一.CSS3概述

1.什么是css

|  |
| --- |
| css: cascading style sheets  层叠 样式 表  层叠样式表，级联样式表，简称样式 |

2.作用

|  |
| --- |
| HTML：负责网页结构的搭建，内容的展示（骨架）  CSS：负责网页的修饰(亚洲4大邪术) |

3.CSS和html属性的使用原则

|  |
| --- |
| W3C建议尽量使用css代替html属性  1.css可以让样式重用(有限)，html属性不能重用  2.代码重用，提高了可维护性 |

二.css的语法规范

|  |
| --- |
| color:颜色的英文单词  background-color:颜色的英文单词  font-size:以px为单位的数字 |

1.css的使用方式

①行内样式,内联样式

|  |
| --- |
| <any style=”样式声明1; 样式声明2”></any>  样式声明：样式属性：值 color:red  内联样式，项目中基本不用  因为：样式代码不能重用  内联样式的优先级最高  只有在学习和测试的时候使用 |

②内部样式

|  |
| --- |
| 在head标签中，添加style标签  在<style></style>中，编写样式规则  样式规则： 选择器{ css属性：值；css属性：值 }  内部样式，可以在本页面中重用，重用有局限性  项目中，使用较少，学习和测试经常使用 |

③外部样式

|  |
| --- |
| 单独创建.css文件  在head中使用link标签，引入这个css文件  能够做到代码的广泛重用，项目就用外部样式  注意 <link rel="stylesheet" href="my.css"> 必须写，不写失效 |

2.css样式特征

①继承性

|  |
| --- |
| 大部分的css效果可以直接被子元素继承 |

②层叠性

|  |
| --- |
| 可以为一个元素定义多个不冲突的样式。  都可以作用到这个元素上 |

③优先级

|  |
| --- |
| 如果一个元素，有多个样式，发生了冲突  按照css默认优先级显示  默认优先级规则  1.内联样式优先级最高  2.就近原则(内部样式，外部样式)  3.浏览器默认样式最低 |

④调整样式优先级

|  |
| --- |
| !important规则，写在；和值之间，使用空格和值分开  添加了!important的css属性，直接获得最高优先级  比内联都高  内联样式不允许添加!important |

三.基础选择器(重点)

|  |
| --- |
| 规范了页面中哪些元素可以使用定义好的样式  选择器就是一个条件，符合这个条件的元素，可以使用定义的样式 |

1.选择器详解

①通用选择器

|  |
| --- |
| \*{样式声明}  页面中所有元素，都要使用这个样式  \*效率低，尽量少用  \*{margin:0;padding:0} 所有元素内外边距清零 |
| 注意，浏览器厂商很多，每种浏览器都会默认给元素一些基础样式  导致，同一篇代码在不同浏览器上，显示效果不同  开发者，希望同一篇代码在不同浏览器，效果一致  css reset，样式重置，把容易发生分歧的css效果，统一化 |

②元素选择器，标签选择器

|  |
| --- |
| elem{样式声明}  页面中所有elem元素，都会执行这个样式 |

③专属定制，ID选择器

|  |
| --- |
| <div id="d1">Lorem</div>  #d1{color:red;} |

④类选择器

|  |
| --- |
| 定义页面上某类元素的样式  定义的是公共样式，谁想用就能用  通过class属性来引用  .类名{样式声明}  <any class=”类名”> 声明的时候有点，引用的时候没有点  类选择器的特殊使用方式  1.多类选择器  <any class=”c1 c2 c3”>一个元素引用多个类名  2.分类选择器  元素选择器+类选择器{样式声明}  p.text-red{….}  类选择器+类选择器{样式声明}  .border-red.text-red{…..} |

⑤群组选择器

|  |
| --- |
| 定义一些选择器的公有样式  选择器1,选择器2,选择器3……{样式声明} |

⑥后代选择器

|  |
| --- |
| 根据元素的后代关系：一层或多层嵌套  更精准的找到的元素  选择器1 选择器2 选择器3….{样式声明} |

⑦子代选择器

|  |
| --- |
| 根据元素的子代关系，精准的找到元素  子代，一层嵌套结构  选择器1>选择器2….{样式声明} |

⑧伪类选择器

|  |
| --- |
| 匹配的是元素某种状态，在这种状态应用的样式  选择器:link{} 链接未被访问的状态  选择器:visited{} 链接已被访问过的状态  选择器:hover{} 鼠标悬停在元素上的状态  选择器:active{} 元素被激活时的状态  选择器:focus{} 元素获取焦点时的状态 |

⑨选择器的权值

|  |
| --- |
| 权值：标识当前选择器的重要程度，权值越大，优先级越高  !important >1000  内联样式 1000  id选择器 100  类选择/伪类 10  元素选择器 1  \*通用 0  继承的样式 无  权值的特点：  1.当一个选择器中含有多个选择器时，需要将所有选择器的权值相加，权值大的优先应用  2.权值相同，就近原则  3.!important直接获取最高权值，不能用在内联样式中  4.群组选择器的权值，单独计算，不能相加  5.选择器权值的计算，不会超过自己的最大数量级  （100个1，加一起也小于10） |

四.尺寸和边框

1.尺寸属性

|  |
| --- |
| 设置元素的宽度和高度  属性  width：  取值 1.以px为单位的数字  2.%相对于父元素的宽度的百分比  max-width: 1.px为单位的数字  2.%是相对自己本身大小的百分比  min-width:  ex:max-width:100%;我可以缩小，但是我最大是原始尺寸的宽度  height:  max-height  min-height  取值以px为单位的数字 |
| 页面中允许设置尺寸的元素  块级元素可以设置宽高  行内元素宽高无效  行内块input默认自带宽高，浏览器不同，默认宽高不同  可以设置宽高  自带宽高属性的元素，可以设置宽高(img,table) |
| 块级元素默认宽高  块级元素不设置宽度，默认是父元素宽度100%  块级元素不设置高度，默认高度是内容撑开，没有内容就没有高度  行内元素宽高，是靠内容撑开，没有内容就没有宽高 |

附加知识点---单位

|  |
| --- |
| 1.px像素  2.in 英寸 1in=2.54cm  3.pt 磅值 1pt=1/72 in,多用于字号大小  4.cm  5.mm  6.% 相对单位  7.em 相对于父元素的值，乘以的倍数，1em=父元素的取值  8.rem 相对于html中设置值，乘以的倍数。1rem=html的取值 |