## css初识

1.通过标签来修改样式的缺点：

1.1 需要记忆哪些标签有哪些属性，如果该标签没有这个属性，那么设置了也没有效果

1.2 当需求变更时我们需要修改大量的代码才能满足现有的需求

1.3 HTML只有一个作用，就是用来添加语义

所以在企业开发中修改样式都是交给CSS来做

2.通过CSS来修改样式有什么好处？

2.1 不用记忆哪些属性属于哪个标签

2.2 当需求变更时我们不需要修改大量的代码就可以满足需求

2.3 在前端开发中CSS只有一个作用，就是用来修改样式

## css的格式

1.格式：

<style type="text/css">

标签名称{

属性名称:属性对应的值;

...

}

</style>

2.注意点：

2.1 style标签必须写在head标签的开始标签和结束标签之间（也就是必须和title标签是兄弟关系）

2.2 style标签中的type属性其实可以不用写，

默认就是type="text/css"

2.3 设置样式时必须按照固定的格式来设置，key: value;

其中冒号不能省略，分号大多数情况下也不能省略

## 文字属性

1.规定文字样式的属性

格式：font-style: italic;

取值：

normal ：正常的，默认就是正常的

italic ：倾斜的，斜体

oblique ：没有斜体的字体倾斜

快捷键：

fs font-style: italic;

fsn font-style: normal;

2.规定文字粗细的属性

格式：font-weight:bold;

单词取值：

bold 加粗

bolder 比加粗更粗

lighter 细线，默认就是细线

快捷键：

fw font-weight:;

fwb font-weight:bold;

fwbr font-weight:bolder;.

数字取值：

100-900之间的整百的数字

3.规定文字大小的属性

格式：font-size: 30px;

单位：px （像素 pxiel）

注意点：通过font-size设置大小一定要带单位，也就是一定要写px

快捷键：

fz30 font-size: 30px;

4.规定文字字体的属性

格式：font-family:"楷体";

4.1. 注意点：如果取值是中文，需要用双引号或单引号括起来

4.2. 设置的字体必须是用户电脑里面已经安装的字体

快捷键：ff font-family: ;

## 文字属性之字体补充

1.如果设置的字体不存在，那么系统会使用默认的字体来显示

宋体

2.如果设置的字体不存在，而我们又不想用默认的字体来显示怎么办？

可以给字体设置备选方案

格式：font-family:"字体1","备选方案1",...;

3.如果想给中文和英文分别单独设置字体，怎么办？

但凡是中文字体，里面都包含了英文

但凡是英文字体，里面都没有中文

也就是说中文字体可以处理英文，而英文字体不能处理中文

注意点：如果想给界面中的英文单独设置字样，

那么英文的字体必须写在中文的前面

补充在企业开发中最常见的字体有以下几个

1.中文：宋体/黑体/微软雅黑

2.英文：Times New Roman/Arial

还需要知道一点，就是并不是名称是英文就一定是英文字体

因为中文字体其实都有自己的英文名称，所以是不是中文字体主要看能不能处理中文

宋体 Simsun

黑体 SimHei

微软雅黑 Microsoft YaHei

## 文字属性的缩写

缩写格式：

font: style weight size family;

例如：

font：italic bold 30px "楷体";

注意点：

1.在这种缩写格式中有的属性可以省略

style可以省略

weight可以省略

2.在这种缩写格式中style和weight可以互换

3.在这种缩写格式中有的属性值是不可以省略的

size不能省略

family不能省略

4.size和family的位置是不能随便乱放的

size一定要写在family的前面，而且size和family必须写在所有属性的最后

## 文本属性

1.文本装饰的属性

格式：text-decoration: underline;

取值：

underline 下划线

line-through 删除线

overline 上划线

none 什么都没有，最常见的用途就是用于去掉超链接的下划线

快捷键：

td text-decoration: none;

tdu text-decoration: underiline;

tdl text-decoration: line-through;

tdo text-decoration: overline;

2.文本水平对齐的属性

格式：text-align: right;

取值：

left 左

right 右

center 中

快捷键

ta text-align: left;

tar text-align: right;

tac text-align: center;

3.文本缩进的属性

格式：text-indent: 2em;

取值：

2em，其中em是单位，一个em代表缩进一个文字的宽度

快捷键

ti text-indent: ;

ti2e text-indent: 2em;

## 文字颜色

1.在CSS中如何通过color属性来修改文字颜色

格式：color: 值;

取值：

1.1 英文单词

一般情况下，常见的颜色都有对应的英文单词，但是英文单词能够表达的颜色是有限的，也就是说不是所有的颜色都能够英文单词来表达

1.2 rgb

rgb其实就是三原色，其中r（red 红色） g（green 绿色） b（blue蓝色）

格式：rgb（0,0,0）

第一个数字就是用来设置三原色的光源元件红色显示的亮度

第二个数字就是用来设置三原色的光源元件绿色显示的亮度

第三个数字就是用来设置三原色的光源元件蓝色显示的亮度

这其中的每一个数字它的取值是0-255之间，0代表不发光，255代表发光，值越大就越亮

红色：rgb（255，0，0）

绿色：rgb（0，255，0）

蓝色：rgb（0，0，255）

黑色：rgb（0，0，0）

白色：rgb（255，255，255）

在前端开发中其实并不常用黑色

只要让红色/绿色/蓝色的值都一样就是灰色

而且如果这三个值越小，那么就越偏黑色，越大就越偏白色

例如： color: rgb（200,200,200）

1.3 rgba

rgba中的rgb和前面讲解的一样，只不过多了一个a，

那么这个a代表透明度，取值是0-1，取值越小就越透明

例如：color: rgba（255,0,0,0.5）;

1.4 十六进制

在前端开发中，通过十六进制来表示颜色，其实本质就是RGB，

十六进制中是通过每两位表示一个颜色

例如： #FFEE00 FF表示R EE表示G 00表示B

什么是十六进制？

十六进制和十进制一样都是一种计数的方式，

在十进制中取值范围0-9，逢十进一

在十六进制中取值范围0-F，逢十六进一

16进制 10进制

0 0

1 1

... ...

9 9

a 10

b 11

c 12

d 13

e 14

f 15

10 16

十六进制和十进制转换的公式

用十六进制的第一位\*１６　+　十六进制的第二位　＝　十进制

15 == 1\*16 + 5 = 21

12 == 1\*16 + 2 = 18

FF == 15\*16 +15 = 240+15 = 255

00 == 0\*16 + 0 = 0

1.5 十六进制缩写

在CSS中，只要十六进制的每两位的值都是一样，那么就可以简写为一位

例如：

#FFEE00 == #FE0

注意点：

1.5.1 如果当前颜色对应的两位数字不一样，那么就不能简写

#123456

1.5.2 如果两位相同的数字不是属于同一个颜色的，也不能简写

#122334

## id选择器

1.什么是id选择器？

作用：根据指定的id名称找到对应的标签，然后设置属性

格式：

#id名称{

属性: 值;

}

注意点：

1. 每个HTML标签都有一个属性叫做id，也就是说每个标签都可以设置id

2. 在同一个界面中id的名称是不可以重复的

3. 在编写id选择器时一定要在id名称前面加上“#”

4. id的名称是有一定的规范的

4.1 id的名称只能由字母/数字，下划线 a-z 0-9 \_

4.2 id名称不能以数字开关

4.3 id名称不能是HTML标签的名称， 不能是a h1 img input

4.4 在企业开发中，一般情况下如果仅仅是为了设置样式，我们不会使用id，因为在前端开发中id是留给js使用的

## 类选择器

1.什么是类选择器？

作用：根据指定的类名称找到对应的标签，然后设置属性

格式：

.类名{

属性: 值;

}

注意点：

1. 每个HTML标签都有一个属性叫做class，也就是说每个标签都可以设置类名

2. 在同一个界面中class的名称是可以重复的

3. 在编写class选择器时一定要在class名称前面加上“.”

4. 类名的命名规范与id的命名规范一样

5. 类名就是专门用来给某个特定的标签设置样式的

6. 在HTML中每个标签可以同时绑定多个类名

格式：

<标签名称 class="类名1 类名2 ...">

错误的写法：

<p class="para1" class="para2">

## id选择器和类选择器的区别

1.id和clss的区别？

1.1

id相当于人的身份证不可以重复

class相当人的名称可以重复

1.2

一个HTML标签只能绑定一个id名称

一个HTML标签可以绑定多个class名称

2.id选择器和class选择器的区别？

id选择器是以#开头

class选择器是以.开头

3.在企业开发中到底用id选择器还是用class选择器？

id一般情况下是给js使用的，所以除非特殊情况，否则不要使用id去设置样式

4.在企业开发中一个开发人员对类的使用可以看出这个开发人员的技术水平

一般情况下，在企业开发中要注重冗余代码的抽取，可以将一些公共的代码抽取到一个类选择器中，然后让标签和这个类选择器绑定即可

## 后代选择器

1.什么是后代选择器？

作用：找到指定标签的所有后代标签，设置属性

格式：

标签名称1 标签名称2{

属性: 值;

}

先找到名称叫做标签名称1的标签，然后再在这个标签下面去查找所有名称叫做标签名称2的标签，然后再设置属性

div p{}

注意点：

1.后代选择器必须用空格隔开

2.后代不仅仅是儿子，也包括孙子/重孙子，

只要最终是放到指定标签中的都是后代

3.后代选择器不仅仅可以使用标签名称，还可以使用其它选择器

## 子元素选择器

1.什么是子元素选择器？

作用：找到指定标签中所有特定的直接子元素，然后设置属性

格式：

标签名称1>标签名称2{

属性: 值;

}

先找到名称叫做“标签名称1”的标签，然后在这个标签中查找所有直接子元素名称叫做“标签名称2”的元素

注意点：

1.子元素选择器只会查找儿子，不会查找其它被嵌套的标签

2.子元素选择器之间需要用>符号连接 ，并且不能有空格

3.子元素选择器不仅仅可以使用标签名称，还可以使用其它选择器

4.子元素选择器可以通过>符号一直延续下去

## 子元素选择器和后代选择器的区别

1.后代选择器和子元素选择器之间的区别？

1.1

后代选择器使用空格作为连接符号

子元素选择器使用>作为连接符号

1.2

后代选择器会选中指定标签中，所有的特定后代标签，也就是会选中儿子/孙子...，只要是放到指定标签中的特定标签都会被选中；

子元素选择器只会选中指定标签中，所有的特定的直接标签，也就是只会选中选定的儿子标签

2.后代选择器和子元素选择器之间的共同点

2.1

后代选择器和子元素选择器可以使用标签名称/id名称/class名称来作为选择器

2.2

后代选择器和子元素选择器都可以通过各自的连接符号一直延续下去

选择器1 选择器2 选择器3 选择器4{}

选择器1>选择器2>选择器3>选择器4{}

3.在企业开发中如何选择

如果想选中指定标签中的所有特定标签，那么就使用后代选择器，

如果只想选中指定标签中的所有特定儿子标签，那么就使用子元素选择器

## 交集选择器

1.什么是交集选择器？

作用：给所有选择器选中的标签中，相交的那部分标签设置属性

格式：

选择器1选择器2{

属性: 值;

}

注意点：

1.选择器和选择器之间没有任何的连接符号

2.选择器可以使用标签名称/id名称/class名称

3.交集选择器仅仅作为了解，企业开发中用的并不多

## 并集选择器

1.什么是并集选择器

作用：给所有选择器选中的标签设置属性

格式：

选择器1,选择器2{

属性: 值;

}

注意点：

1.并集选择器必须使用“，”来连接

2.选择器可以使用标签名称/id标签/class名称

## 兄弟选择器

1.相邻兄弟选择器 CSS2

作用：给指定选择器**后面紧跟的那个选择器**选中的标签设置属性

格式：

选择器1+选择器2{

属性： 值；

}

注意点：

1.1 相邻兄弟选择器必须通过“+”连接

1.2 相邻兄弟选择器只能选中紧跟其后的那个标签

2.通用兄弟选择器 CSS3

作用：给指定选择器后面的所有选择器选中的所有标签设置属性

格式：

选择器~选择器2{

属性: 值；

}

注意点：

2.1 通用兄弟选择器必须通过“~”连接

2.2 通用兄弟选择器选中的是指定选择器**后面**某个选择器选中的所有标签，无论有没有被隔开都可以选中

## 序选择器

CSS3中新增的选择器最具代表性的就是序选择器

1.同级别的第几个

:first-child 选中同级别中的第一个标签 注意点：不区分类型

:last-child 选中同级别中的最后一个标签 注意点：不区分类型

:nth-child(n) 选中同级别中的第n个标签 注意点：不区分类型

:nth-last-child(n) 选中同级别中的倒数第n个标签 注意点：不区分类型

:only-child 选中父元素中唯一的元素

2.同级别中同类型的第几个

:first-of-type 选中同级别中同类型的第一个标签

:last-of-type 选中同级别中同类型的最后一个标签

:nth-of-type(n) 选中同级别中同类型的第n个标签

:nth-last-of-type(n) 选中同级别中同类型的倒数第n个标签

:only-of-type 选中父元素中某个类型的标签

3.序选择器补充

p:nth-child(odd) 选中同级别中的所有奇数

p:nth-child(even) 选中同级别中的所有偶数

p:nth-child(xn+y) x和y是用户自定义的，而n是一个计数器，从0开始递增

同理，同级别同类型中…

p:nth-of-type(odd)

p:nth-of-type(even)

p:nth-of-type(xn+y)

## 属性选择器

1.什么是属性选择器？

作用：根据指定的属性名称找到对应的标签，然后设置属性

格式：

[attribute]

作用：根据指定的属性名称找到对应的标签，然后设置属性

[attribute=value]

作用：找到的指定属性，并且属性的取值等于value的标签，然后设置设置属性

最常见的应用场景，就是用于区分input属性

input[type=password]{}

【这里的等号后面的“双引号”可写可不写？】

<input type="text" name="" id="">

<input type="password" name="" id="">

<input type="radio" name="" id="">

<input type="checkbox" name="" id="">

## 属性选择器补充

1.属性的取值是以什么开头的

[attribute|=value] CSS2 了解

[attribute^=value] CSS3 掌握

两者之间的区别：

CSS2中的只能找到value开头，并且value是被“-”和其它内容隔开的；

CSS3中的只要是以value开头的都可以找到，无论有没有被-隔开

2.属性的取值是以什么结尾的

[attribute$=value] CSS3 掌握

3.属性的取值是否包含某个特殊的值的

[attribute~=value] CSS2 了解

[attribute\*=value] CSS3 掌握

两者之间的区别：

CSS2中的只能找到独立的单词，也就是包含value，并且value是被空格隔开的；

CSS3中的只要包含value就可以找到

## 通配符选择器

1.什么是通配符选择器？

作用：给当前界面上所有的标签设置属性

格式：

\*{

属性: 值;

}

注意点：

由于通配符选择器是设置界面上所有的标签的属性，所以在设置之前会遍历所有的标签，如果当前界面上的标签比较多，那么性能就会比较差，所以在企业开发中一般不会使用通配符选择器

## 选择器练习

/\*标签选择器\*/  
 /\* p{ color: red; }\*/  
  
 /\*id选择器\*/  
 /\* #identity1{ color: blue; }  
 #identity2{ color: blue; }\*/  
  
 /\*类选择器\*/  
 /\* .para{ color: #ff0000;}\*/  
  
 /\*后代选择器\*/  
 /\*div p{color: blueviolet}\*/  
  
 /\*子元素选择器\*/  
 /\*div>p{color: red}\*/  
  
 /\*交集选择器\*/  
 /\*p.para{color: blue}\*/  
  
 /\*并集选择器\*/  
 /\* #identity1,identity2{color: red}\*/  
  
 /\*通用兄弟选择器\*/  
 /\* h1~p{color: red} \*/  
  
 /\*序选择器\*/  
/\* p:nth-of-type(n+1){ color: blue; } \*/  
  
 /\*属性选择器\*/  
/\* [class]{color: blueviolet} \*/  
/\* p[class]{color: red} \*/  
 [class=para]{  
 color: greenyellow;  
 font-size: 20px;  
 text-decoration: underline;  
  
 }

## CSS三大特性之继承性

1.什么是继承性？

作用：给父元素设置一些属性，子元素也可以使用，这个我们就称之为继承性

注意点：

1.并不是所有的属性都可以继承，只有以color/font-/text-/line开头的属性才可以继承

2.在CSS的继承中不仅仅是儿子可以继承，只要是后代都可以继承

3.继承性中的特殊性

3.1 a标签中的文字颜色和下划线是不能继承的

3.2 h标签中的文字大小是不能继承的

应用场景：

一般设置网页上的一些共性信息，例如网页的文字颜色，字体，文字大小等内容

body{}

## CSS三大特性之层叠性

1.什么是层叠性？

作用：层叠性就是CSS处理冲突的一种能力

注意点：

层叠性只有在多个选择器选中“同一个标签”，然后又设置了“相同的属性”，才会发生层叠性

CSS全称 Cascading Style Sheet

## CSS三大特性之优先级

1.什么是优先级？

作用：当多个选择器选中同一个标签，并且给同一个标签设置相同的属性时，如何层叠就由优先及来确定

2.优先级判断的三种方式

2.1 间接选中就是继承

如果是间接选中，那么就是谁离目标标签比较近就听谁的

2.2 相同的选择器（直接选中）

如果是直接选中，并且都是同类型的选择器，那么就是谁写在后面就听谁的

2.3 不同选择器（直接选中）

如果都是直接选中，并且不是相同类型的选择器，那么就会按照选择器的优先级来层叠

id>类>标签>通配符>继承>浏览器默认

## 优先级补充之“！impotant”

1.什么是!important？

作用：用于提升某个直接选中标签的选择器中的**某个属性**的优先级的，可以将被指定的属性的优先级提升为最高

注意点：

1.!important只能用于直接选中，不能用于间接选中

2.通配符选择器选中的标签也是直接选中的

3.!important只能提升被指定的属性的优先级，其它的属性的优先级不会被提升

4.!important必须写在属性值的分号前面

5.!important前面的感叹号不能省略

\*{color: blue **!important**;  
 font-size: 10px;}

## 优先级之权重问题

1.什么是优先级的权重？

作用：当多个选择器混合在一起使用时，我们可以计算权重来判断谁的优先级最高

2.权重的计算规则

2.1 首先先计算选择器中有多少个id，id多的选择器优先级最高

2.2 如果id的个数一样，那么再看类名的个数，类名个数多的优先级最高

2.3 如果类名的个数一样，那么再看标签名称的个数，标签名称个数多的优先级最高

2.4 如果id个数一样，类名个数也一样，标签名称个数也一样，那么就不会继续往下计算了 ，那么此时谁写在后面听谁的，也就是说优先级如果一样，谁写在后面听谁的

注意点：

只有选择器是直接选中标签的才需要计算权重，否则一定会听直接选中的选择器的

## div和span标签

1.什么是div？

作用：一般用于配合CSS完成网页的基本布局

2.什么是span？

作用：一般用于配合CSS修改网页中的一些局部信息

3.div和span有什么区别？

3.1 div会单独占领一行，而span不会单独占领一行

3.2 dic是一个容器级别的标签，而span是一个文本级的标签

4.容器级的标签和文本级的标签的区别？

容器级的标签中可以嵌套其它所有的标签；

文本级的标签只能嵌套文字/图片/超链接

容器级的标签：div h ul ol dl li dt dd...

文本级的标签：sapn p buis stong em ins del...

注意点：

哪些标签是文本级的，哪些标签是容器级的，我们不用刻意去记忆，在企业开发中一般情况下要嵌套都是嵌套在div中，或者按照组标签来嵌套

## CSS元素的显示模式（行内还是块级）

在HTML中HTML将所有的标签分为两类，分别是容器级和文本级；

在CSS中CSS也将所有的标签分为两类，分别是块级元素和行内元素

1.什么是块级元素，什么是行内元素？

块级元素会独占一行；

行内元素不会独占一行

容器级的标签：div h ul ol dl li dt dd...

文本级的标签：span p b u i s strong em ins del...

块级元素：p div h ul ol dl li dt dd...（只有p和上面不一样）

行内标签：span b u i s strong em ins del...

2.块级元素和行内元素的区别

2.1 块级元素

独占一行

如果没有设置宽度，那么默认和父元素一样宽；

如果设置了宽高，那么就按照设置的来显示

2.2 行内元素

不会独占一行

如果没有设置宽度，那么默认和内容一样宽；

行内元素是不可以设置宽度和高度的

2.3 行内块级元素

为了能够让元素既不独占一行，又可以设置宽度和高度，那么就出现了行内块级元素

## CSS显示模式转换

1.如何转换CSS元素的显示模式？

设置元素的display属性

2.display取值

block 块级

inline 行内

inline-block 行内块级

3.快捷键

di display: inline;

db display: block;

dib display: inline-block;

## 背景颜色

1.如何设置标签的背景颜色

在CSS中有一个background-color属性，就是专门用来设置标签的背景颜色的

2.取值：

具体单词

rgb

rgba

十六进制

## 背景图片

1.如何设置背景图片？

在CSS中有一个叫做background-image: url();的属性，就是专门用于设置背景图片的

注意点：

1.图片的地址必须放在url()中，图片的地址可以是本地的地址，也可以是是网络的地址

2.如果图片的大小没有标签的大小大，那么会自动在水平和垂直方向平铺和填充

3.如果网页上出现了图片，那么浏览器会再次发送请求获取图片

## 背景平铺属性

1.如何控制背景图片的平铺方式？

在CSS中有一个background-repeat属性，就是专门用于控制背景图片的平铺方式的

取值：

repeat 默认，在水平和垂直都需要平铺

no-repeat 在水平和垂直都不需要平铺

repeat-x 只在水平方向平铺

repeat-y 只在垂直方向平铺

应用场景：

可以通过背景图片的平铺来降低图片的大小，提升网页的访问速度

## 背景定位属性

1.如何控制背景图片的位置？

在CSS中有一个叫做background-position的属性，就是专门用于控制背景图片的位置

2.格式：

background-position: 水平方向 垂直方向;

3.取值

3.1具体的方位名词

水平方向：left center right

垂直方向：top center bottom

3.2具体的像素

例如：background-position: 100px 200px;

记住一定要写单位，也就是一定要写px

记住具体的像素是可以接收负数的

注意点：

同一个标签可以同时设置背景颜色和背景图片，如果颜色和图片同时存在，那么图片会覆盖颜色

## 背景缩写

1.背景属性缩写的格式：

background：背景颜色 背景图片 平铺方式 关联方式 定位方式；

2.注意点：

background属性中，任何一个属性都可以被省略

3.什么是背景关联方式？

默认情况下背景图片会随着滚动条的滚动条滚动，如果不想让背景图片随着滚动条的滚动而滚动，那么我们可以修改背景图片和滚动条的关联方式

4.如何修改背景关联方式？

在CSS中有一个叫做background-attachmnet的属性，这个属性就是专门用于修改关联方式的

格式：

background-attachment: scroll;

取值：

scroll 默认值，会随着滚动条的滚动而滚动；

fixed 不会随着滚动条的滚动而滚动

## 背景和插入图片的区别

1.背景图片和插入图片区别？

1.1

背景图片仅仅是一个装饰，不会占用空位置

插入图片会占用位置

1.2

背景图片有定位属性，所以可以很方便的控制图片的位置；

插入图片没有定位属性，所以控制图片的位置不太方便

1.3

插入图片的语义比背景图片的语义要强，所以在企业开发中如果你的图片想被搜索引擎收录，那么推荐使用插入图片

## CSS精灵图

1.什么是CSS精灵图？

CSS精灵图是一种图像合成技术

2.CSS精灵图作用？

可以减少请求的次数，以及可以降低服务器处理压力

3.如何使用CSS精灵图？

CSS的精灵图需要配合背景图片和背景定位来使用

## 边框属性上

1.什么是边框？

边框就是环绕在标签宽度和高度周围的线条

2.边框属性的格式

2.1 连写（同时设置四条边的边框）

border: 边框的**宽度** 边框的**样式** 边框的**颜色**

快捷键

bd+ border: 1px solid #000

注意点：

1.连写格式中颜色属性可以省略，省略之后默认就是黑色

2.连写格式中样式不能省略，省略之后就看不到边框了

3.连写格式中宽度可以省略，省略之后还是可以看到边框

2.2 连写（分别设置四条边的边框）

border-top: 边框的宽度 边框的样式 边框的颜色

border-right: 边框的宽度 边框的样式 边框的颜色

border-bottom: 边框的宽度 边框的样式 边框的颜色

border-left: 边框的宽度 边框的样式 边框的颜色

快捷键：

bt+： border-top: 1px solid #000;

br+

bb+

bl+

## 边框属性下

2.3 连写（分别设置四条边的边框）

border-width: 上 右 下 左

border-style: 上 右 下 左

border-color: 上 右 下 左

注意点：

1.这三个属性的取值是按照顺时针来赋值

也就是按照上右下左来赋值，而不是按照日常生活中的上下左右

2.这三个属性的取值省略时的规律

2.1 上 右 下 左 > 上 右 下 > 左边的取值和右边的一样

2.2 上 右 下 左 > 上 右 > 左边的取值和右边的一样 下边的取值和上边一样

2.3 上 右 下 左 > 上 > 右下左边取值和上边一样

3.非连写（方向+要素）

border-top-width: 5px;

border-top-style: solid;

border-top-color: blue;

border-right-width: 10px;

border-right-style: dashed;

border-right-color: green;

border-bottom-width: 15px;

border-bottom-style: dotted;

border-bottom-color: purple;

border-left-width: 20px;

border-left-style: double;

border-left-color: pink;

### 下拉三角



.box6{  
 width: 0;  
 height: 0;  
 border: 25px solid;  
 border-color: #f00 #0ff #f0f #00f;  
 /\*border-bottom: none;\*/  
}

让border-bottom:none 则会出现下拉三角

再把后三个的颜色改为#fff #fff #fff，则出现

## 内边距属性

1.什么是内边距？

边框和内容之间的距离就是内边距

2.格式

2.1 非连写

padding-top: ;

padding-right: ;

padding-bottom: ;

padding-left: ;

2.2 连写

padding: 上 右 下 左;

3.这个内边距宽度属性的取值省略时的规律

3.1 上 右 下 左 > 上 右 下 > 左边的取值和右边的一样

3.2 上 右 下 左 > 上 右 > 左边的取值和右边的一样 下边的取值和上边一样

3.3 上 右 下 左 > 上 > 右下左边取值和上边一样

注意点：

1.给标签设置内边距之后，标签占有的宽度和高度会发生变化

2.给标签设置内边距之后，内边距也会有背景颜色

## 外边距属性

1.什么是外边距？

标签和标签之间的距离就是外边距

2.格式：

2.1 非连写

margin-top: ;

margin-right: ;

margin-bottom: ;

margin-left: ;

2.2 连写

margin: 上 右 下 左:

3.这个外边距宽度属性的取值省略时的规律

3.1 上 右 下 左 > 上 右 下 > 左边的取值和右边的一样

3.2 上 右 下 左 > 上 右 > 左边的取值和右边的一样 下边的取值和上边一样

3.3 上 右 下 左 > 上 > 右下左边取值和上边一样

4.注意点：

外边距的那一部分是没有背景颜色的

## 外边距合并现象

在默认布局的垂直方向上，默认情况下外边距是不会叠加的，会出现合并现象，谁的外边距比较大就听谁的

## CSS盒子模型

1.什么是CSS盒子模型？

CSS盒子模型仅仅是一个形象的比喻，HTML中所有的标签都是盒子

结论

1.在HTML中所有的标签都可以设置

宽度/高度 == 指定可以存放内容的区域

内边距 == 填充物

边框 == 手机盒子自己

外边距 == 盒子和盒子之间的间隙

## 盒子模型的宽度和高度

1.内容的宽度和高度

就是通过标签的width/height属性设置的宽度和高度

2.元素的宽度和高度（除去margin的其它部分）

宽度 = 左边框 + 左内边距 + width + 右内边距 + 右边框

高度 = 上边框 + 上内边距 + height + 下内边距 + 下边框

3.元素空间的宽度和高度

宽度 = 左外边距 + 左边框 + 左内边距 + width + 右内边距 + 右边框 + 右外边距

高度 = 上边框 + 上内边距 + height + 下内边距 + 下边框

## 盒子box-sizing属性

1.CSS3中新增了一个box-sizing属性，这个属性可以保证我们给盒子新增padding和border之后，盒子元素的宽度和高度不变

2.box-sizing属性是如何保证增加padding和border之后，盒子元素的宽度和高度不变。

和我们前面学习的原理一样，增加padding和border之后要想保证盒子元素的宽高不变，那么就必须减去一部分内容的宽度和高度

3.box-sizing取值

3.1 content-box

元素的宽高 = 边框 + 内边距 + 内容宽高

3.2 border-box

元素的宽高 = width/height宽高

注意点：

1.如果两个盒子是嵌套关系，那么设置了里面一个盒子的顶部的外边距，外面的一个盒子也会被顶下来

2.如果外面的盒子不想一起被顶下来，那么可以给外面的盒子添加一个边框属性

3.在企业开发中，一般情况下如果需要控制嵌套关系盒子之间的距离，应首先考虑padding，其次再考虑margin，margin的本质是用于控制兄弟关系之间的间隙的

注意点：

1.在嵌套关系的盒子中，我们可以利用margin：0 auto；的方式来让里面的盒子在外面的盒子中水平居中

2.margin: 0 auto; 只对水平方向有效，对垂直方向无效

## 盒子居中和内容居中

text-align: center; 和 margin: 0 auto; 区别

text-align: center; 作用：设置盒子中存储的文字/图片水平居中

margin: 0 auto; 作用：让盒子自己水平居中

## 清空默认边距

1.为什么要清空默认边距（外边距和内边距）

在企业开发中为了更好的控制盒子的宽高和计算盒子的宽高等等，所以在企业开发中，编写代码之前第一件事情就是清空默认的边距

2.如何清空默认的边距

格式

\*{

margin: 0;

padding: 0;

}

3.注意点

通配符选择器会找到（遍历）当前界面中所有的标签，所以性能不好

企业开发中可以从这个网址中拷贝

http://yui.yahooapis.com/3.18.1/build/cssreset/cssreset-min.css

【

body,div,dl,dt,dd,ul,ol,li,h1,h2,h3,h4,h5,h6,pre,code,form,fieldset,legend,input,textarea,p,blockquote,th,td{margin:0;padding:0}

】

## 行高和字号

1.什么是行高？（行高是每一行内容的高度，盒子高是标签的高度）

在CSS中所有的行都有行高

注意点：

行高和盒子高不是同一个概念

规律：

1.文字在行高中默认是垂直居中的

2.在企业开发中我们经常将盒子的高度和行高设置为一样，那么这样就可以保证一行文字在盒子的高度中是垂直居中的

简而言之就是：要想一行文字在盒子中垂直居中，那么只需要设置这行文字的“行高等于盒子的高”即可

3.在企业开发中如果一个盒子有多行文字，那么我们就不能使用设置行高等于盒子高来实现让文字垂直居中，只能通过设置padding来让文字居中

## 还原字体和字号

注意点：

1.在企业开发中，如果一个盒子中存储的是文字，那么一般情况下我们会以盒子左边的内边距为基准，不会以右边的内边距为基准，因为这个右边的内边距有误差

2.右边内边距的误差从何而来？因为右边如果放不下一个文字，那么文字就会换行显示，所以文字和内边距之间的距离就有了误差

3.顶部的内边距并不是边框文字顶部的距离，而是边框到行高顶部的距离

## 网页的布局方式

1.什么是网页的布局方式？

网页的布局方式其实就是指浏览器是如何对网页中的元素进行排版的

1.标准流（文档流/普通流）排版方式

1.1 其实浏览器默认的排版方式就是标准流的排版方式

1.2 在CSS中将元素分为三类，分别是块级元素/行内元素/行内块级元素

1.3 在标准流中有两种排版方式，一种是垂直排版，一种是水平排版

垂直排版，如果元素是块级元素，那么就会垂直排版

水平排版，如果元素是行内元素/行内块级元素，那么就会水平排版

2.浮动流排版方式

2.1 浮动流是一种“半脱离标准流”的排版方式

2.2 浮动流只有一种排版方式，就是水平排版，它只能设置某个元素左对齐或者右对齐

注意点：

2.1.浮动流中没有居中对齐，也就是没有center这个取值

2.2.在浮动流中不可以使用margin: 0 auto;的

特点：

1.在浮动流中是不区分块级元素/行内元素/行内块级元素的，无论是块级元素/行内元素/行内块级元素都可以水平排版

2.在浮动流中无论是块级元素/行内元素/行内块级元素都可以设置宽高

3.综上所述，浮动流中的元素和标准流中的行内块级元素很像

3.定位流排版方式

## 浮动元素的脱标

1.什么是浮动元素的脱标？

脱标：脱离标准流

当某一个元素浮动之后，那么这个元素看上去就像被从标准流中删除了一样，这个就是浮动元素的脱标

2.浮动流脱标之后会有什么影响？

如果前面一个元素浮动了，而后面一个元素没有浮动，那么这个时候前面的一个元素就会盖住后面的一个元素

### 浮动元素的排序规则

1.浮动元素的排序规则

1.1 相同方向上的浮动元素，先浮动的元素会显示在前面，后浮动的元素会显示在后面

1.2 不同方向上的浮动元素，左浮动会找左浮动，右浮动会找右浮动

1.3 浮动元素浮动之后的位置，由浮动元素之前在标准流中的位置来确定

## 浮动元素贴靠现象

什么是浮动元素贴靠现象?

如果父元素的宽度能够显示所有浮动元素, 那么浮动的元素会并排显示

如果父元素的宽度不能显示所有浮动元素, 那么会从最后一个元素开始往前贴靠

如果贴靠了前面所有浮动元素之后都不能显示, 最终会贴靠到父元素的左边或者右边

## 浮动元素字围现象

1.什么是浮动元素字围现象？

浮动元素不会挡住没有浮动元素中的文字，没有浮动的文字会自动给浮动的元素让位置，这个就是浮动元素字围现象

2.应用场景：

图文混排

## 浮动元素高度问题

1.在标准流中内容的高度可以撑起父元素的高度

2.在浮动流中浮动的元素是不可以撑起父元素的高度的

## 清除浮动方式一

1.清除浮动方式一：

给前面一个父元素设置高度

注意点：

在企业开发中，我们能不写高度就不写高度，所以这种方式用得很少

## 清除浮动方式二

1.清除浮动的第二种方式

给后面的盒子添加clear属性

clear属性取值：

none：默认取值，按照浮动元素的排序规则来排序（左浮动找左浮动，右浮动找右浮动）

left：不要找前面的左浮动元素

right：不要找前面的右浮动元素

both：不要找前面的左浮动元素和右浮动元素

注意点：

当我们给某个元素添加clear属性之后，这个属性的margin属性就会失效

## 清除浮动方式三

1.清除浮动的第三种方式

隔墙法

2.外墙法

2.1 在两个盒子中间添加一个额外的块级元素

2.2 给这个额外添加的块级元素设置clear: both;属性

注意点：

外墙法它可以让第二个盒子使用margin-top属性

外墙法不可以让第一个盒子使用margin-bottom属性

3.内墙法

3.1 在第一个盒子中所有子元素最后添加一个额外的块级元素

3.2 给这个额外添加的块级元素设置clear: both;属性

注意点：

内墙法它可以让第二个盒子使用margin-top属性

内墙法它可以让第一个盒子使用margin-bottom属性

4.外墙法和内墙法之间的区别

外墙法不能撑起第一个盒子的高度，而内墙法可以撑起第一个盒子的高度

5.在企业开发中不常用隔墙法来清除浮动

## 伪元素选择器

1.什么是伪元素选择器？

伪元素选择器作用就是给指定标签的内容前面添加一个子元素或者给指定标签的内容后面添加一个子元素

2.格式：

标签名称::before{

属性名称: 值;

}

给指定标签的内容前面添加一个子元素

标签名称::after{

属性名称: 值;

}

给指定标签的内容后面添加一个子元素

## 清除浮动方式四

.box1::after{  
 /\*设置添加的子元素的内容为空\*/  
 content:"";  
 /\*设置添加的子元素为块级元素\*/  
 display: block;  
 /\*设置添加的子元素的高度为0\*/  
 height: 0;  
 /\*设置添加的子元素看不见\*/  
 visibility: hidden;  
 /\*给添加的子元素设置clear: both;\*/  
 clear: both;  
}

## 清除浮动方式五

1.overflow: hidden;作用

1.1 可以将超出标签范围的内容裁剪掉

1.2 清除浮动

1.3 可以通过overflow: hidden;让里面的盒子油墨margin-top之后，外面的盒子不被顶下来

## 定位流

1.定位流分类

1.1 相对定位

1.2 绝对定位

1.3 固定定位

1.4 静态定位

2.什么是相对定位？

相对定位就是相对于自己以前在标准流中的位置来移动

3.相对定位注意点

3.1 相对定位是不脱离标准流的，会继续在标准流中占用一份空间

3.2 在相对定准中同一个方向上的定位属性只能使用一个

3.3 由于相对定位是不脱离标准流的，所以在相对定位中是区分块级元素/行内元素/行内块级元素

3.4 由于相对定位是不脱离标准流的，并且相对定位的元素会占用标准流中的位置，所以当给相对定位的元素设置margin/padding等属性的时候会影响到标准流的布局

4.相对定位应用场景

4.1 用于对元素进行微调

4.2 配合后面学习的绝对定位来使用

## 定位流-绝对定位

1.什么是绝对定位？

绝对定位就是相对于body来定位

2.绝对定位注意点

2.1 绝对定位的元素是脱离标准流的

2.2 绝对定位的元素是不区分块级元素/行内元素/行内块级元素

## 绝对定位-参考点

1.规律

1.1 默认情况下所有的绝对定位的元素，无论有没有祖先元素，都会以body作为参考点

1.2 如果一个绝对定位的元素有祖先元素，并且祖先元素也是定位流，那么这个绝对定位的元素就会以定位流的那个祖先元素作为参考点

1.2.1 只要是这个绝对定位元素的祖先元素都可以

1.2.2 指的定位流是绝对定位/相对定位/固定定位

1.2.3 定位流中只有静态定位不行

1.3 如果一个绝对定位的元素有祖先元素，并且祖先元素也是定位流，而且祖先元素中有多个元素都是定位流，那么这个绝对定位的元素会以离它最近的那个定位流的祖先元素为参考点

## 绝对定位-注意点

1.如果一个绝对定位的元素是以body作为参考点，那么其实是以网页首屏的宽度和高度作为参考点，而不是以整个网页的宽度和高度作为参考点

2.一个绝对定位的元素会忽略祖先元素的padding

## 绝对定位-子绝父相

相对定位弊端：

相对定位不会脱离标准流，会继续在标准流中占用一份空间，所以不利于布局界面

绝对定位弊端：

默认情况下绝对定位的元素会以body作为参考点，所以会随着浏览器的宽度高度的变化面变化

子绝父相：

子元素用绝对定位，父元素用相对定位

## 绝对定位-水平居中

1.如何让绝对定位的元素水平居中？

只需要设置绝对定位元素的 left: 50%

然后再设置绝对定位元素的 margin-left: -元素宽度的一半px【-这里是负号】

## 定位流-固定定位

1.什么是固定定位？

固定定位和前面学习的背景关联方式很像.

背景定位可以让背景图片不随着滚动条的滚动而滚动，

而固定定位可以让某个盒子不随着滚动条的滚动而滚动

注意点：

1.固定定位的元素是脱离标准流的，不会占用标准流中的空间

2.固定定位和绝对定位一样不区分行内/块级/行内块级

固定定位应用场景：广告

## 定位流-z-index属性

1.什么是z-index属性？

默认情况下所有的元素都有一个默认的z-index属性，取值为0，

z-index属性的作用是专门用于控制定位流元素的覆盖关系的

1.默认情况下定位流的元素会盖住标准流的元素

2.默认情况下定位流的元素后面编写的倒盖住前面编写的

3.如果定位流的元素设置了z-index属性，那么谁的z-index属性比较大，谁就是显示在上面

注意点：

1.从父现象

1.1 如果两个元素的父元素都没有设置z-index，那么谁的z-index属性比较大，谁就显示在上面

1.2 如果两个元素的父元素设置了z-index属性，那么子元素的z-index属性就会失效，也就是说谁的父元素的z-index属性比较大谁显示在上面

## a标签的伪类选择器

1.通常我们的观察发现a标签存在一定的状态

1.1 默认状态，从未被访问过

1.2 被访问过的状态

1.3 鼠标悬停在a标签上的状态

1.4 鼠标长按的状态

2.什么是a标签的伪类选择器？

a标签的伪类选择器是专门用来修改a标签不同状态的样式的

3.格式

:link 修改从未被访问过的状态下的样式

:visited 修改被访问过的状态下样式

:hover 修改鼠标悬停在a标签上时状态下的样式

:active 修改鼠标长按状态下的样式

4.注意点

4.1 a标签的伪类选择器可以单独出现，也可以一起出现

4.2 a标签的伪类选择器如果一起出现，那么有严格的顺序，要求编写的顺序，必须严格地遵守爱恨原则 love hate

4.3 如果默认状态的样式和被访问过状态的样式一样，那么可以缩写

注意点：

1. 在企业开发中编写a标签的伪类选择器最好写在标签选择器的后面

2.在企业开发中和a标签相关的属性都写在标签选择器中（显示模式/宽度/调试/padding/margin）

3.在企业开发中和a标签文字/背景相关的都写在伪类选择器中

## 过渡模块

1.过渡三要素

1.1 必须要有属性发生变化

1.2 必须告诉系统哪个属性需要执行过渡效果

1.3 必须告诉系统过渡效果持续时长

2.注意点

当多个属性需要同时执行过渡效果时用逗号隔开即可

transition-property: width,background-color;

transition-duration: 5s,5s;

## 过渡模块连写

1.过渡连写格式

transition: width 5s linear 0s;

transition: 过渡属性 过渡时长 运动速度 延迟时间;

2.过渡连写注意点

2.1 和分开写一样，如果想给多个属性添加过渡效果也是用逗号隔开即可

2.2 连写的时候可以省略后面的两个参数，因为只要编写了前面的两个参数就已经满足了过渡的三要素

2.3 如果多个属性持续时间/运动的速度/延迟的时间都一样，那么可以简写为

transition:all 0s;

## 过渡模块弹性效果（拉长）

1.编写过渡套路

1.1 不要管过渡，先编写基本界面

1.2 修改我们认为需要修改的属性

1.3 再回过头去给被修改属性的那个元素添加过渡即可

## 过渡模块-手风琴效果