# fsCSS

解决问题

垂直方向外边距重叠问题：

* + 垂直方向外边距重叠问题：
            父元素中有一个子元素。
            当给子元素设置上外边距的时候，发现上外边距是设置给了父元素，导致父元素和子元素一块下移。
    解决方案：
            1. 给父元素加 overflow:hidden; 父元素就开启了 BFC 格式，那父元素的布局就不会受到子元素的影响。
            2. 给子元素加 float:left; 子元素就开启了 BFC格式，子元素的布局就不会影响到父亲
                或者 给父元素 加 float:left; 父元素开启了 BFC格式，父元素的布局就不会影响到子元素
            3. (了解)给父元素加 上内边距1px
            4. (了解)给父元素加 上边框

实现：在元素中文字是一行显示的，如果有溢出的话，溢出部分隐藏，而且最后用省略号代替。

* + .box1{
                width: 200px;
                height: 100px;
                background-color: antiquewhite;
                /\* 设置以一行去显示文字，也就是不换行 \*/
                white-space: nowrap;
                /\* 溢出部分隐藏 \*/
                overflow: hidden;
                /\* 文字溢出后最后加省略号 \*/
                text-overflow: ellipsis;
            }

### 1. 高度塌陷的问题
\* 块元素默认宽度是父元素的100%，高度默认是由内容撑起来的。
\* 当我们没有给父元素设置高度时，如果它里面的所有子元素都浮动的话，父元素的高度就塌陷了，也就是变为0了
\* 高度塌陷的话，会对页面的布局造成影响。父元素后面的元素就会上移！！！

* + \* 第一种：给父元素设置高度即可。这样无论子元素是否浮动，父元素都是固定的高度。但是不是很推荐使用，因为有时我们也无法确定高度应该是多少！
    \* 第二种：给父元素开启 BFC 格式，即使用 overflow:hidden; 这样父元素就是独立的一块，不会受到子元素的影响。而且它会自动找到一个合适的高度！
    \* 第三种：给受影响的元素加 clear:left|right|both; 属性，表示清除浮动对它的影响。left-表示清除左浮动的影响； right-表示清除右浮动的影响； both-表示清除左和右浮动的影响。
      这种能解决父元素高度塌陷后对后面元素布局的影响，但是父元素的高度还是0！
    \* 第四种：给父元素中的最后加一个空 div。给空 div 设置 clear:both; 这样父元素中其它元素都浮动后，空 div也会找到自己合适的位置。就不会造成父元素高度塌陷，从而不会影响到后面元素的布局。
      不推荐使用。因为这样页面中可能会加很多的空 div 标签。
    简单来说，以后如果知道父元素高度，那就用第一种，不知道父元素高度那就用第二种。
    第五种:
    利用今天学习的伪元素选择器的方式：
    父元素::after{
        content:"";
        display:block;
        clear:both;
    }
    利用这种方式的话挺好，而且我们往往将这个样式代码写在`全局样式`中(将父元素那里替换为 .clear)，然后实际应用中哪个父元素要清除浮动的影响，就直接给那个父元素的 class 加个属
  + BFC：就是块级格式上下文，元素开启了BFC后，它就是独立的一块了，它的布局不会受到子元素的影响。
            开启 BFC的方式有：
            1. overflow:hidden;
            2. float:left;
            3. position:absolute;

盒子模型

标准盒模型

* + 1) 内容区(content)：盒子里面实际能放东西的大小
    - 内容区(content)
              1. 内容区就是一个盒子可以放多大的内容
              2. 我们平常给元素设置宽度(width)和高度(height)，就是设置给了内容区
              3. 单位：可以是 px，也可以是 百分比。百分比就是相对于父元素来说的。
              4. 块元素的特点，独占一整行，宽度默认是父元素的100%，高度默认由内容撑起来。在页面垂直摆放。
              5. 行内元素的特点，不会独占一整行，宽度和高度都是由内容撑起来的，内容有多大就占多大地，在页面水平摆放
                  之前我们说img标签是行内元素，但它却是可以设置宽高的，是因为，img标签严格来说是一种 行内块元素。
              6. 块元素可以设置宽度和高度；行内元素不能设置宽度和高度，即使设置了也不生效。
              关于宽高还常见的有：
              min-width
              min-height
  + 3) 边框(border)
    - 2. 边框由3部分样式组成：
      * 1) 边框宽度 border-width 默认3px
                        border-width:数值; 4个边是一样的宽度
                        border-width:数值1 数值2; 上下 左右
                        border-width:数值1 数值2 数值3; 上 左右 下
                        border-width:数值1 数值2 数值3 数值4; 上 右 下 左 顺时针
                        注意这个规律，后面学习 边框颜色 边框样式 内边距 外边距 都是这个规律。
                        也可以单独给某个方向的边框设置：border-xxx-width，xxx 就是 top bottom left right
      * 2) 边框颜色 border-color 默认黑色
                        和上面的边框宽度规律一样。一个值、二个值、三个值、四个值。也可以单独设置某个方向边框颜色
      * 3) 边框样式 border-style 实线、虚线、点线、双实线等等
                        边框的样式是必写的一个属性。可选值有：
                        solid 实线 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
                        dotted 点线
                        dashed 虚线
                        double 双实现
    - 1. 一个盒子有4个边框，border-top、border-bottom、boder-left、border-right
              3. 边框的简写：
                  1) 单方向的： \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
                      border-top-width border-top-color border-top-style
                      可以简写为：border-top:color style width; 属性值不区分前后顺序
                              border-bottom、border-left、border-right
                  2) 4个方向： \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
                      简写为 border:style color width; 属性值不区分前后顺序
                          比如：border:1px solid blue;
              边框这部分最重要的是记住：边框由三部分组成，边框的简写(单方向的简写、4个方向的简写)
  + 2) 内边距(padding)：内容区与边框之间的距离，也就是那个泡沫
    - 内边距
              1. 内边距(padding)：指的就是内容区与边框之间的距离。（也就是泡沫那部分）
              2. 内边距也有4个，padding-top、padding-bottom、padding-left、padding-right
              3. 内边距也可以简写为：
                  padding:数值; 4个方向的内边距一样
                  padding:数值1 数值2; 上下 左右
                  padding:数值1 数值2 数值3; 上 左右 下
                  padding:数值1 数值2 数值3 数值4; 上 右 下 左
              设置了内边距后，背景颜色会漫出到内边距。
  + 4) 外边距(margin)：盒子与盒子之间的距离
    - 外边距
              1. 外边距(margin)：指的是盒子与盒子之间的距离
              2. 一个盒子有 top、bottom、left、right4个方向，外边距也有4个，
                  margin-top、margin-right、margin-bottom、margin-left
              3. 外边距可以简写为：margin: ;
                  margin: 数值; 4个外边距都一样
                  margin: 数值1 数值2; 上下 左右
                  margin: 数值1 数值2 数值3; 上 左右 下
                  margin: 数值1 数值2 数值3 数值4; 上 右 下 左
                  外边距的取值可以是正数、也可以是负数。
              在页面中元素都是靠上靠左摆放。所以，上外边距和左外边距能够决定当前元素的位置。
              右和下外边距决定的是别的盒子的位置。
    - 外边距(margin)可以实现元素自己水平居中。margin:0 auto;
              与 text-align 区分开，text-align 实现的是元素内的内容水平居中。
    - 1. 有两个垂直摆放的元素，上面的元素设置了下外边距a，下面的元素设置了上外边距b，两个之间的距离是 a 和 b 中较大的那个数。
    - 2. 有两个水平摆放的元素，左边的元素设置了右外边距a，右边的元素设置了左外边距b，两个之间的距离是 a + b。

怪异盒模型：

* + 1. 主要指的是 IE 浏览器
            2. 组成部分：内容区、内边距、边框、外边距
            3. 实际占据的空间大小：
                水平方向占据的大小：宽度 + 左右外边距
                垂直方向占据的大小：高度 + 上下外边距
            4. width 和 height：
                在怪异盒模型中，width 和 height 是设置给到盒子的边框。
            使用 box-sizing 属性可以实现标准盒模型与怪异盒模型之间的转换。
            box-sizing:content-box; 盒子模型就是标准盒模型，默认值
            box-sizing:border-box; 盒子模型就是怪异盒模型

浮动

文档流

* + \* 文档：一个 HTML 页面就是一个文档（document）。
    \* \*\*文档流：网页的最底层就是文档流。\*\*
    \* 之前，我们所说的块元素、行内元素的特点，其实严格来说应该是元素在文档流中的特点：
    \* 块元素在文档流中的特点：
      元素会独占一整行，在页面中会垂直排列，宽度默认是父元素宽度的100%，高度由内容撑起来
    \* 行内元素在文档流中的特点：
      元素不会独占一整行，默认的宽度和高度都是由内容撑起来的。 行内元素是不能设置宽度和高度的。

1. 设计浮动最初的目的是为了做文字环绕图片的。
        2. 元素在网页中默认是靠上靠左摆放，当一个元素浮动后，该元素就脱离了文档流，
            元素后面的内容就会向上移动。
        3. 元素浮动后会脱离文档流，该元素就变为了 行内块元素。
        4. 行内块元素：就是既拥有块元素的可以设置宽高的特点，又有行内元素不独占一整行的特点。
        5. 元素浮动后的其它特点：
            1) 子元素浮动后，不会超出它的父元素
            2) 元素浮动后，遇到另一个浮动元素也会停下来
            3) 元素浮动后，遇到一个不浮动的块元素会停下来

定位

1. static，默认值，静态定位，也就是没有定位。

2. relative，相对定位

* + 相对定位（relative）
            1. 相对定位就是 position:relative;
            2. 相对定位就是相对于它原来的位置进行定位
            3. 相对定位配合使用 top、bottom、left、right 进行移动，如果上下左右都设置了的话，上 和 左 优先。
            4. 相对定位的特点：
                1) 相对定位后原来的位置会保留，也就是不会脱离文档流，但是元素的层级会提升。
            5. 相对定位的应用场景：
                1) 小范围内的移动
                2) 相对定位往往配合绝对定位使用
    - .box1{
                  width: 200px;
                  height: 200px;
                  background-color: red;
                  /\* 开启相对定位，相对于原来的位置 \*/
                  position: relative;
                  /\* top: 50px;
                  left: 50px; \*/
                  /\* top: -50px;
                  left: -50px; \*/
                  /\* right: 50px;
                  bottom: 50px; \*/
                  left: 200px;
                  right: 20px;
              }

3. fixed，固定定位

* + 固定定位
            1. 固定定位： postion:fixed;
            2. 固定定位是参照浏览器窗口去定位的
            3. 固定定位的特点：
                1) 使用固定定位后，元素原来的位置不保留，也就是元素会脱离文档流，层级会提升
                2) 固定定位的元素不会随着滚动条的滚动而滚动
            4. 应用场景：
                网站的广告、返回顶部。
    - .box1{
                  width: 100px;
                  height: 100px;
                  background-color: red;
                  /\* 元素开启固定定位，相对于浏览器窗口定位，会脱离文档流 \*/
                  position: fixed;
                  /\* top: 100px;
                  left: 100px; \*/
                  right: 50px;
                  bottom: 100px;
              }

4. sticky，粘性定位，CSS3新增

* + 粘性定位
            1. 粘性定位 position:sticky;
            2. 粘性定位利用到了相对定位和固定定位。会在这两个定位之间进行转换。
            3. 粘性定位默认是相对定位，当滚动条滚动的时候就成了固定定位。
            4. 垂直手机官网，导航条就用到了粘性定位。
            5. 粘性定位的要求：粘性定位元素的父元素的高度要大于粘性定位元素
    - .nav{
                  width: 100%;
                  height: 74px;
                  background-color: red;
                  /\*
                      开启粘性定位
                      默认是相对定位，相对于它原来的位置进行定位，所以下面设置了 0 0 没有变化
                      当滚动条滚动的时候，就变为了固定定位，相对于浏览器窗口定位， 0 0 就跑到了浏览器窗口左上角
                  \*/
                  position: sticky;
                  top: 0;
                  left: 0;
              }

5. absolute，绝对定位

* + 绝对定位
            1. 绝对定位： position:absolute;
            2. 绝对定位是参照最近的有定位(static不算有定位)的祖先元素去定位。如果所有的祖先元素都没有定位，
                那就相对于浏览器视口去定位。
            3. 一般，可以给body设置一个相对定位，如果所有的祖先元素都没定位的话，就相对于body进行定位。
            4. 绝对定位的特点：
                1) 元素开启绝对定位后，原位置不保留，脱离文档流，层级会提升。
            5. 一般某个元素开启绝对定位后，它的祖先元素要开启相对定位。
            6. 应用：
                我们看到页面中一个元素上又显示了另外一个元素，那就用绝对定位。
    - .box1{
                  width: 500px;
                  height: 500px;
                  background-color: red;
                  /\* 开启相对定位 \*/
                  position: relative;
              }
              .box2{
                  width: 100px;
                  height: 100px;
                  background-color: blue;
              }
              .box3{
                  width: 100px;
                  height: 100px;
                  background-color: orange;
                  /\* 开启绝对定位，会相对于最近的有定位的祖先元素定位，会脱离文档流 \*/
                  position: absolute;
                  top: 0;
                  left: 0;
              }

字体图标(iconfont)

[1. 有些东西，看着像图标，其实它是一个字体图标，可以像字体一样使用它
        2. 字体图标有专门的网站，我们去网站搜索下载即可。https://www.iconfont.cn/
        下载字体图标的步骤：
        1. 打开https://www.iconfont.cn/网址
        2. 点击 图标库--官方图标库
        3. 在搜索框搜索我们想要的图标
        4. 搜索到我们想要的图标后，加入购物车（前提是你登录一下）
        5. 点击购物车 -- 点击下载代码，就将图标的代码都下载了
        6. 下载完的代码文件，是在一个压缩包中
        字体图标的使用：
        1. 将下载好的压缩包解压
        2. 将解压后的各种文件，比如 .css .js .json 全局引入到项目中的css文件夹中
        3. 在要使用字体图标的页面中引入 iconfont.css文件
        4. 写一个 span 标签，class 必须是 iconfont，也可以有class名字
        5. span标签中的内容写的就是： 打开 demo\_index.html，找你想要的字体图标的 Unicode编码
        6. 要调整图标的大小，就用自己的class去调整字体大小](https://www.iconfont.cn/)

样式属性属性

颜色

* + rgb 表示颜色
    - 1) rgb 指的就是红色、绿色、蓝色，它们是三原色，通过它们的浓度的不同就可以调配出所有的颜色
                  2) 语法格式：rgb(红色的浓度,绿色的浓度,蓝色的浓度)
                  3) 浓度就是 0-255 之间的数，0表示最小，255最大
                  4) 浓度还可以使用百分比来表示，0%-0，100%-255，其实百分比还是最终转为 0-255 之间的数
    - 可以使用 十六进制的 rgb 表示颜色 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
                  1) 十六进制的rgb其实和上面第二种的原理一样，只不过是将颜色的浓度用 16 进制的数表示了
                  2) 十六进制的数有：0 1 2 3 4 5 6 7 8 9 a b c d e f
                  3) 之前 rbg 浓度是 0-255，现在换为16进制后，浓度就是 00-ff
                  4) 语法格式：#红色绿色蓝色，每种颜色都得用两位的16进制表示
                  5) 比如：#25aa2b、#aabbcc.
                  6) 像这种 #aabbcc 可以简写为 #abc，#112266 可以简写为 #126，但是 #112235 不能简写
                  7) 白色：#ffffff 简写 #fff，黑色 #000000 简写为 #000
  + rgba 表示颜色及透明度
    - 1) 和 rbg 表示法差不多，就是多了个透明度
                  2) 语法格式：rgba(红色的浓度,绿色的浓度,蓝色的浓度,透明度)
                  3) 透明度：
                      0-完全透明
                      1-完全不透明
                      0.6 可以简写 .6

字体属性

* + 1. color 属性：可以设置字体的颜色。
  + 2. font-size 属性：可以设置字体的大小。浏览器默认字体是16px，支持的最小字体是12px。
  + 3. font-weight 属性：可以设置字体的粗细。可选值有：
                1) normal 正常粗细
                2) bold 加粗
                3) bolder 更粗
                4) light 细的
                5) lighter 更细
                6) 100、200、300、400、500、600、700、800、900
                    400就相当于normal，也就是正常粗细 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
                    700就相当于bold，也就是加粗 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
  + 4. font-style 属性：可以设置字体倾斜。可选值有：
                1) normal，默认，正常字体
                2) italic 倾斜
  + 5. font-family属性：设置字体。可以设置多个字体，设置多个字体的话，用逗号隔开。
                比如： font-family: 宋体,微软雅黑; 如果用户电脑上支持第一个字体，那就用第一个，如果不支持的话，就尝试第二个

文本属性
1. 文本属性就是可以对元素内的内容做一些设置。

* + 1) text-align：设置元素内的内容的对齐方式，可选值有：
                    left-左对齐、center-水平居中、right-右对齐
                    这里的内容指的是：文字 和 行内元素。
  + 2) text-decoration：文本装饰的意思，可以对内容设置下划线、删除线、去掉下划线等等，可选值有：
                    underline：下划线
                    overline：上划线
                    none：正常文本
                    line-through：删除线
  + 3. text-transform 文本转换，可选值：
                1) none 默认，不做任何改动
                2) capitalize 将单词的首字母大写
                3) uppercase 将单词转为全大写
                4) lowercase 将单词转为全小写
  + 4. text-indent 文本缩进，可以设置元素中内容首行缩进多少
                取值可以是数字、也可以是百分比。 可以是正数，也可以是负数。正数往右缩进，负数往左
  + 单位：em。
            em 其实是一个百分比的单位，它是相对于当前字体大小来说的
            1em = 1 个 font-size
    - .box2{
                  font-size: 30px;
                  /\* 设置首行文本缩进的距离 \*/
                  text-indent: 2em;
              }

背景属性
背景属性就是给元素设置背景相关的样式的。

* + 1) background-color：背景颜色
  + 2) background-image：背景图片
                    背景图片默认是从元素的左上角开始显示，
                    如果背景图片比元素小，那么背景图片默认会水平和垂直方向都平铺
                    如果背景图片比元素大，那么背景图片会将整个元素覆盖，背景图片会显示不全
    - background-image: url(./images/6.webp);
    - 一定要将背景图片和图片标签给区分开！！！
  + 3) background-repeat：背景图片的平铺方式
                    可选值：
                    repeat 默认值，就是水平和垂直方向都平铺
                    repeat-x 只在水平方向平铺
                    repeat-y 只在垂直方向平铺
                    no-repeat 不平铺 \*\*\*\*\*\*\*\*\*\*\*\*\*\*
  + 4) background-size：背景图片的大小
                    background-size:数值1 数值2;
                    数值1表示背景图片的宽度，数值2表示背景图片的高度
                    如果只设置一个值，会认为是设置了背景图片的宽度，高度会等比例缩放
                    取值也可以是百分数，是相对要设置背景的那个元素的百分比。
                    取值也可以是 cover，背景图片会等比例放大，直到覆盖了整个元素
                    取值也可以是 contain，背景图片会等比例放大，直到某个边碰到元素，就停止了
  + 5) background-position：背景图片的位置
                    background-position: 数值1 数值2;
                    数值1表示水平方向移动的距离；数值2表示垂直方向移动的距离。 \*\*\*\*\*\*\*\*\*\*
                    取值如果是正数，那就向右下移动 \*\*\*\*\*\*\*\*
                    取值如果是负数，那就向左上移动 \*\*\*\*\*\*\*\*
                    如果只写一个数的话，表示水平方向的移动距离，垂直方向会垂直居中
                    取值也可以是百分比，是相对于元素的百分比。元素的左上角是 0%,0% 元素的右下角是100%，100%
                    取值还可以是 top、right、bottom、left、center \*\*\*\*\*\*\*\*\*\*\*\*\*
  + 6) background-attachment：背景图片是否随着滚动条的滚动而滚动
    该属性设置背景图像是否固定或者随页面滚动。简单来说就是一个页面有滚动条的话，滑动滚动条背景是固定的还是随页面滑动的。
    \* scroll 背景图片随着页面的滚动而滚动，这是默认的。
    \* fixed 背景图片不会随着页面的滚动而滚动。
  + 7) background：背景属性的简写方式
    这些简写属性的值不区分前后顺序

其他属性

* + letter-spacing：字符间距，该属性可以设置元素内字符间距的大小。
            正值的话，字符间距变大；负值的话，字符间距变小。
  + line-height
            1. line-height：就是行高的意思。
            2. 初中写日记的本中有两条线，我们就是在两条线之间写字。这两条线之间的距离就称为行高。
            3. 在浏览器中，文字也是两条看不见的线之间呢，而且文字会在这两条线之间垂直居中。
            4. 文字会在行高中垂直居中
            5. 行高的取值：可以是 px，也可以直接是数字，比如 1 2 1.5，如果是数字的话，行高=数字\*当前字体大小
            行高最重要的应用就是可以用来设置文字在容器中垂直居中。其实就是把行高设置为容器的高度。

overflow属性
溢出的意思，可以对元素中溢出的那部分做设置。

* + 可选值有：
                1) visible：默认值，不对溢出部分做任何处理
                2) hidden：溢出部分隐藏
                3) scroll：不论是否有溢出，都有滚动条
                4) auto：有溢出，就有滚动条，没溢出就没有滚动条
  + 对 overflow:hidden; 说明：
            overflow:hidden; 除了有 将溢出部分隐藏 的作用外，
            还有一个更重要的作用：就是对元素开启 BFC 格式，blocking format context，也就是块级格式上下文的意思。
            也就是 元素开启BFC后，这个元素就是一个独立的元素，它的布局不受子元素的影响。
  + opacity 透明度
            1. 我们可以通过 opacity 属性给元素设置透明度。
            2. 将 opacity 和 rgba() 区分开。rgba() 设置颜色的透明度，而 opacity 是设置元素的透明度
            3. 透明度的取值：
                0 - 完全透明
                1 - 不透明
                0.5 - .5
  + cursor 光标
            1. cursor 可以设置鼠标光标的形状
            2. 常见的就是将鼠标光标设置为 小手
            3. 将鼠标光标变为小手 cursor:pointer;
  + vertical-align
            1. vertical-align：是垂直对齐方式的意思
            2. 学习 html中table标签中的tr里面td时，td 标签有 valign 属性
            3. 我们可以用vertical-align属性对单元格中的内容做垂直对齐，可选值：top-上、middle-中、bottom-下
            4. 它还可以设置给图片，用来设置文字和图片的对齐方式

display 属性

* + 1. display 是 显示的意思。
            2. 我们可以通过 display属性实现元素类型之间的转换。比如将行内元素转为块元素，将块元素转为行内元素或者
                将 行内元素转为 行内块元素等等。
            3. display属性的可选值：
                block 将元素转为块元素
                inline 将元素转为行内元素
                inline-block 将元素转为行内块元素（不是很推荐，因为行内块元素会识别元素间的空白）块元素转为行内块元素，但是会识别元素间的空白，最好用浮动
                none 将元素隐藏！ 注意 和 overflow:hidden;区分

列表属性

* + 1. list-style-type(了解)：可以设置列表的项目符号类型，取值：disc-实心圆、circle-空心圆、A、a
            2. list-style-image(了解)：可以设置列表的项目符号为一张图片
            3. list-style-position(了解)：可以设置列表项目符号的位置，取值可以是：inside、outside
  + 列表样式属性都是 list-style-xxx 格式，所以可以简写为：list-style: type image position;
            最重要的是掌握： list-style:none; 不要列表项目符号

选择器
选择器的优先级：
        比如：使用内部样式给某个元素设置样式时，
        我们分别使用了 全局选择器、标签选择器、类选择器、id选择器都给同一个元素设置了样式，
        那谁生效？
        基本选择器优先级：
        行内样式 > id选择器 > 类选择器 > 标签选择器 > 全局选择器
        选择器的权重（了解）
        行内样式 1000
        id选择器 100
        类选择器 10
        标签选择器 1
        全局选择器 0

1. 全局选择器(通配符选择器)

* + 全局选择器：
            1. 可以选中页面中的所有的元素
            2. 语法格式：
                \*{
                    color:red;
                }

2. 标签选择器(元素选择器)

* + 标签选择器：
            1. 就是选中指定的元素
            2. 语法格式：
                标签{
                    属性:属性值;
                }

3. 类选择器(class选择器)

* + 类选择器：
            1. 就是通过标签的class属性去选中元素，然后给元素设置样式
            2. 语法格式：
                .class的值{
                    属性:属性值;
                }
            class属性的说明：
            1. class属性是每个html标签都有的一个属性
            2. class属性值是可以重复的，也就是说页面中可以有两个标签的class的值一样
            3. 一个标签的class属性值可以有多个，用空格隔开

4. id选择器

* + 关于id属性的说明：
            1. 每个标签都有id属性
            2. 在一个页面中标签的id属性值是不能重复的
            id选择器：
            1. id选择器就是根据标签的id属性值去选择元素，然后给它设置样式
            2. 语法格式：
                #id值{
                    属性:属性值;
                }

并集选择器：
        1. 并集选择器就是可以一次选中多个其它选择器选中的元素。
        2. 语法：
            选择器1,选择器2,...{
                属性:属性值;
            }

交集选择器：
        1. 就是选择符合多个条件的元素
        2. 语法格式：
            选择器1选择器2...选择器n{
            }

* + 选中div元素，而且div的class="box" 里面的子元素a \*/
            div.box>a{
                color: red;
            }
            /\* 选中div元素里面的所有的class="box"的元素里面的子元素a \*/
            div .box>a{
                color: yellow;
            }

关系选择器

* + 子元素选择器
            1. 选择父元素中指定的子元素
            2. 语法格式：
                父元素>子元素{
                    属性:属性值;
                }
  + 后代元素选择器：
            1. 后代元素选择器就是用来选择祖先元素的后代元素。
            2. 语法格式：
                祖先元素 后代元素{
                    属性:属性值;
                }
  + 相邻兄弟选择器
            1. 相邻兄弟选择器就是选中某个元素后面紧挨着它的兄弟
            2. 语法格式：
                前一个兄弟+后一个兄弟{
                    属性:属性值;
                }
    - 用得少
  + 通用兄弟选择器
            1. 通用兄弟选择器就是选择某个元素后面的指定的所有的兄弟。
            2. 语法格式：
                A~B{
                    属性:属性值;
                }
    - 用得少

伪类选择器

* + 伪类选择器
            1. 伪类选择器表示的是一种状态，比如：未访问的超链接、已访问的超链接、鼠标悬浮、鼠标点击等等
            2. 语法格式： :xxx
            3. 伪类选择器有：
                1) :link 表示未访问超链接 只能用于超链接
                2) :visited 表示访问过的超链接 只能用于超链接
                3) :hover 表示鼠标悬浮时 能用于所有的元素 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
                4) :active 表示鼠标按下去时 能用于所有的元素
            4. 强调：这4个伪类选择器用在超链接上时，顺序是：:link :visited :hover :active
    - /\* 未访问过的超链接 \*/
              a:link{
                  background-color: red;
              }
              /\* 已访问过的超链接 \*/
              a:visited{
                  background-color: blue;
              }
              /\* 鼠标悬浮时 \*/
              a:hover{
                  background-color: green;
              }
              /\* 鼠标点击时 \*/
              a:active{
                  background-color: yellow;
              }
              .box1{
                  width: 200px;
                  height: 200px;
                  background-color: skyblue;
              }
              /\* 选中class="box1" 的元素，而且鼠标要悬浮在其上面 \*/
              .box1:hover{
                  background-color: red;
              }
              /\* 选中 class="box1" 的元素，而且鼠标点击下去 \*/
              .box1:active{
                  background-color: orange;
              }
  + CSS3新增的伪类选择器: 这个是在所有儿子中
            1. :first-child 可以选中第一个子元素，是在所有的兄弟中排第一个
            2. :last-child 可以选中最后一个子元素，是在所有的兄弟中排最后一个
            3. :nth-child(值) 可以选中第n个子元素，它的取值可以是 数字、表达式(2n+3)、odd、even
    - /\* 选中页面中所有p标签，而且得是子元素,且是在所有兄弟中的排第一个子元素 \*/
              /\* p:first-child{
                  background-color: aqua;
              } \*/
              /\* 选中p标签，而且得是最后一个子元素 \*/
              /\* p:last-child{
                  background-color: palegreen;
              } \*/
              /\* 选中p标签，而且得是第4个子元素 \*/
              /\* p:nth-child(4){
                  background-color: palevioletred;
              }
  + 这个实在同类型的儿子中
            4. :first-of-type 可以选中第一个子元素，是在同类型子元素中的第一个        5. :last-of-type 可以选中最后一个子元素，是在同类型子元素中的最后一个
            6. :nth-of-type 可以选中第n个子元素，是在同类型子元素中的第n个
    - /\* 选中p标签，而且得是第一个子元素 \*/
              /\* p:first-child{
                  background-color: aqua;
              } \*/
              /\* 选中p标签，而且得是最后一个子元素 \*/
              /\* p:last-child{
                  background-color: palegreen;
              } \*/
              /\* 选中p标签，而且得是第4个子元素 \*/
              /\* p:nth-child(4){
                  background-color: palevioletred;
              }
              p:nth-child(odd){
                  background-color: red;
              }
              p:nth-child(3n){
                  background-color: orange;
              } \*
  + 7. :not() 否定选择器,:not() 否定选择器。比如，.box>h4:not(.two) 意思是选择 class="box" 里面的 h4 元素，但是不包括 class="two" 的h4 元素。          8. :focus() 鼠标聚焦
            9. :checked 选中的单选按钮、复选中
    - p:not(.pp){
                  background-color: red;
              }
              .input1:focus{
                  background-color: red;
              }
              .input2:checked{
                  width: 50px;
                  height: 50px;
              }

伪元素选择器
1. 伪元素选择器就是通过 css 的技术给页面中加一个元素。这个元素不是html真正的标签。所以称为伪元素。
        2. 伪元素有两种，它可以在某个元素内的最前面或者某个元素内的最后面加元素。

* + 语法：
                1) 给某个元素内的最后面加伪元素(也可以理解为给某个元素内最后加个子元素)
                    :after{
                        content:"";
                    }
                    或者
                    ::after{
                        content:"";
                    }
                2) 给某个元素内的最前面加伪元素(也可以理解为给某个元素内最前面加个子元素)
                    :before{
                        content:"";
                    }
                    或者
                    ::before{
                        content:"";
                    }
  + 4. 应用：
                1) 利用伪元素加文字
                2) 利用伪元素加图片
                3) 利用伪元素自定义新增内容(掌握)

写法
写css样式有三种写法：
        1. 行内样式
        2. 内部样式
        3. 外部样式
        如果我们使用这三种写法都对同一个元素设置了样式，那到底谁生效呢？？？
        这就是优先级的问题。
        优先级：
        行内样式的优先级最高
        内部样式和外部样式的优先级一样，谁写在后面谁生效！！！

行内样式：
        行内样式就是将CSS样式直接写在 html 的开始标签中。
        语法：
            属性:属性值;
        特点：
            1. 写法简单
            2. 不能复用
            3. css代码和html代码混合在一块了，耦合性太强
            不推荐使用
        常见的css样式（属性）：
        宽度 width 注意在css中单位是不能省略的！
        高度 height
        背景颜色 background-color
        块元素的特点：
        独占一整行，宽度默认是父元素的 100%，高度默认由内容撑起来，在页面垂直摆放！

内部样式：
        1. 内部样式是将css样式代码写在head标签的style标签里面。
        2. 语法格式：
            选择器{
                属性:属性值;
                属性:属性值;
                ...
            }
            选择器目前还没有学习，先模仿着用。
        3. 特点：
            1) html代码和css代码耦合性降低了。但是并不是彻底没有耦合性，因为css还是写在了html文件中。
            2) 可以达到样式在当前页面的重用
            是一种比较推荐的方式，但还不是我们工作中要写的！

* + html中的每个标签都可以有 class 属性，而且class值是可以重复

外部样式：
        1. 外部样式：将CSS样式代码写在了外部的 .css 文件中
            然后，哪个 html文件要用该样式，就引入 css 文件。
        2. 语法格式：
            选择器{
                属性:属性值;
                属性:属性值;
                ...
            }
        3. 外部样式特点：
            1) 可以达到在整个项目中样式复用
            2) html 代码 和 css 代码彻底分开，没有耦合
            工作中我们就是使用这种写法。

* + <link rel="stylesheet" href="./css/09.css">
  + link标签可以引入外部的css文件。它还有别的作用：
            可以引入一张图片，作为网页的图标。
    即.ico文件可作为网页标签图标
    - <link rel="icon" href="./images/2.ico">