# DAY05 CSS

# 1 CSS优先级

### 不同选择器不同样式

|  |
| --- |
| 当多个不同的选择器选中同一个元素，发生冲突时，如果设置了不同的样式会叠加效果同时生效  p {color:red;}  .txt {background-color: yellow;}  <div>  <p class="txt">测试文字</p>  </div> |

### 相同选择器相同样式

|  |
| --- |
| 相同的选择器，会采用顺序读取的原则，后来的会覆盖之前的  但前提：要渲染的样式相同，值不同  .txt {background-color: yellow;}  .txt {background-color: green;}  <div>  <p class="txt">测试文字</p>  </div> |

### 最近的祖先样式要比其他祖先样式优先级高

|  |
| --- |
| 在CSS中，有一些属性是具有继承特性的，也就是父元素的CSS属性会传递到子元素上  使用的就是就近原则，哪个近，就应用哪个样式  <div style="color: green;">  <div style="color: blue;">  <div>测试文字3</div>  </div>  </div> |

### 指定样式大于继承样式

|  |
| --- |
| <div style="color: pink;">  <span style="color:gold">测试文字4</span>  </div> |

### 选择器不同的权值

|  |  |
| --- | --- |
| **选择器** | **权值** |
| !important | 10000 |
| style=“”行内样式 | 1000 |
| id选择器 | 0100 |
| 类 伪类 | 0010 |
| 标签名 伪元素 | 0001 |
| 通用\* 子选择器> 相邻兄弟+ | 0000 |
| 继承样式 | 无 |

### 原则

|  |
| --- |
| **权值大的优先渲染**  !important>行内样式>ID>(类选择器=伪类选择器)>标签名选择器>通用选择器>继承  **权值会累加，但不会越级！**  比如11个类选择器，值不是10\*11=110,而是99.9不会越级到下一级  **记忆诀窍：**  选择器控制范围越大的，优先级越低 |

# 2 选择器

### 7. 并列选择器

|  |
| --- |
| 选中可以同时被多个选择器选中的元素  注意：多个选择器之间不能有空格，必须紧挨着写！  /\* 选中span元素且class值为danger\*/  span.danger{color:red;}  /\* 选中class值同时具有btn与succ的元素\*/  .btn.succ {color: green;} |

### 8. 后代(包含)选择器

|  |
| --- |
| 选中某个元素内的儿子、孙子、重孙子….元素  祖先选择器 后代选择器 { }  div span { } 选中div下的所有span  易错点：空格容易忘记写  使用场景：内部结构比较简单，没有那么多相同标签 |

### 9. 直接子代选择器

|  |
| --- |
| 选中元素下面的直接儿子元素  div > span { } 选中div下的所有儿子span |

### 10. 兄弟选择器

|  |
| --- |
| 选中元素后面的兄弟元素——不要前面的兄弟，也不要子代  p~span { } 选中p元素后面的span兄弟元素 |

### 11. 相邻兄弟选择器

|  |
| --- |
| 选中元素后面紧挨着的相邻兄弟元素——不要前面的，也不要有间隔的，也就是最多选中1个  p+span { } 选中p元素后面紧挨着的那一个span兄弟元素 |

### 12.伪元素

我们可以给指定元素的前面/后面添加内容，还可以对添加的内容进行样式设置

|  |
| --- |
| ::before 指的是在元素最开始的位置插入内容  ::after 指的是在元素最末尾的位置插入内容  li::before {  content:"子曰:";  }  li::after {  content: "--摘自《论语》";  color: gold;  }  注意:content属性必须写！如果暂时不确定生成的内容，可以写成content:""; |
| **面包屑导航 breadcrumb**  a+a::before {  content: ">";  color: red;  }  <!-- 首页>学习用品>笔记本电脑>戴尔燃7000 -->  <div class="nav">  <a href="#">首页</a>  <a href="#">学习用品</a>  <a href="#">笔记本电脑</a>  <a href="#">戴尔燃7000</a>  </div> |

# 3 CSS中的值

### CSS中的色值

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 颜色 | 单词表示法 | RGB表示法(24位色) | RGBA表示法(32位色) | 十六进制(哈希）表示法 | 十六进制缩写表示法 |
| 红色 | red | rgb(255,0,0) | rgba(255,0,0,1) | #FF0000 | #F00 |
| 绿色 | green | rgb(0,255,0) | rgba(0,255,0,1) | #00FF00 | #0F0 |
| 蓝色 | blue | rgb(0,0,255) | rgba(0,0,255,1) | #0000FF | #00F |
| 青色 | cyan | rgb(0,255,255) | rgba(0,255,255,1) | #00FFFF | #0FF |
| 品红 | magenta | rgb(255,0,255) | rgba(255,0,255,1) | #FF00FF | #F0F |
| 黄色 | yellow | rgb(255,255,0) | rgba(255,255,0,1) | #FFFF00 | #FF0 |
| 白色 | while | rgb(255,255,255) | rgba(255,255,255,1) | #FFFFFF | #FFF |
| 黑色 | black | rgb(0,0,0)什么颜色都没有 | rgba(0,0,0,1) | #000000 | #000 |
| 中灰 | gray | rgb(128,128,128)256/2 | rgba(128,128,128,1) | #808080 | #808080 |

### CSS中的尺寸

|  |
| --- |
| px像素(Pixel)屏幕中最小的一个发光点，网页种经常使用 |
| % 百分比，指的是占父级元素的尺寸占比，参考的是父级元素的大小 |
| 数字 没有任何单位，比如不透明度 |
| 角度值 deg 0~360,也可以有负数 比如270deg=-90deg |
| 时间单位 1s=1000ms |
| vw / vh APP开发时会用到的视口单位 |
| em倍率 / rem根倍率单位  1em指的是1倍当前字体的大小  1rem指的是1倍根元素(html)的字体大小 |

# 4. 页面元素的显示模式

一个HTML元素可以有多种显示模式

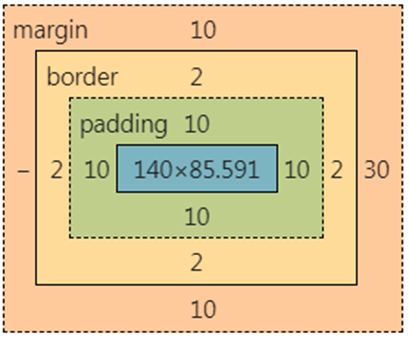
|  |
| --- |
| **块级元素——display:block;**  常见标签: div h1~h6 p from table ul ol li...  排列方式：按照书写的顺序，从上往下依次排列  独占一行(一行指的是父元素宽度的100%)，还可以自设宽高 |
| **行内元素——dispaly:inline;**  常见标签:span a i b mark...  排列方式：按照书写顺序从左往右依次排列，可以和其他内容同行，一行放不下才显示在下一行  无法设置宽度和高度，行内元素的大小是靠内容撑起来的  img元素是特殊的行内元素，它有自己的宽高，还可以设置宽高 |
| **行内块元素——display:inline-block;**  常见标签：input button  既有行内元素的特点，也有块级元素的特点  既可以与其他元素共处一行(行内)，也可以设置宽高(块级) |

# 流与文档流

|  |
| --- |
| **流(stream):** 车流 水流  多个元素的有序排列称为流 |
| **文档流:**  网页中多个元素按照自己的特性有序排列，形成了文档流  比如:块级元素是从上到下依次排列，行内与行内块元素从左到右依次排列  **文档流中元素的特点：**   1. 依次紧密排列，中间没有大的空白，多个div/span在竖直/水平方向紧挨着 2. 流中的元素不会出现一个元素叠摞在另一个元素上面的情况 3. 如果项目中需要出现叠摞的现象，就只能脱离文档流 |

# CSS盒子模型(Box Model)

### 默认的盒子模型

一个HTML元素，需要占用页面的空间。

内容、边框、内容与边框之间的距离、不同元素直接的距离都需要占用页面空间

所以我们需要了解具体的计算方式

|  |
| --- |
| content: 内容区域  padding: 内间距—元素内容与边框之间的距离  border: 元素的边框  margin: 外间距—元素与元素之间的距离 |
| 涉及到四个方向:left左 right右 top上 bottom下 |
| **一个元素在水平方向上占用的总空间：**  中文：  英文： |
| **一个元素在垂直方向上占用的总空间：**  中文：  英文： |

# 作业

1. 自行总结并记忆HTML元素的三种显示模式和它们的特点
2. 自行总结归纳 默认盒子模型总大小的计算公式，要求：水平+竖直+中文+英文
3. 完成汉唐首页导航栏
4. 完成“华硕无双”商品卡片
5. CSS属性总结表自己完成