CSS叫做层叠样式表，用来设置页面中元素的样式。背景颜色、字体颜色、字体大小。。。

CSS负责结构、表现、行为中的表现

**编写的位置**

1.内联样式

将样式编写到标签的style属性中<p style="color:red;"></p>

这种样式只会对当前标签起作用，不能对样式进行复用，不方便后期维护，不推荐使用

2.内部样式表

将样式表编写到head中的style标签中<style type="text/css"></style>

内部样式表，进一步将表现和结构分离，可以同时为多个元素设置样式，方便后期的维护

3.外部样式表

将样式表编写到外部的CSS文件中，然后通过link标签将外部文件引入

<link rel="stylesheet" type="text/css" href="文件的路径"/>

将样式编写到外部样式表中，可以在不同的页面中使用同一个样式表，完全将表现和结构分离，方便后期的维护，推荐使用的方式

**基本语法**

**选择器 元素=标签**

通过选择器可以选中页面中的一组元素，然后为其设置样式

1. 元素选择器

根据标签名，选中页面中的指定元素

语法：标签名{ }

例子：div{} p{} h1{}

1. id选择器

根据元素的id属性值选中一个唯一的元素

语法：#id {}

例子： #box1{} #hello{}

1. 类选择器(可重复)

根据元素的class属性值，选中一组元素

语法：.class{}

例子： .hello{} .box{}

1. 通配选择器

选中页面中的所有元素

语法：\*{}

通配选择器的性能比较差，尽量避免使用

1. 并集选择器

可以同时选中符合多个选择器的元素

语法：选择器1,选择器2,选择器N{}

例子： div,p,#box,.hello{}

1. 交集选择器

可以选中满足多个条件的元素

语法：选择器1选择器2选择器N{}

例子：p.hello{} 适用class选择器，id选择器一般不用这个方法

1. 后代元素选择器

选中指定元素的指定后代元素

语法：祖先元素 后代元素{}

例子：div span {} div p{}

1. 子元素选择器

选中指定元素的指定子元素

语法：父元素 > 子元素 {}

例子：div > span {} div > p{}

**声明块**

声明块中实际上就是一个一个CSS声明

每一个CSS声明都是一个样式，实际上就是一个名值对的结构

名和值之间使用 : 链接

左边是样式的名字 : 右边是样式的值

每一个声明以;结尾

例子color:red; font-size:20px;

**元素之间的关系**

父元素 直接包含子元素的的元素叫做父元素

子元素 直接被父元素包含的元素叫做子元素

祖先元素 直接或间接包含后代元素的元素叫做祖先元素，父元素也是祖先元素

后代元素 直接或间接被祖先元素包含的元素叫后代元素，子元素也是后代元素

兄弟元素 拥有相同父元素的元素叫做兄弟元素

**块元素和内联元素**

**块元素** 独占页面中的一行，无论他的内容的多少

一般使用块元素对页面进行布局

常见的块元素div p h1~h6

**内联元素** 只占用自身的大小，不会独占一行

内联元素也叫行内元素（inline）

一般内联元素都是用来为文本来设置效果

常见的内联 span（没有任何的语义，span标签专门用来选中文字） a img

包裹规则

一般都是使用块元素去包裹内联元素，而不会使用内联去包裹块元素

a元素可以包含任意元素，除了a本身

p元素不能包含任何块元素

**伪类和伪元素**

伪类和伪元素用来表示元素所处的一个特殊的状态，或者是一个特殊的位置

:link 表示一个普通的链接（未访问过的链接）例： a:link{ }

:visited 表示访问过的链接

:hover 鼠标移入的链接，也可以为其他元素设置hover

:active 正在被点击的链接，也可以为其他元素设置active

以上顺序不能变

:focus 表示元素获取焦点的状态，一般用于文本框 例： input：focus{ }

::selection 表示内容被选中的状态 在火狐中使用::-moz-selection来代替

:first-letter 表示第一个字符 例： p:first-letter{ }

:first-line 表示文字的第一行

:before 选中元素的最前边

一般该伪类都会结合content一起使用，通过content可以向指定位置添加内容

:after 选中元素的最后边

一般该伪类都会结合content一起使用，通过content可以向指定位置添加内容

**属性选择器** 根据元素的属性选择指定元素

[属性名] 选取含有指定属性的元素 例: <p title=”abc”>11 </p> p[title]{ }

[属性名="属性值"] 选取属性值等于指定值的元素

[属性名^="属性值"] 选取属性值以指定内容开头的元素

[属性名$="属性值"] 选取属性值以指定内容结尾的元素

[属性名\*="属性值"] 选取属性值中包含指定内容的元素

**兄弟元素选择器**

选取后一个兄弟元素 前一个 + 后一个

选取后边所有的兄弟元素 前一个 ~ 后边所有

**子元素的伪类**

:first-child 寻找父元素的第一个子元素，在所有的子元素中排序 例: p:first-child{ }

:last-child 寻找父元素的最后一个子元素，在所有的子元素中排序

:nth-child 寻找父元素中的指定位置子元素，在所有的子元素中排序

例子 p:nth-child(3)可以使用even，来找到偶数的子元素

可以使用odd，来找到奇数的子元素

:first-of-type 寻找指定类型中的第一个子元素

:last-of-type 寻找指定类型中的最后一个子元素

:nth-of-type 寻找指定类型中的指定子元素

**否定伪类**  从一组元素中将符合要求的元素剔除出去

语法： :not(选择器)

例子： .abc:not(div)

**样式的继承**

1. 为祖先元素设置样式，会同时应用到它的后代元素上，这一特性称为样式的继承。
2. 通过样式的继承可以将一些样式统一设置个祖先元素，这样所有的后代都会应用到相同的样式。
3. 但是并不是所有的样式都会继承，比如：背景相关的，边框相关的，定位相关的。具体参考文档

**选择器的优先级，**当使用选择器为元素设置样式时，如果样式发生了冲突，采用哪个样式由选择器的优先级来决定。

优先级：内联样式 1000 、 id选择器 100 、 类和伪类选择器10

元素选择器 1 、 通配选择器 0 、 继承的样式 没有优先级

1. 当样式发生冲突时，需要将相关的选择器优先级进行求和计算，优先级高的优先显示，如果优先级一样，则显示靠后的样式

2.优先级计算时，总大小不能超过他的最大的数量级 例：10个优先级为10 最高是99

3.可以在样式后边添加一个!important，如果样式中添加了该内容，则该样式将会获取最大的优先级，将会优先于所有的样式显示，包括内联样式，但是这个属性要慎用。

**选择器的性能**览器在解析一组选择器时，是从后边往前一个一个的解析的

如果选择器编写过于长的话，浏览器解析起来性能会比较差，所以在编写选择器时，越短越好。

\*通配选择器，性能也比较差，要避免使用通配选择器

**长度单位**

px像素，像素就是构成一个图片的最小的单位，我们的屏幕就是由一个一个像素点构成

一个像素指的就是一个像素点

在不同的显示器中，一个像素的大小是不同的，越清晰的屏幕像素越小

% 可以将一个元素的样式值设置为一个百分比的值，这样浏览器将会根据父元素的值去计算出相应的值

当父元素的值改变时，子元素的值会按照一定比例一起改变，经常用于**自适应**的页面

em 会相对于当前元素的字体大小来计算

1em = 1font-size

em经常用于设置文字相关的一些样式，因为当文字大小发生改变时，em会随之改变

**颜色单位**

颜色单词 ,直接使用英文单词来表示颜色:red green blue orange

RGB值 ,通过红 绿 蓝三元色的不同组合来搭配出各种不同的颜色

语法：rgb(红色,绿色,蓝色) 这三个值需要一个0-255之间的值

0表示没有 255表示最大

rgb(50,200,30)

也可以使用百分数来设置RGB值，需要0%-100%之间的值,百分数最终也是转换为0-255的

0%相当于0 100%相当于255

rgb(100%,0%,0%)

十六进制RGB值,也是一种RGB值的表示方式，不同的是它使用的是16进制数字来表示而不是10进制

语法：#红色绿色蓝色

这里的颜色需要一个00-ff之间的值

例子：#ff0000

如果颜色的是两位两位重复的，可以进行简写

比如 #aabbcc 可以写成 #abc 比如 #bbffaa 可以写成 #bfa

**文本样式**

**字体**

color 字体的颜色

**font-size**字体的大小

浏览器中默认的字体大小一般都是**16px**，而我们开发时一般会统一为12px

font-family 设置文字的字体 例如： font-family:微软雅黑;

font-style 设置斜体 italic（斜体） oblique(偏斜体) normal（正常，默认）

**font-weight** 设置文字的加粗效果 bold（加粗）lighter（细） normal（正常）100-900

font-variant 小型大写字母 small-caps

font 文字的简写属性 例: font: bold 60px “宋体”;

可以同时设置所有的字体相关的样式

语法：font: [加粗 斜体 小大字母] 大小[/行高] 字体

加粗，斜体，小大字母，顺序无所谓，写不写都行，如果不写在使用默认值

**文本大小，和字体必须写**，且大小必须是倒数第二个，字体必须是最后一个

大小后可以设置行高，可写可不写，如果不写则使用默认值

**文本样式**

line-height 行高

文本默认都是在行高中垂直居中的,通过line-height可以修改行高

行间距 = 行高 - 字体大小

在font也可以指定行高（字体大小/行高）如果不制定则用默认值，行高设置要在font后

text-transform 设置文本的大小写

text-decoration 设置文本修饰

text-align 设置文本对齐

text-indent 设置首行缩进

它需要一个长度单位，如果是正值则首行向右移动，如果是负值则向左移动

letter-spacing 字符间距

word-spacing 单词间距

**背景**

background-color背景颜色

background-image背景图片，需要一个url地址作为参数

例子：background-image:url(图片的路径)

background-repeat 设置背景图片重复方式

可选值：repeat默认值，背景图片会平铺显示

沿x轴和y轴双方向重复

no-repeat背景图片不重复

repeat-x背景图片沿水平方向重复

repeat-y背景图片沿垂直方向重复

background-position设置背景图片的位置

设置方式一，可以直接通过几个位置的关键字来设置图片的位置

top left right bottom center

可以通过以上关键字两两组合的形式，将背景图片设置到元素的任意位置

如果仅仅指定一个值，则第二个值默认是center

设置方式二, 可以直接指定两个值，来设置背景图片的偏移量

例子： background-position : x轴偏移量 y轴偏移量;

x轴偏移量，用来指定图片的水平位置,如果指定一个正值，则图片向右移动，如果指定一个负值，则图片向左移动。y轴偏移量，用来指定图片的垂直位置，如果指定一个正值，则图片向下移动，如果指定一个负值，则图片向上移动

background-attachment用来设置背景是否随页面滚动

可选值scroll 默认值，背景图片会随页面一起滚动

fixed背景图片不随页面滚动，会固定在页面的指定位置，设置该属性的背景，则背景会永远相对于浏览器窗口进行定位，一般这种背景都会设置给body

background背景的简写属性

可以通过它来设置所有的背景相关的样式

该简写属性没有顺序的要求，也没有数量的要求，不写的属性使用默认值

opacity用来设置背景的不透明度，可选值0-1

0表示完全透明 1表示完全不透明 0.5半透明

IE8及以下的浏览器不支持该样式，可以使用滤镜来代替 filter:alpha(opacity=值)

这里值需要一个0-100之间的值 0相当于完全透明 100完全不透明