**CSS层叠样式表（ Cascading style sheets）**

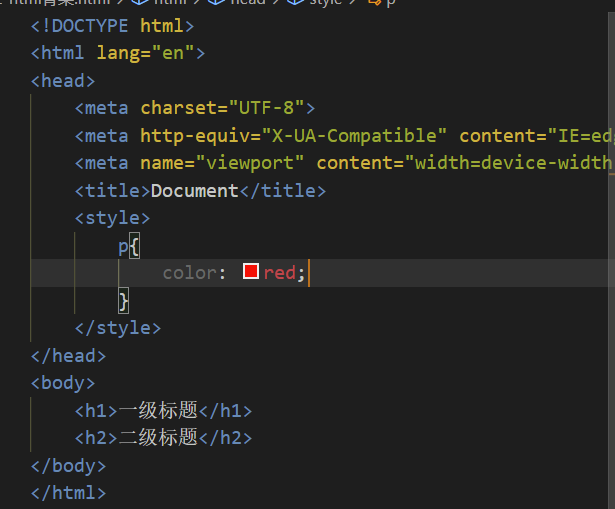
CSS作用：

给Html标签设置样式

# CSS引入方式：

1、内嵌式：CSS写在style标签中

提示：style标签一般写在head标签里面，title标签下面）



2、外联式：CSS写在一个单独的.css文件中

提示：需要通过link标签在网页中引入

3、行内式：CSS写在标签的style属性中

提示：之后配合js使用

三种引入方式使用建议：1、内嵌式，作用范围当前页面，使用场景小案例

2、外联式，作用范围多个页面，使用场景项目中

3、行内式，作用范围当前标签，使用场景配合js使用

# 基础选择器：

标签选择器

结构：标签名{css属性名：属性值；}

作用：通过标签名，找到页面中所有这类标签，设置样式

注意：1、标签选择器选择的是一类标签，而不是单独一个

2、标签选择器无论嵌套关系多深，都能找到对应标签

类选择器

结构：.类名{css属性名：属性值；}

作用：通过类名，找到页面中所有带有这个类名的标签，设置样式

注意：1、所有标签上都有class属性，class属性的属性值称为类名

2、类名可以由数字、字母、下划线、中划线组成，但不能以数字或中划线开头

3、一个标签可以同时有多个类名，类名之间以空格隔开

4、类名可以重复，一个类选择器可以同时选中多个标签

id选择器

结构：#id属性值{css属性名：属性值；}

作用：通过id属性值，找到页面中带有这个id属性值的标签，设置样式

注意：1、所有标签都有id属性

2、id属性值类似于身份号码，在一个页面中是唯一的，不可重复的

3、一个标签上只能有一个id属性值

4、一个id选择器只能选中一个标签

通配符选择器

结构：\*{css属性名：属性值；}

作用：找到页面中所有标签，设置格式

注意：1、开发中极少使用

2、在小页面中可能用于去除标签默认的margin和padding

# 字体和文本样式：

字体大小

属性名：font-size

取值：数字+px

注意：1、谷歌浏览器默认文字大小16px

2、单位需要设置，否则无效

字体粗细

属性名：font-weight

取值：1、关键字，正常normal，加粗bold

2、整百纯数字100~900，正常400，加粗700

注意：1、不是所有字体都提供了9种粗细

2、400和700两种取值用得最多

字体倾斜

属性名：font-style

取值：正常normal，倾斜italic

常见字体系列

无衬线字体（sans-serif）

1、特点：文字笔画粗细均匀，并且首尾无装饰

2、场景：网页中大多采用无衬线字体

3、常见该系列字体：黑体、Arial

衬线字体（serif）

1、特点：文字笔画粗细不均，并且首尾有笔锋装饰

2、场景：报刊书籍中应用广泛

3、常见该系列字体：宋体、Times New Roman

等宽字体（monospace）

1、特点：每个文字或字母的宽度相等

2、场景：一般用于程序代码编写，有利于代码的阅读和编写

3、常见该系列字体：Consolas、fira code

字体系列

属性名：font-family

常见取值：具体字体1，具体字体2，、、、，具体字体n，字体系列

具体字体：”Microsoft YaHei”、微软雅黑、黑体、宋体、楷体等

字体系列：sans-serif、serif、monospace等

渲染规则：1、从左往右按照顺序查找，如果电脑中未安装该字体，则显示下一个字体

2、如果都不支持，则会根据系统显示最后字体系列的默认字体

注意：1、如果字体名称中存在多个单词，推荐使用引号包裹

2、最后一项字体系列不需要引号包裹

3、网页开发时，尽量使用系统常见自带字体，保证不同用户都可正常显示

样式的层叠问题

如果给同一个标签设置了相同的样式，此时样式会层叠，写在最下面的会生效

字体font相关属性的连写(复合属性)

属性名：font

取值：font:style值 weight值 size值 family值;

省略要求：只能省略前两个，如果省略了相当于设置了默认值

注意：如果需要同时设置单独和连写形式

1、要么把单独的样式写在连写下面

2、要么把单独的样式写在连写里面

文本缩进

属性名：text-indent

取值：1、数字+px

2、数字+em（推荐：1em=当前标签的font-size大小）

文本水平对齐方式

属性名：text-align

取值：1、left左对齐

2、center居中对齐

3、right右对齐

text-align:center能让哪些元素水平居中：

1、文本

2、span标签、a标签

3、input标签、img标签

注意：如果需要让以上元素水平居中，text-align属性需要给以上元素所在的父元素设置

文本修饰

属性名：text-decoration

取值：1、underline下划线

2、line-through删除线

3、overline上划线

4、none无装饰线

注意：开发中会使用text-decoration:none;清除a标签默认的下划线

行高

作用：控制一行的上下间距

属性名：line-height

取值：1、数字+px

2、倍数（当前标签font-size的倍数）

应用：1、让单行文本垂直居中可以设置line-height:文字父元素高度

2、网页精准布局时，会设置line-height:1可以取消上下间距

行高与font连写的注意点：1、如果同时设置了行高和font连写，注意覆盖问题

2、font:style值 weight值 size/line-height值 family值;

行高=上间距+文字高度+下间距

颜色常见取值

属性名：1、color文字颜色

2、background-color背景颜色

属性值：1、关键词，预定义的颜色名字，red、green等

2、rgb表示法，红绿蓝三原色（每项取值0~255），rgb(0,0,0)、rgb(255,255,255)等

3、rgba表示法，红绿蓝+a表示透明度（取值0~1），rgba(255,0,0,0.5)等

4、十六进制表示法，#开头+六位数（将数字转换成16进制表示），#000000、#ff0000

标签居中

如果需要让div、p、h这类大盒子水平居中：可以通过margin：0 auto；实现

注意：1、对于上述这类大盒子居中，直接给当前元素本身设置即可

2、margin：0 auto；一般针对于固定宽度的盒子，如果大盒子没有设置宽度，则会默认占满父元素的宽度

# 谷歌调试工具：

F12调出

支持在网页中调试文本等的大小、启用或停用代码，显示层叠、错误等

查错步骤：1、点击检查

2、在元素列表中找到对应元素

3、看样式列表中有没有自己写的选择器，如果没有，一般选择器写错了（拼写或语法）

4、如果有选择器但没有样式，看是否有删除线（有的话，样式被注释或者覆盖），看是否有小三角形（有的话，可能属性值后面没有分号，出现中文标点，属性名或属性值单词拼写错误）

# 选择器进阶：

复合选择器

后代选择器：空格

作用：根据Html标签的嵌套关系，选择父元素后代中满足条件的元素

选择器语法：选择器1 选择器2{css}

结果：在选择器1所找到标签的后代中，找到满足选择器2的标签，设置样式

注意：1、后代包括：儿子、孙子、重孙子等

2、后代选择器中，选择器与选择器之间通过空格隔开

子代选择器：>

作用：根据Html标签的嵌套关系，选择父元素子代中满足条件的元素

选择器语法：选择器1 > 选择器2{css}

结果：在选择器1所找到标签的子代中，找到满足选择器2的标签，设置样式

注意：1、子代只包括儿子

2、子代选择器中，选择器与选择器之间通过>隔开

并集选择器

作用：同时选择多组标签，设置相同的样式

选择器语法：选择器1，选择器2{css}

注意：1、并集选择器中的每组选择器之间通过逗号隔开

2、并集选择器中的每组选择器可以是基础选择器或者复合选择器

3、并集选择器中的每组选择器通常一行写一个，提高代码可读性

交集选择器

作用：选中页面中同时满足多个选择器的标签

语法：选择器1选择器2{css}

注意：1、交集选择器中的选择器之间使紧挨着的，没有东西分割

2、如果有标签选择器，标签选择器必须写在最前面

hover伪类选择器

作用：选中鼠标悬停在元素上的状态，设置样式

语法：选择器:hover{css}

注意点：1、伪类选择器选中的元素的某种状态

2、任何一个标签都可以添加伪类，任何一个标签都可以鼠标悬停

Emmet语法

作用：通过简写语法，快速生成代码

语法：

|  |  |  |
| --- | --- | --- |
| 记忆 | 示例 | 效果 |
| 标签名 | div | <div></div> |
| 类选择器 | .red | <div class=”red”></div> |
| id选择器 | #one | <div id=”one”></div> |
| 交集选择器 | p.red#one | <p class=”red” id=”one”></p> |
| 子代选择器 | ul>li | <ul><li></li></ul> |
| 内部文本 | ul>li(我是li的内容) | <ul><li>我是li的内容</li></ul> |
| 创建多个 | ul>li\*3 | <ul><li></li><li></li><li></li></ul> |

# 背景相关属性：

背景颜色

属性名：background-color(bgc)

属性值：关键字、rgb表示法、rgba表示法、十六进制等

注意：1、背景颜色默认值是透明：rgba(0,0,0,0)、transparent

2、背景颜色不会影响盒子大小，并且还能看清盒子的大小和位置，一般布局中会习惯先给盒子设置背景颜色

背景图片

属性名：background-image(bgi)

属性值：background-image:url(‘图片的路径’);

注意点：1、背景图片中url中可以省略引号

2、背景图片默认实在水平和垂直方向平铺的

3、背景图片仅仅是指给盒子起到装饰效果，类似于背景颜色，不能撑开盒子

背景平铺

属性名：background-repeat(bgr)

作用：图片只显示一次

属性值：

|  |  |
| --- | --- |
| 取值 | 效果 |
| repeat | （默认值）水平和垂直方向都平铺 |
| no-repeat | 不平铺 |
| repeat-x | 沿水平方向平铺 |
| repeat-y | 沿垂直方向平铺 |

背景位置

属性名：background-position(bgp)

属性值：background-position：水平方向位置 垂直方向位置；

1、方位名词（最多只能表示9个位置）：

水平方向：left、center、right

垂直方向：top、center、bottom

2、数字+xp（坐标）：

原点（0，0）：盒子左上角

x轴：水平向右

y轴：垂直向下

注意：1、方位名词取值和坐标取值可以混合使用

2、第一个取值表示水平，第二个取值表示垂直

背景相关属性连写(复合属性)

属性名：background(bg)

属性值：单个属性值的合写，取值之间以空格隔开

书写顺序（无顺序要求，推荐）：background:color image repeat position

省略问题：1、按需求省略

2、特殊情况：在p端，如果盒子大小和背景图片大小一样，此时可以直接写background：url()

注意：如果需要设置单独的样式和连写：1、要么把单独的样式写在连写下面

2、要么把单独的样式写在连写的里面

img标签和背景图片的区别

在网页中显示一张图片效果：

方法一：直接写上img标签，img标签是一个标签，不设置宽高会默认以原尺寸显示

方法二：div标签+背景图片，需要设置div的宽高，因为背景图片知识装饰的CSS样式，不能撑开div标签

img用来实现网页中重要的图片，主要以展现图片为主

background主要起装饰作用，装饰其他功能

# 元素显示模式：

块级元素

显示特点：1、独占一行（一行只能显示一个）

2、宽度默认是父元素的宽度，高度默认由内容撑开

3、可以设置宽高

代表标签：div、p、h系列、ul、li、dl、dt、dd、form、header、nav、footer

行内元素

显示特点：1、一行可以显示多个

2、宽度和高度默认由内容撑开

3、不可以设置宽高

代表标签：a、span、b、u、i、s、strong、ins、em、del

行内块元素

显示特点：1、一行可以显示多个

2、可以设置宽高

3、宽高由内容撑开，即不设置也无内容则默认为0

代表标签：input、textarea、button、select

特殊情况：img标签有行内块元素特点，但是谷歌调试工具中显示结果是inline

元素显示模式转换

目的：改变元素默认的显示特点，让元素符合布局要求

语法：

|  |  |  |
| --- | --- | --- |
| 属性 | 效果 | 使用频率 |
| display:block | 转换成块级元素 | 较多 |
| display:inline-block | 转换成行内块元素 | 较多 |
| display:inline | 转换成行内元素 | 极少 |

Html嵌套规范注意点

1、块级元素一般作为大容器，可以嵌套：文本、块级元素、行内元素、行内块元素等

但：p标签中不要嵌套div、p、h等块级元素

2、a标签内部可以嵌套任意元素

但：a标签不能嵌套a标签

# CSS三大特性：

继承性

特性：子元素有默认继承父元素样式的特点（子承父业）

可以继承的常见属性（文字控制属性都可以继承）

1、color

2、font-style、font-weight、font-size、font-family

3、text-indent、text-align

4、line-height

注意：可以通过调试工具判断样式是否可以继承

继承失效的特殊情况：

1、a标签的color会继承失效

2、h系列标签的font-size会继承失效

层叠性

特性：

1、给同一个标签设置不同的样式，样式会层叠叠加，共同作用在标签上

2、给同一个标签设置相同的样式，样式会层叠覆盖，写在最后的样式才会生效

注意：当样式冲突时，只有当选择器优先级相同时，才能通过层叠性判断结果

优先级

特性：不同选择器具有不同的优先级，优先级高的选择器样式会覆盖优先级低的选择器样式

优先级公式：

继承< 通配符选择器 < 标签选择器 < 类选择器 < id选择器 < 行内选择器 < !improtant

注意：1、!important写在属性值的后面，分号的前面

2、!important不能提升继承的优先级，只要是继承，优先级最低（自己有样式，不满足继承条件）

3、实际开发中不建议使用!important

权重叠加计算

场景：如果是复合选择器，此时需要通过权重叠加计算方法，判断最终哪个选择器优先级最高会生效

权重叠加计算公式：（每一级之间不存在进位）

（0，0，0，0）

第一级

第二级

第三级

第四级

**行内样式的个数**

**id选择器的个数**

**类选择器的个数**

**标签选择器的个数**

**复合选择器中**：

比较规则：

1、先比较第一级数字，如果比较出来了，之后的统统不看

2、如果第一级数字相同，再去比较第二级数字，如果比较出来了，之后的不看

3、同上

4、如果最终所有数字相同，表示优先级相同，则比较层叠性（谁写在下面谁说了算）

注意：!important如果不是继承，则权重最高

# PxCook的基本使用：

设计模式：处理png、jpg等照片格式

开发模式：处理PSD等格式

# 盒子模型：

盒子模型的介绍

盒子的概念：

1、页面中的每一个标签，都可以看作是一个“盒子”，通过盒子的视角更方便的进行布局

2、浏览器在渲染（显示）网页时，会将网页中的元素看作是一个个的矩形区域，我们也形象地称之为盒子

盒子模型：CSS中规定每个盒子分别由内容区域（content）、内边距区域（padding）、边框区域（border）、外边距区域（margin）构成，这就是盒子模型

内容区域的宽度和高度

作用：利用width和height属性默认设置是盒子内容区域的大小

属性：width/height

常见取值：数字+px

边框(border)--复合属性

属性名：border

属性值：单个取值的连写，取值之间以空格隔开，不分先后顺序

如：border:10px solid red;

快捷键：bd+tab

注意：border会撑大盒子尺寸

边框(border)--单方向设置

场景：只给盒子的某个方向单独设置边框

属性名：border-方位名词

属性值：连写的属性值

边框(border)--单个属性

作用：给设置边框粗细、边框样式、边框颜色效果

单个属性：

|  |  |  |
| --- | --- | --- |
| **作用** | **属性名** | **属性值** |
| 边框粗细 | border-width | 数字+px |
| 边框样式 | border-style | 实现solid、虚线dashed、点线dotted |
| 边框颜色 | border-color | 颜色取值 |

内边距(padding)

属性名：padding

用法：1、padding:数字+px ;（设置四个方向）

2、padding:数字+px 数字+px 数字+px 数字+px ;

padding属性可以当作复合属性使用，表示单独设置某个方向的内边距

padding最多取4值（上、右、下、左），未取值的方向内边距大小和其对面的大小相同

注意：padding会撑大盒子尺寸

CSS盒模型(自动内减)

场景：给盒子设置border或padding时，盒子会被撑大，不想盒子被撑大

解决方法：

1、手动内减

操作：自己计算多余大小，手动在内容中减去

缺点：计算量大，麻烦

2、自动内减

操作：给盒子设置属性box-sizing : border-box ; 即可

优点：浏览器会自动计算多余大小，自动在内容中减去

外边距(margin）

属性名：margin

用法：1、margin:数字+px ;（设置四个方向）

2、margin:数字+px 数字+px 数字+px 数字+px ;

margin属性可以当作复合属性使用，表示单独设置某个方向的内边距

margin最多取4值（上、右、下、左），未取值的方向内边距大小和其对面的大小相同

注意：margin会撑大盒子尺寸

清除默认内外边距

场景：浏览器会默认给部分标签设置默认的margin和padding，但一般在项目开始前需要先清除这些标签默认的margin和padding，后续自己设置

比如：1、body标签默认有margin:8px

2、p标签默认有上下的margin

3、ul标签默认有上下的margin和padding-left

解决：

margin:0

padding:0

版心居中

版心：网页的有效内容

用法：margin:数字+px auto ;

外边距折叠现象

1、合并现象

场景：垂直布局的块级元素，上下的margin会合并

结果：最终两者的距离为margin的最大值

解决方法：只给其中一个盒子设置margin

2、塌陷现象

场景：互相嵌套的块级元素，子元素的margin-top会作用在父元素上

结果：导致父元素一起往下移动

解决方法：1、给父元素设置border-top或者padding-top（分隔父子元素的margin-top）

2、给父元素设置overflow:hidden

3、转换成行内块元素

4、设置浮动

行内元素的垂直内外边距

想要通过margin或padding改变行内标签的位置，无法生效

# CSS浮动：

结构伪类选择器

作用与优势：

1、作用：根据元素在Html中的结构关系查找元素

2、优势：减少对于Html中类的依赖，有利于保持代码整洁

3、场景：常用于查找某父级选择器中的子元素

选择器：

|  |  |
| --- | --- |
| **选择器** | **说明** |
| E:first-child { } | 匹配父元素后中第一个子元素，并且是E元素 |
| E:last-child { } | 匹配父元素中最后一个子元素，并且是E元素 |
| E:nth-child(n) { } | 匹配父元素中第n个子元素，并且是E元素 |
| E:nth-last-child(n) { } | 匹配父元素中倒数第n个子元素，并且是E元素 |

n的注意点：

1、n为：0、1、2、3、4等

2、通过n可以组成常见公式（公式中n从0开始取值）

|  |  |
| --- | --- |
| **功能** | **公式** |
| 偶数 | 2n、even |
| 奇数 | 2n+1、2n-1、odd |
| 找到前5个数 | -n+5 |
| 找到从第5个往后 | n+5 |

伪元素

伪元素：一般页面中的非主体内容可以使用伪元素

区别：

1、元素：Html设置的标签

2、伪元素：由CSS模拟出的标签效果

种类：

|  |  |
| --- | --- |
| **伪元素** | **作用** |
| ::before | 在父元素内容的最前添加一个伪元素 |
| ::after | 在父元素内容的最后添加一个伪元素 |

注意：

1、必须设置content属性才能生效

2、伪元素默认是行内元素

标准流

标准流：又称文档流，是浏览器在渲染显示网页内容时默认采用的一套排版规则，规定了应该以何种方式排列元素

常见标准流排版规则：

1、块级元素：从上往下，垂直布局，独占一行

2、行内元素或行内块元素：从左往右，水平布局，空间不够自动折行

浮动的作用

早期的作用：图文环绕

现在的作用：网页布局（让div标签完美的在一行排列）

场景：浏览器在解析行内块和行内标签时，如果标签换行书写，会产生一个空格的间隙

浮动的代码

float:left

float:right

浮动的特点

1、浮动元素会脱离标准流（简称脱标），在标准流中不占位置

2、浮动元素比标准流高半个级别，可以覆盖标准流中的元素

3、浮动找浮动，下一个浮动元素会在上一个浮动元素后面左右浮动

4、浮动的标签顶对齐

5、浮动元素有特殊的显示效果，一行可以显示多个，可以设置宽高

6、浮动后的标签具备行内块特点

7、父级的宽度不够，子级会自动换行

注意：浮动之后的元素不能通过text-align:center或者margin:0 auto

CSS属性书写顺序

1、浮动/display

2、盒子模型：margin、border、padding、宽高、背景色

3、文字样式

优势：浏览器执行效率高，打开速度快

清除浮动的介绍

含义：清除浮动带来的影响（如果子元素浮动了，此时子元素不能撑开标准流的块级父元素）

目的：需要父元素有高度，从而不影响其他网页元素的布局

场景：父子级标签，子级浮动，父级没有高度，后面的标签标准流盒子会受影响

清除浮动的方法

1、直接设置父元素高度

优点：简单方便

缺点：有些布局中不能固定父元素高度，如新闻列表、京东推荐模块

2、额外标签法

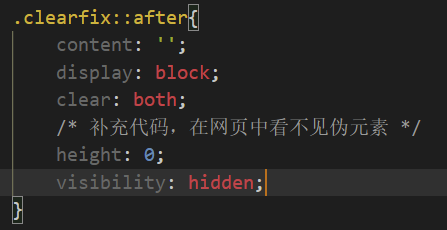
操作：1、在父元素内容的最后添加一个块级标签

2、给添加的块级元素设置clear:both

缺点：会在页面中添加额外的标签，让页面的Html结构变得复杂

3、单伪元素清除法

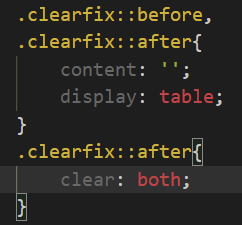
操作：用伪元素代替了额外标签



优点：项目中使用，直接给标签加类即可清除浮动

4、双伪元素清除法

操作：



优点：项目中使用，直接给标签加类即可清除浮动，可同时处理外边距折叠和清除浮动

注意：.clearfix::before作用解决外边距塌陷问题

5、给父元素设置overflow:hidden

操作：直接给父元素设置overflow:hidden

# CSS定位：

定位的基本介绍

网页常见布局方式：

1、标准流

块级元素独占一行 → 垂直布局

行内元素/行内块元素一行显示多个 → 水平布局

2、浮动

可以让原本垂直布局的块级元素变成水平布局

3、定位

可以让元素自由的摆放在网页的任意位置

一般用于盒子之间的层叠情况

定位常见应用场景：

1、可以解决盒子与盒子之间的层叠问题

定位之后的元素层级最高，可以层叠在其他盒子上面

2、可以让盒子始终固定在屏幕中的某个位置

定位的基本使用

使用定位的步骤：

1、设置定位方式

属性名：position

常见属性值：

|  |  |
| --- | --- |
| **定位方式** | **属性值** |
| 静态定位 | static |
| 相对定位 | relative |
| 绝对定位 | absolute |
| 固定定位 | fixed |

2、设置偏移值

偏移值设置分为两个方向，水平和垂直方向各选一个使用

选取的原则一般是就近原则

|  |  |  |  |
| --- | --- | --- | --- |
| **方向** | **属性名** | **属性值** | **含义** |
| 水平 | left | 数字+px | 距离左边的距离 |
| 水平 | right | 数字+px | 距离右边的距离 |
| 垂直 | top | 数字+px | 距离上边的距离 |
| 垂直 | bottom | 数字+px | 距离下边的距离 |

相对定位

介绍：相对于自己之前的位置进行移动

代码：position:relative;

特点：

1、需要配合方位属性实现移动

2、相对于自己原来的位置进行移动

3、在页面中占位置 → 没有脱标

应用场景：

1、配合绝对定位组cp

2、用于小范围的移动

注意：如果left和right都有，以left为准，top和bottom都有，以top为准

绝对定位

介绍：相对于非静态定位的父元素进行定位移动

代码：position:absolute;

特点：

1、需要配合方位属性移动

2、默认相对于浏览器可视区域(body)进行移动

3、在页面中不占位置 → 已经脱标

4、改变标签的显示模式特点：具备行内块特点(!!注意只是具备特点，但不是行内块!!)

5、绝对定位查找父级的方式：就近查找定位的父级，如果逐层找不到这样的父级，就以浏览器参照进行定位

应用场景：配合相对定位组cp(子绝父相)

注意：绝对定位后的盒子不能使用margin:0 auto;实现居中，因为已经脱标了

固定定位

介绍：相对于浏览器进行定位移动

代码：position:fixed;

特点：

1、需要配合方位属性进行移动

2、相对于浏览器可视区域进行移动

3、在页面中不占位置 → 已经脱标

应用场景：让盒子固定在屏幕中某个位置

元素的层级关系

不同布局方式元素的层级关系：标准流 < 浮动 < 定位

不同定位之间的层级关系：

1、相对、绝对、固定默认层级相同

2、此时HTML中写在下面的元素层级更高，会覆盖上面的元素（后来者居上）

设置定位之间的层级关系：

属性名：z-index

属性值：数字

用法：配合定位，默认值为0，属性值越大，越靠上

# CSS装饰：

认识基线

基线：浏览器文字类型元素排版中存在用于对齐的基线（baseline）

文字对齐问题

场景：解决行内/行内块元素垂直对齐问题

问题：浏览器遇到行内和行内块标签当作文字处理，默认文字是按基线对齐，所以当图片和文字在一行中显示时，其实底部不是对齐的

垂直对齐方式

属性名：vertical-align

属性值：

|  |  |
| --- | --- |
| **属性值** | **效果** |
| baseline | 默认，基线对齐 |
| top | 顶部对齐 |
| middle | 中部对齐 |
| bottom | 底部对齐 |

光标类型

场景：设置鼠标光标在元素上时显示的样式

属性名：cursor

常见属性值：

|  |  |
| --- | --- |
| **属性值** | **效果** |
| default | 默认值，通常是箭头 |
| pointer | 小手效果，提示用户可以点击 |
| text | 工字形，提示用户可以选择文字 |
| move | 十字光标，提示用户可以移动 |

边框圆角

场景：让盒子四个角变得圆润，增加页面细节，提升用户体验

属性名：border-radius

常见取值：数字+px、百分比

原理：

半径

赋值规则：从左上角开始赋值，然后顺时针赋值，没有赋值的看对角

边框圆角常见应用：

1、正圆：盒子必须是正方形，设置border-radius:50%

2、胶囊按钮：盒子必须是长方形，设置border-radius:盒子高度的一半

溢出显示效果

溢出部分：指的是盒子内容部分所超出盒子范围的区域

场景：控制内容溢出部分的显示效果，如显示、隐藏、滚动条等

属性名：overflow

常见属性值：

|  |  |
| --- | --- |
| **属性值** | **效果** |
| visible | 默认值，溢出部分可见 |
| hidden | 溢出部分隐藏 |
| scroll | 无论是否溢出，都显示滚动条 |
| auto | 根据是否溢出，自动显示或隐藏滚动条 |

元素本身隐藏

场景：让某元素本身在屏幕中不可见。如：鼠标悬停的标签:hover之后元素隐藏

常见属性：

1、visibility:hidden

2、display:none

区别：前者占位隐藏，后者不占位隐藏

元素整体透明度

场景：让某元素整体（包括内容）一起变透明

属性名：opacity

属性值：0~1之间的数字（1表示完全不透明，0表示完全透明）

注意点：opacity会让元素整体变透明，包括里面的内容，如文字、子元素等

边框合并

场景：让相邻表格边框进行合并，得到细线边框效果

代码：border-collapse:collapse;

用CSS画三角形

场景：在网页中展示出小三角形时，除了可以使用图片外，还可以使用代码完成

实现原理：利用盒子边框完成

实现步骤：

1、设置一个盒子

2、设置四周不同颜色的边框

3、将盒子宽高设置为0，仅保留边框

4、得到四个三角形，选择其中一个后，其他三角形（边框）设置颜色为透明

# 选择器拓展：

链接伪类选择器

场景：常用于选中超链接的不同状态

选择器语法：

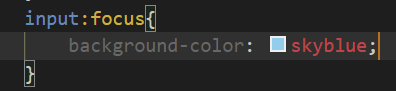
|  |  |
| --- | --- |
| **选择器语法** | **功能** |
| a:link { } | 选中a链接未访问过的状态 |
| a:visited { } | 选中a链接访问之后的状态 |
| a:hover { } | 选中鼠标悬停的状态 |
| a:active { } | 选中鼠标按下的状态 |

注意点：如果需要同时实现以上四种伪类状态效果，需要按照link、visited、hover、active的顺序书写

焦点伪类选择器

场景：用于选中元素获取焦点时的状态，常用于表单控件

选择器语法：



效果：表单控件获取焦点时默认会显示外部轮廓线

属性选择器

场景：通过元素上的HTML属性来选择元素，常用于选择input标签

选择器语法：

|  |  |
| --- | --- |
| **选择器** | **功能** |
| E[attr] | 选择具有attr属性的E元素 |
| E[attr=”val”] | 选择具有attr属性并且属性值等于val的E元素 |

# CSS样式补充：

精灵图

场景：项目中将多张小图片，合并成一张大图片，这张大图片称之为精灵图

优点：减少服务器发送次数，减轻服务器压力，提高页面加载速度

精灵图的使用步骤：

1、创建一个盒子，设置盒子的尺寸和小图尺寸相同

2、将精灵图设置为盒子的背景图片

3、修改背景图位置：通过PxCook测量小图片左上角坐标，分别取负值设置给盒子的background-position:x y

注意：精灵图的标签都用行内标签

背景图片大小

作用：设置背景图片大小

语法：background-size:宽度 高度

取值：

|  |  |
| --- | --- |
| **取值** | **场景** |
| 数字+px | 简单方便，常用 |
| 百分比 | 相对于当前盒子自身的宽高百分比 |
| contain | 包含，将背景图片等比例缩放，直到不会超出盒子的最大 |
| cover | 覆盖，将背景图片等比例缩放，直到刚好填满整个盒子没有空白 |

background连写拓展

完整连写：background: color image repeat position/size;

注意点：background-size和background连写同时设置时，需要注意覆盖问题

解决：1、要么单独的样式写连写的下面

2、要么单独的样式写连写的里面

文字阴影

作用：给文字添加阴影效果，吸引用户注意

属性名：text-shadow

取值：

|  |  |
| --- | --- |
| **参数** | **作用** |
| h-shadow | （必须）水平偏移量，允许负值 |
| v-shadow | （必须）垂直偏移量，允许负值 |
| blur | （可选）模糊度 |
| color | （可选）阴影颜色 |

注意：阴影可以叠加设置，每组阴影取值之间以逗号隔开

盒子阴影

作用：给盒子添加阴影效果，吸引用户注意，体现页面的制作细节

属性名：box-shadow

取值：

|  |  |
| --- | --- |
| **参数** | **作用** |
| h-shadow | （必须）水平偏移量，允许负值 |
| v-shadow | （必须）垂直偏移量，允许负值 |
| blur | （可选）模糊度 |
| spread | （可选）阴影扩大 |
| color | （可选）阴影颜色 |
| inset | （可选）将阴影改为内部阴影 |

过渡

作用：让元素的样式慢慢的变化，常配合hover使用，增强网页交互体验

属性名：transition

常见取值：

|  |  |
| --- | --- |
| **参数** | **取值** |
| 过渡的属性 | all:所有能过渡的属性都过渡  具体属性名如：width:只有width有过渡 |
| 过渡的时长 | 数字+s（秒） |

注意：

1、过渡需要：默认状态和hover状态样式不同，才能有过渡效果

2、transition属性给需要过渡的元素本身加

3、transition属性设置在不同状态中，效果不同：

给默认状态设置，鼠标移入移出都有过渡效果

给hover状态设置，鼠标移入有过渡效果，移出没有过渡效果

# 项目前置认知：

网页与网站的关系

网页：网站中的每一“页”

网站：提供特定服务的一组网页集合

骨架结构标签

<!DOCTYPE html>文档类型声明，告诉浏览器该网页的HTML版本

<html lang=”en”>标识网页使用的语言

作用：搜索引擎归类+浏览器翻译

常见语言：zh-CN简体中文/en英文

<meta charset=”UTF-8”>标识网页使用的字符编码

作用：保存和打开字符编码需要统一设置，否则可能会出现乱码

常见字符：

1、UTF-8：万国码，国际化的字符编码，收录了全球语言的文字

2、GB2312：6000+汉字

3、GBK：20000+汉字

注意：开发中统一使用UTF-8字符编码

<meta http-equiv=”X-UA-Compatible” content=”IE=edge”>解决IE浏览器兼容问题

<meta name=”Viewport” content=”width=device-width, initial-scale=1.0”>移动端网页

SEO三大标签

Search Engine Optimization:搜索引擎优化

作用：让网站在搜索引擎上的排名靠前

提升SEO的常见方法：

1、竞价排名

2、将网页制作成html后缀

3、标签语义化（在合适的地方使用合适的标签）

三大标签:

1、title：网页标题标签

2、description：网页描述标签

3、keywords：网页关键词标签

favicon标题图标

常见代码：<link rel=”shortcut icon” href=”ico图标路径” type=”image/x-icon”>

# 项目结构搭建：

文件和目录准备

1、新建项目文件夹：name-pc-client，在VScode中打开

实际开发中，项目文件夹不建议使用中文

所有项目相关文件都保存在name-pc-client目录中

2、复制favicon.ico到name-pc-client目录中

3、复制images和uploads目录到name-pc-client目录中

images:存放网站固定使用的图片素材，如：logo、样式修饰图片等

uploads:存放网站非固定使用的图片素材，如：商品图片、宣传图片等

4、新建index.html在根目录

5、新建css文件夹保存网站的样式，并新建以下css文件：

base.css:基础公共样式

common.css:该网站中多个页面相同模块的重复样式，如：头部、底部等

index.css:首页样式

基础公共样式

场景：一般项目开始前，首先会去除掉浏览器默认样式，设置为当前项目需要的初始化样式

作用：防止不同浏览器中标签默认样式不同的影响，同意不同浏览器的默认显示效果，方便后续项目开发

要求：已经准备好base.css代码，项目中可以直接引入使用

index页面骨架

