CSS常见引用方式：

内嵌式：CSS写在style标签中，作用在当前页面，适用于小案例

外联式：CSS写在单独的CSS文件中，通过link标签引入，适用于项目中

Eg:<link rel=”stylesheet” href=”xxx.css”>

行内式：CSS写在标签的style属性中，作用于当前标签，配合js使用

**选择器**

**标签选择器**，以标签命名：选中所有这种类型标签

**类选择器**：结构：.类名{css属性名：属性值}

注意：不能以数字或者中划线开头；

一个标签可以有多个类名，类名之间以空格隔开；

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}

注意：所有标签上都有id属性；

id属性值类似于身份证号码，在一个页面中是唯一的，不可重复的；

一个标签上只能有一个id属性值；

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：找到页面中所有的标签，设置样式

开发中只会在极特殊情况下才会用到

**字体样式：**

字体大小：font-size（网页当中默认字号是16px）

字体样式：font-weight（正常400 加粗700）

字体粗细：font-style（控制倾斜 italic 正常：normal）

字体类型：font-family（字体：Windows微软雅黑；macOS：苹方）

字体类型：font属性连写

**常见字体系列**：

网页中大多采用无衬线字体（sans-serif）

报刊书籍大多应用衬线字体（serif）

程序代买编写一般用等宽字体（monospace）

如果给同一个标签设置了相同的属性，此时样式会层叠（覆盖），写在最下面的会生效

**字体font相关属性的连写**

属性名：font（复合属性）

取值：font : style weight size family;（中间空格隔开，要一一对应）

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

**文本缩进：text-indent** 一般取值用em（em：一个字的大小）

Eg:

<style>

.xx

{

text-indent : 2em ; //首行缩进两字符（1em=当前标签的font-size大小）

}

</style>

**文本水平居中方法总结：text-align：center**

text-align：center可以让：

1.文本 2.span标签、a标签 3.input标签、img标签居中

注意：要让以上元素水平居中，text-align：center

**文本修饰：text-decoration**

underline：下划线

line-through：删除线

overline：上划线

none：无装饰线

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

**行高（line-height）**

作用：控制一行的上下间距

取值：数字+px

应用：**让单行文本垂直居中可以设置line-height：文字父元素高度**

网页精准布局时，会设置**line-height：1**取消上下间距

行高与font连写要注意覆盖问题：

font：style weight size/line-height family；（字号和行高用“/”隔开）

**颜色表示：**

rgb：红绿蓝三原色，每项取值范围：0~255；rgba（255,255,255）

rgba：红绿蓝三原色+a表示透明度，a取值范围是0~1 ；rgba（255,255,255,0.5）

十六进制表示法：#开头，将数字转换成十六进制表示：#000000（简写#000）；#e92322

**透明度：opacity**：value（0~1）；

**标签水平居中方法总结 margin：0 auto**（0：上下间距；auto 左右间距）

**padding**：内边距（内容区和边框的距离）

**margin**：外边距（边框外边的距离）

margin：10px auto 20px（上 左 下边距）

margin-top：上边距

margin：10px 20px 30px 40px（上 右 下 左边距（顺时针））

**Border-radius**：设置圆角

**选择器**：

**后代选择器**：空格（空格隔开，选中所有后代）

语法：选择器1 选择器2{css}

**子代选择器**：> (大于号隔开，只选中儿子)

语法：选择器1>选择器2{css}

**并集选择器**： ，（逗号隔开，同时选择多组标签，设置相同样式）

语法：选择器1，选择器2{css}

**交集选择器**：紧挨着（选择器直接紧挨着，选中页面中同时满足多个选择器的标签）

语法：语法：选择器1选择器2{css}

***hover伪类选择器***：（选中鼠标悬停在元素上的状态，设置样式）

语法： 选择器：hover{css}

**背景图片：background-image** (bgi)

属性值：background-image：url（图片的路径）

**背景平铺：background-repeat**（bgr）

属性值：

repeat：默认值（水平和垂直方向都平铺）

no-repeat：不平铺

repeat-x：水平平铺

repeat-y：垂直平铺

**背景位置：background-position**（bgp）

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

**背景相关属性的连写形式**：

属性名：background

书写顺序（可以更改与省略）：color image repeat position；

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

img放重要图片，background-image放修饰图片

**去掉ul li 中的默认点：list-style：none；**

**元素显示模式**

**块级元素：（div，p，h系列**，ul，li，dl……）

特点：1.独占一行

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

**行内元素：（a，span**，b，u……）

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

**行内块元素：（input，textarea**，button，select……）

特点：1.一行可以显示多个

2.可以设置宽高

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

元素显示模式转换：（改变元素默认的显示特点，让元素符合布局要求）

语法：

**display：block**；转换成块级元素

**display：inline-block**；转换成行内块级元素

display：inline；转换成行内元素（极少）

嵌套注意：p标签中不要嵌套div，p，h等块级元素

A标签内部可以嵌套任意元素，a标签不能嵌套a标签

**优先级**：不同的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选择器

公式：

继承<通配符选择器<标签选择器<类选择器<id选择器<行内样式<!important

**注意**：

!important写在属性值的后面，分号的前面

**!important**不能提升继承的优先级，只要是继承优先级最低

实际开发中不建议用 !important

**权重叠加计算**：如果是复合选择器，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

计算公式：（0（第一级）,0（第二级）,0（第三级）,0（第四级））

第一级：行内样式个数

第二级：id选择器个数

第三级：类选择器个数

第四级：标签选择器个数

先比较第一级数字，比较出来了之后通通不看，依次类推

!important如果不是继承，则权重最高

**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

**边框border连写形式**

border：粗细 样式 颜色；（不分先后）

eg：border：10px solid red；

样式：solid（实线） dashed（虚线） dotted（点线）

border单方向设置：

border-right/left/top/bottom

border-right：5px dotted grey；

***border会撑大盒子***

**padding：如果盒子本身没有指定width/height属性，则此时padding不会撑开盒子大小**

**居中**

**块级盒子水平居中条件：margin：0 auto；**

1.盒子必须设定宽度（width）

2.盒子左右边距都设置为auto（上下边距可更改）

**让行内元素或行内块元素（将其视为普通文本）水平居中**，给父级元素添加：

text-align：center；

**嵌套块垂直外边距的塌陷**

对于两个嵌套关系（父子关系）的块元素，父元素有上外边距同时子元素也有上外边距，此时父元素会塌陷较大的外边距值

解决方案：

1. 可以为父元素定义上边框，不想要边框有颜色可以设成：border：1px soild transparent（透明）；
2. 可以为敷衍塞定义上内边距
3. 可以为父元素添加overflow：hidden；

清除内外边距：

\*（通配符选择器）

{

Padding：0；

Margin：0；

}

**圆角边框**

border-radius：length；（数值越大弧线越大）

radius半径（圆的半径）原理：（椭）圆与边框的交集形成圆角效果

画一个圆形只需在正方形盒子中把border-radius设成正方形宽度的一半border-radius：50%；（宽高一半，更常用）

圆角矩形：把border-radius设成高度的一半

分开写：border-top-left-radius（左上角）

border-top-right-radius（右上角）

border-bottom-left-radius（左下角）

border-bottom-right-radius（右下角）

**盒子阴影**

box-shadow：h-shadow v-shadow blur spread color inset；

h-shadow 必需。水平阴影位置，允许负值

v-shadow 必需。垂直阴影位置，允许负值

blur 可选。模糊距离。

spread 可选。阴影的尺寸。

color 可选。阴影的颜色。

Inset 可选。将外部阴影（outset）改为内部阴影。

**注意**：默认的是外阴影（outset），但是不可以写这个单词，否则导致阴影无效

Eg：box-shadow：10px 10px 10px -4px rgba（0,0,0，.3）

text-shadow属性：将阴影应用于文本

语法：text-shadow：h-shadow v-shadow blur color；

**传统网页布局的三种方式：标准流，浮动，定位**

**网页布局第一准则**：多个块级元素纵向排列找标准流，多个块级元素横向排列找浮动

**标准流**：标签按照规定好默认方式排列

1.块级元素会独占一行，从上向下顺序排列

常用元素：div、hr、p、h1~h6、ul、ol、dl……

2.行内元素会按照顺序，从左到右顺序排列，碰到父元素边缘则自动换行

常用元素：span、a、i、em……

**浮动**

特性：**浮动元素会脱离标准流（脱标）**

浮动的元素会一行显示并且元素顶部对齐

浮动的元素会具有行内块元素的特性

**注意点**：先用标准流的父元素排列上下位置，之后内部子元素采取浮动排列左右位置

一个元素浮动了，理论上其余兄弟元素也要浮动

**浮动的盒子只会影响浮动盒子后面的标准流，不会影响前面的标准流。**

**清除浮动**

本质：清除浮动元素脱离标准流造成的影响

策略：闭合浮动，只让浮动在父盒子内部影响，不影响父盒子外面的其他盒子

**方法**：1.额外标签法

2.父级添加overflow属性

3.父级添加after伪类属性

4.父级添加双伪元素

**额外标签法（隔墙法）**：W3C推荐做法

在浮动元素末尾添加一个空的标签。例如：<div style=”clear : both;”>或者其他标签（如<br/>）

优点：通俗易懂，书写方便 缺点：铁架许多无意义的标签，结构化较差

注意：要求这个新的标签必须是块级元素

**父级添加overflow属性**：将其属性值设置为hidden、auto或scroll

优点：代码简洁 缺点：无法显示代码溢出部分

注意：是给父元素添加代码

**父级添加after伪类属性**：

.clearfix:after

{

content:””;

display:block;

height: 0;

clear:both;

visibility:hidden;

}

.clearfix: //IE6\7专有

{

\*zoom:1;

}

优点：没有增加标签，结构更简单 缺点：照顾低版本浏览器

**父级添加双伪元素**

.clearfix : before , .clearfix : after

{

content:””;

display : table;

}

.clearfix:after

{ clear : both; }

.clearfix: //IE6\7专有

{ \*zoom:1; }

优点：代码简洁

缺点：照顾低版本浏览器

**Css书写顺序：**

布局定位属性：display/position/float/clear/visibility/overflow

自身属性：width/height/margin/padding/border/background

文本属性：color/font/text-decoration/text-align

其他属性：content/cursor/border-radius/box-shadow

**页面布局整体思路：**

1. 确定版心（可视区）
2. 分析页面中的行模块，以及每个行模块中的列模块
3. 一行中的列模块经常浮动布局，先确定每个列的大小，之后确定列的位置
4. 制作HTML结构，遵循：先有结构，后有样式的原则
5. **先理清结构布局，再写代码**