个写法。 |

# 3. CSS的元素显示模式

## 3.1 什么是元素的显示模式

作用：网页的标签非常多，在不同类型的标签，了解他们的特点更好的布局我们的网页。

元素显示模式就是元素（标签）以什么方式进行显示，比如<div>自己占一行，比如一行可以放多个<span>。

HTML元素一般分为块元素和行内元素两种类型。

## 3.2 块元素

常见的块元素有<h1>~<h6>、<p>、<div>、<ul>、<ol>、<li>等，其中<div>标签是最典型的块元素。

**块级元素特点：**

* 比较霸道，自己独占一行
* 高度，宽度，外边距以及内边距都可以控制
* 宽度默认是容器（父级宽度）的100%
* 是一个容器及盒子，里面可以放行或者块级元素。

注意：

* 文字类的元素内不能使用块级元素
* <p>标签主要用于存放文字，因此<p>里面不能放块级元素，特别是不能放<div>
* 同理，<h1>~<h6>等都是文字类块级标签，里面也不能放其他块级元素。

## 3.3 行内元素

常见的行内元素有<a>,<strong>,<em>,<i>,<del>,<s>,<ins>,<u>,<span>等，其中<span>标签是最典型的行内元素，有的地方也将行内元素称为内嵌元素。

行内元素的特点：

* 相邻行内元素在一行上，一行可以显示多个。
* 高，宽直接设置无效的。
* 默认宽度就是它本身内容的宽度。
* 行内元素只能容纳文本或其他行内元素。

注意：

* 链接里面不能再放链接
* 特殊情况链接<a>里面可以放块级元素，但是<a>转换下一块级模式最安全。

## 3.4 行内块元素

在行内元素中有几个特殊的标签------<img/>,<input/>,<td>，它们同时具有块元素和行内元素的特点，有些资料称它们为行内块元素。

行内块元素的特点：

* 和相邻行内元素（行内块）再一行上，但是他们之间会有空白缝隙，一行可以显示多个（行内元素特点）
* 默认宽度就是它本身内容的宽度（行内元素特点）
* 高度，行高，外边距以及内边距都可以控制（块级元素特点）

## 3.5 元素显示模式转换

特殊情况下，我们需要元素模式的转换，简单理解：一个模式的元素需要另外一种模式的特性。

比如想要增加链接<a>的触发范围。

* 转换为块元素： display: block;
* 转换为行内元素：display:inline;
* 转化为行内块：display: inline-block

## 3.6 一个小工具的使用 snipaste

snipaste是一个简单但强大的截图工具，也可以让你将截图贴回到屏幕上。

常用快捷方式

* F1可以截图，同时测量大小，设置箭头，书写文字等
* F3在桌面置顶显示
* 点击图片，alt可以取色（按下shift可以切换取色模式）
* 按下esc取消图片显示。

## 3.7 一个小技巧 单行文字垂直居中的代码

CSS没有给我们提供文字垂直居中的代码，这里我们可以使用一个小技巧来实现。

解决方案：让文字的行高等于盒子的高度 就可以让文字在当前盒子内垂直居中。

line-height 和 height的值一样

## 3.8 单行文字垂直居中的原理

|  |  |
| --- | --- |
|  | 上空隙 |
| 行高 | 文字本身高度 |
|  | 下空隙 |

简单理解：行高的上空隙和下空隙把文字挤到中间了，如果行高小于盒子高度，文字会偏上，如果行高大于盒子高度，则文字偏下。

# 4. CSS的背景

通过CSS背景属性，可以给页面元素添加背景样式。

背景属性可以设置背景颜色，背景图片，背景平铺，背景图片位置，背景图像固定等。

## 4.1 背景颜色

background-color属性定义了元素的背景颜色。

|  |
| --- |
| background-color: 颜色值; |

一般情况下元素背景颜色默认值是transparent（透明），我们也可以指定背景颜色为透明色。

## 4.2 背景图片

background-image属性描述了元素的背景图像，实际开发常见logo或者一些装饰性 小图片或者是超大的背景图片，优点是非常便于控制位置（精灵图也是一种运用场景。）

|  |
| --- |
| background-image: url(url路径); |

|  |  |
| --- | --- |
| 参数值 | 作用 |
| none | 无背景图（默认的） |
| url | 使用绝对或相对地址指定背景图片 |

## 4.3 背景平铺

如果需要在HTML页面上对背景图片进行平铺，可以使用backgroud-repeat属性。

|  |
| --- |
| background-repeat: repeat | no-repeat | repeat-x | repeat-y |

|  |  |
| --- | --- |
| 参数值 | 作用 |
| repeat | 背景图像在纵向和横向上平铺（默认的） |
| no-repeat | 背景图片不平铺 |
| repeat-x | 背景图像在横向上平铺 |
| repeat-y | 背景图片在纵向平铺 |

## 4.4 背景图片位置

利用background-position属性可以改变图片在背景中的位置。

|  |
| --- |
| background-position: x y; |

参数代表的意思是：x坐标和y坐标，可以使用方位名词或者精确单位

|  |  |
| --- | --- |
| 参数值 | 说明 |
| length | 百分数 | 由浮点数字和单位标识符组成的长度值 |
| position | top | center | bottom | left | right |

1.参数是方位名词

* 如果指定的两个值都是方位名词，则两个值前后顺序无关，比如left top和top left效果一致。
* 如果只指定了一个方位名词，另一个值省略，则第二个值默认居中对齐。

2. 参数是精确单位

* 如果参数值是精确左边，那么第一个肯定是x坐标，第一定是y坐标。
* 如果只写一个，那么第一个值是x，那么第二个值默认垂直居中。

3. 参数是混合单位

* 如果指定的两个值是精确单位和方位名词混合使用，则第一个值是x坐标，第二个值是y坐标

文本

描述已自动生成

图片包含 图形用户界面

描述已自动生成

## 4.5 背景图片固定（背景附着）

background-attachment属性设置背景图像是否固定或者随着页面的其余部分滚动。

background-attachment后期可以制作视差滚动的效果。

|  |
| --- |
| background-attachment: scroll | fixed |

|  |  |
| --- | --- |
| 参数 | 作用 |
| scroll | 背景图像是随对象内容滚动 |
| fixed | 背景图像时固定的 |

## 4.6 背景复合写法

为了简化背景属性的代码，我们可以将这些属性合并简写在同一个属性background中，从而节约代码量。

当使用简写属性时，没有特定的书写顺序，一般习惯约定顺序为：

|  |
| --- |
| background: 背景颜色 背景图片地址 背景图像滚动 背景图片位置  background: transparent url(image.jpg) repeat-y fixed top; |

这是实际开发中，我们更提倡的写法。

## 4.7 背景色半透明

CSS3为我们提供了背景半透明的效果

|  |
| --- |
| background: rgba(0,0,0,0.3); |

* 最后一个值时alpha透明度，取值范围是(0~1)
* 我们习惯把0.3的0省略掉，写为background: rgba(0,0,0.3);
* 注意：背景半透明是指盒子背景半透明，盒子里面的内容不受影响
* CSS新增属性，是IE9+版本浏览器才支持的。
* 但是现在实际开发，我们不太兼容性写法了，可以放心使用。

# 5. CSS的三大特性

## 5.1层叠性

相同选择器给设置相同的样式，此时一个样式就会覆盖（层叠）另一个冲突的样式。层叠性主要解决样式冲突的问题。

层叠性原则：

* 样式冲突，遵循的原则是就近原则，哪个样式里结构近，就执行哪个样式。
* 样式不冲突，不会层叠。

## 5.2 继承性

CSS中的继承: 子标签会继承父标签的某些样式，如文本的颜色和字号。

屏幕上有字

描述已自动生成

* 恰当地使用继承可以简化代码，降低CSS样式的复杂性
* 子元素可以继承父元素的样式（text-，font-,line-这些元素开头的可以继承，以及color属性）

**行高的继承**

|  |
| --- |
| body {  font:12px/1.5 Microsoft YaHei ;  } |

* 行高可以跟单位也可以不跟单位
* 如果子元素没有设置行高，则会继承父元素的行高为1.5
* 此时子元素的行高是：当前子元素的文字大小\*1.5
* body行高1.5 这样写法最大的优势就是里面子元素可以根据自己文字大小自动调整行高。

## 5.3 优先级

当同一个元素指定多个选择器，就会有优先级的产生。

* 选择器相同，则执行层叠性
* 选择器不同，则根据选择器权重执行。

**选择器权重如下：**

表格

描述已自动生成

文本

描述已自动生成

**优先级注意点**

* 权重是由4组数字组成，但是不会有进位。
* 可以理解为类选择器永远大于元素选择器，id选择器永远大于类选择器，以此类推
* 等级判断从左向右，如果某一位数值相同，则判断下一位数值。
* 简单记忆：通配符和继承权重为0，标签选择器为1，类（伪类）选择器为10，id选择器100，行内样式表为1000，!importatn无穷大。
* 继承的权重是0，如果该元素没有直接选中，不管父元素权重多高，子元素得到的权重都是0。

权重叠加：如果是复合选择器，则会由权重叠加，需要计算权重。

* div ul li ----->0,0,0,3
* .nav ul li ------>0,0,1,2
* a:hover ------> 0,0,1,1
* .nav a --------> 0,0,1,1