

HTML文档结构

两种打开HTML文件的方式

标签的分类1

head内常用标签

body内常用标签

基本标签

标签分类2

特殊符号

常用标签

img标签

a标签

标签具有的两个重要书写

列表标签

表格标签

表单标签

验证form表单提交数据

CSS

CSS选择器

基本选择器

组合选择器

属性选择器

分组与嵌套

伪类选择器

伪元素选择器

选择器优先级

CSS属性相关

设置长宽

字体属性

文字属性

背景图片

边框

display属性

盒子模型

浮动

浮动带来的影响

溢出属性

定位

验证浮动和定位是否脱离文档流（原来的位置是否还保留）

z-index模态框

透明度opacity

HTML就是书写网页的一套标准

```
1 <!--单行注释-->
2 <!--
3 多行注释1
4 多行注释2
5 多行注释3
6 -->
```

```
7  由于HTML代码非常的杂乱无章并且很多，所以我们习惯性的用注释来划定区域，方便后续查找
8  <!--导航条开始-->
9  导航条所有的html代码
10 <!--导航条结束-->
11 <!--左侧菜单栏开始-->
12 左侧菜单栏的html代码
13 <!--左侧菜单栏结束-->
14
15
```

HTML文档结构

```
1  <html>
2      <head></head> :head内的标签不是给用户看的，而是定义一些配置，主要是给浏览器看到
3      <body></body> :body内的标签 写什么浏览器就渲染什么，用户就能看到什么
4  </html>
```

ps:文件的后缀名其实是给用户看到，只不过对应不同的文件，文件后缀名有不同的软件来处理添加功能

注意:HTML代码是没有格式的，可以全部写在一行，只不过习惯了缩进来表示代码

两种打开HTML文件的方式

- 找到文件所在的位置右键选择浏览器打开
- 在pycharm内部集成了自动调用浏览器的功能，直接点击即可（前提是安装了对应的浏览器）

标签的分类1

1. 双标签
2. 单标签（自闭和标签）

head内常用标签

```
1  <title>标题</title> 显示网页标题
2  <style></style> 内部用来书写css代码\
3  <script>
4      alert(123)
5  </script> 内部用来书写js代码
6  <script src="my.js"></script> 还可以引入外部js文件
7  <link rel="stylesheet" href="my.css"> 引入外部css文件
8
9
10 <meta name="keywords" content="python全站,Linux运维培训,Python自动化,GO语言
    培训"> 当你在用浏览器搜索的时候，只要输入了keywords后面指定的关键字，那么该网页都有可能
    被百度搜索出来展示给用户（网页推荐）
11
12 <meta name="description" content="淘宝网-亚洲最大的网上交易平台，提供各类服装、
    美容、家具、数码、话费、数亿优质商品，等安全交易保障服务"> 网页的描述信息
```

body内常用标签

肉眼能够在浏览器上看到的花里胡哨的页面，内部都是HTML代码

基本标签

```
1 <h1>标题h1</h1> 标题标签1-6级标题
2 <b>加粗</b>
3 <i>斜体</i>
4 <u>下划线</u>
5 <em>强斜体</em>
6 <strong>强加粗</strong>
7 <s>删除线</s>
8 <p>虽然他送了我玫瑰花</p>
9 <br>换行
10 <hr> 水平分割线
11 <p>段落</p>
```

标签分类2

```
1 # 1. 块儿标签：独占一行
2   h1-h6 p div
3   1. 块级标签可以修改长宽，行内标签不可以，修改了也不会变化
4   2. 块级标签内部可以嵌套任意的块级标签和行内标签，
5       但是p标签虽然是块级标签，但是他里面只可以嵌套行内标签，不能嵌套块级
   如果套了，问问题也不大，浏览器会自动帮你解开（浏览器会面向用户
   的，不会轻易的报错，
   哪怕有报错，用户也基本感觉不出来）
6 # 2. 行内标签：自身文本多大就占多大
7   i u s b strong em span
8   行内标签不能嵌套块级标签
```

特殊符号

```
1 &nbsp; 空格
2 &gt; 大于号
3 &lt; 小于号
4 &  &
5 &yen;  ¥
6 &copy;  ©
7 &reg;  ®
```

常用标签

```
1 div  块儿级标签
2 span 行内标签
3
4 构造页面初期最常使用的,页面布局一般先用div和span占位之后再调整样式，尤其是
5 div使用非常的频繁
6 div可以看成一块区域,也就意味着用div来提前规定所有的区域，之后往该区域的内部填写内容即可
7 普通的文本先用span标签
8
```

img标签

```
1  #图片标签
2  <img src="" alt="" title="">
3
4  src:
5      1. 图片路径: 可以是本地的也可以网上的
6      2. url: 自动向该url发送get请求获取数据
7
8  alt:
9      当图片加载不出来的时候, 给图片的描述信息
10 title:
11     当鼠标悬浮到图片上显示的提示信息
12
13 height:
14 width:
15     当只修改一个的时候, 另外一个参数会等比例的缩放
16     如果修改了两个参数, 并且没有考虑比例的问题, 图片就会失真
17
```

a标签

```
1  #链接标签
2  <a href="https://www.baidu.com" target="_blank">百度知道</a>
3
4  ""
5  当a标签指定的网址从来没有被点击过, 那么a标签的字体颜色是蓝色
6  如果点击过了, 就会是紫色, (浏览器给记忆了)
7  ""
8
9  href:
10     放的是url, 用户点击, 就会跳转到url页面
11     可以放其他a标签的id 值, 点击即可跳转到对应的标签位置
12
13 target
14     默认是_self, 在当前页面跳转
15     新建页面跳转 _blank
16 # a标签的锚点功能
17 "" 点击一个文本标题, 页面自动跳转到哦标题对应的内容区域""
18 <a href="" id="d1">顶部Top</a>
19 <h1 id="d22">11</h1>
20 <div style="height: 1000px;background-color: tomato"></div>
21 <a href="" id="d2">中间Middle</a>
22 <div style="height: 1000px;background-color: blue"></div>
23 <a href="#d1">回到顶部</a>
24 <a href="#d2">回到中间</a>
25 <a href="#d22">回到中间</a>
26
```

标签具有的两个重要书写

1. **id**值
类似于标签的身份证号，在同一个html页面上id值不能重复
2. **class**值
该值有点类似于面向对象里面的继承，一个标签可以继承多个class值

标签可以有自带的属性，也可以有自带的属性

```
1 <p id="d1111" class="c11" username="zhao" password="123"></p>
```

列表标签

- 无序列表

```
1 <ul>
2   <li>第一</li>
3   <li>第二</li>
4   <li>第三</li>
5 </ul>
6 虽然ul很丑，但是在页面布局的时候，只要是排版一致的几行数据基本上使用的都是ul标签
```

- 有序列表

```
1 <ol type="1" start="5">
2   <li></li>
3   <li></li>
4   <li></li>
5 </ol>
6
7 <!--1 A I a....-->
```

- 标题列表

```
1 <dl>
2   <dt>标题一</dt>
3   <dd>内容一</dd>
4   <dt>标题二</dt>
5   <dd>内容二</dd>
6   <dt>标题三</dt>
7   <dd>内容三</dd>
8
9 </dl>
```

表格标签

```
1 <table>
2   <thead>
3     <tr>  一个tr就是一行
4       <th></th>  加粗文本
5       <td></td>  正常文本
```

```

6         </tr>
7     </thead> 表头（字段信息）
8     <tbody>
9
10    </tbody> 表体（数据信息）
11 </table>
12
13 <td border="1" >zha</td> #加外边框
14 <td colspan="2" >zha</td> #水平方向占多行
15 <td rowspan="2" >zha</td> #垂直方向占多行

```

表单标签

```

1  """
2  获取用户数据，（用户输入的内容，用户选择，用户上传...）基于网络发送给后端服务器
3  """
4
5  <form action=""></form> 在该form 标签内部书写的获取用户的数据都会被form标签提交到后端
6
7
8  action:控制数据提交的后端路径(给哪个服务端提交数据)
9      1.什么都不写，默认朝当前页面所在的url提交数据
10     2.写全路径:https://www.baidu.com 朝百度服务端提交
11     3.只写路径后缀:action='/index/' 自动识别当前服务端的IP和Port拼接到前面
12     host:port/index/
13
14  <label for="d1"> 第一种: 直接将input框写在label内
15     username:<input type="text" id="d1">
16 </label>
17 <label for="d2">password:</label> 第二种: 通过id链接即可，无需嵌套
18 <input type="password" id="d2">
19 PS:input不跟label关联也没有问题
20
21 """
22 label和input都是行内标签
23
24
25 """input标签类似于前端的变形金刚，通过type属性变形"""
26 text:普通文本
27 password:密文
28 date :日期
29 submit:用来触发form表单提交数据的动作
30 button:一个普普通通的按钮，本身没有任何功能，学玩js后可以给它绑定自定义各种功能
31 reset:重置内容
32 radio:单选
33     默认选中要加checked="checked"
34     <input type="radio" name="gender" checked="checked">男
35     当标签的属性名和属性值一样的时候可以简写
36     <input type="radio" name="gender" checked>女
37 checkbox:多选
38     <input type="checkbox" checked>JayChou
39 file:获取文件,也可以一次性获取多个,加个multiple
40     <input type="file" mutiple>
41 hidden:隐藏当前input框

```

```

42         <input type="hidden">
43
44
45
46 ""select标签,默认是单选,可以加multiple变多选,默认选中用selected""
47 <p>周杰伦
48     <select name="" id="" multiple>
49         <option value="">可爱女人</option>
50         <option value="" selected>霍元甲</option>
51         <option value="" selected>晴天</option>
52         <option value="">稻香</option>
53         <option value="">退后</option>
54         <option value="">安静</option>
55         <option value="">爱在西元前</option>
56     </select>
57 </p>
58
59 <p>带标题的省份:
60     <select name="" id="">
61         <optgroup label="上海">
62             <option value="">浦东</option>
63             <option value="">黄埔</option>
64         </optgroup>
65         <optgroup label="山西">
66             <option value="">运城</option>
67             <option value="">稷山</option>
68         </optgroup>
69         <optgroup label="北京">
70             <option value="">朝阳区</option>
71             <option value="">东城区</option>
72         </optgroup>
73     </select>
74 </p>
75
76 ""textarea标签 获取大段文本(文本域)""
77     <textarea name="" id="" cols="30" rows="10" maxlength="20">
78 </textarea>
79
80 # 能够触发form表单数据的按钮有哪些
81     1. <input type="submit" value="注册">
82     2. <button>点我</button>
83
84 #所有获取用户输入的标签都应该有name属性,
85 name类似于字典的key
86 用户的数据类似于字典的value
87
88 <p>性别:
89     <input type="radio" name="gender">男
90     <input type="radio" name="gender">女
91     <input type="radio" name="gender">其他
92 </p>

```

验证form表单提交数据

```
1 pip install Flask
```

```
1 #form表单默认的提交数据的方式是get请求，数据是直接放在url后面的
2 http://127.0.0.1:5000/index/?gender=on
3 #可以通过method
4 <form action="http://127.0.0.1:5000/index/" method="post">
5
6
7 针对用户选择的标签，用户不需要输入内容，但是你需要提前给这些标签加上内容values值
8     <p>性别：
9         <input type="radio" name="gender" value="male">男
10        <input type="radio" name="gender" checked value="female">女
11        <input type="radio" name="gender" value="others">其他
12    </p>
13    <p>爱好：
14        <input type="checkbox" checked name="hobby" value="Read">Read
15        <input type="checkbox" checked name="hobby" value="Movie">Movie
16        <input type="checkbox" name="hobby" value="JayChou">JayChou
17        <input type="checkbox" name="hobby" value="Python">Python
18    </p>
19
20    <p>省份
21        <select name="province" id="">
22            <option value="hb">河北</option>
23            <option value="sx" selected="selected">山西</option>
24            <option value="sz">深圳</option>
25            <option value="cq">重庆</option>
26        </select>
27    </p>
28
29
30    """
31    form表单提交文件需要注意
32        method必须是post
33        enctype="multipart/form-data"
34        enctype类似于数据提交的编码格式
35            默认是urlencoded 只能够提交普通的文本数据
36            formdata 可以支持提交文件数据
37    """
38
39    <form action="http://127.0.0.1:5000/index/" method="post"
40        enctype="multipart/form-data">
41
42    #针对用户输入的标签，如果加了value 那就是默认值
43    <p>
44        <label for="d1">
45            username:<input type="text" id="d1" name="username" value="默认值">
46        </label>
47    </p>
48
49    disables 禁用属性
```



```

50  readonly 只读
51
52  <label for="d1">
53      username:<input type="text" id="d1" name="username" value="默认值"
54      disabled>
55  </label>
56
57  <label for="d2">
58      username:<input type="text" id="d2" name="username" value="默认值"
59      readonly>
60  </label>

```

```

1  # -*- coding: UTF-8 -*-
2  # @Date : 2022/10/9 17:53
3  from flask import Flask,request
4
5  app = Flask(__name__)
6
7  @app.route('/index/', methods=['GET', 'POST']) # 当前url既可以支持get, 也可以支持post
8  def index():
9      print(request.form)#获取form表单提交过来的非文件数据
10     print(request.files)#获取文件数据
11     file_obj=request.files.get('myfile')
12     # print(file_obj.name)
13     file_obj.save(file_obj.name)
14     return 'OK'
15
16
17 app.run()
18

```

CSS

层叠样式表：就是给HTML标签添加样式的，让他变得更好看

```

1  # 注释
2  /*单行注释*/
3
4  /*
5  多行注释
6  多行注释
7  多行注释
8  */
9
10 通常在写CSS样式的时候也会用注释来划定样式区域（应为HTML代码多，所有对应的CSS代码也会很多）
11
12  /*这是博客园首页的CSS样式文件*/
13  ...
14  /*顶部导航条样式*/
15  ...
16  /*左侧菜单栏样式*/
17  ...
18  /*右侧菜单栏样式*/

```

```

19  ...
20
21  # CSS样式语法结构
22  选择器{
23      属性1:值1;
24      属性2:值2;
25  }
26
27  # CSS三种引入方式
28      1. style标签内部直接书写
29          <style>
30              h1{
31                  color: yellow;
32              }
33          </style>
34      2. link标签引入外部css文件(最正规的方式 解耦合)
35          <link rel="stylesheet" href="./my.css">
36      3. 行内样式(一般不用)
37          <h1 style="color: lime">老板好</h1>
38

```

```

1  ""
2  CSS学习流程
3      1. 先学如何查找标签
4          CSS查找标签的方式一定要会
5      2. 之后再学如何添加样式
6  ""

```

CSS选择器

基本选择器

```

1  # id选择器
2      #d1{
3          color: lawngreen;
4      }
5  # 类选择器
6      .c1{
7          color: red;
8      }
9  # 元素/标签选择器
10     span{
11         color: mintcream;
12     }
13  # 通用选择器
14     *{
15         color: darkgreen;
16     }

```

组合选择器

```
1  # 在前端中 将标签嵌套用亲戚关系来描述层级
2  <div>div
3      <p>div.p1</p>
4      <p>div.p2
5          <span>div.p2.span</span>
6      </p>
7      <span>div.span1</span>
8      <span>div.span2</span>
9  </div>
10
11  div里的p span都是div的后代
12  p是div的后代
13  p里面的span是p的儿子，是div的孙子
14  div是p的父亲
15
16  # 后代选择器
17      div span{
18          color: red;
19      }
20  # 儿子选择器
21      div>span{
22          color: red;
23      }
24  # 毗邻选择器
25      div+span{ /*同级别，紧挨着的第一个*/
26          color: lime;
27      }
28  # 弟弟选择器
29      div~span{ /*同级别所有的span*/
30          color: yellow;
31      }
```

属性选择器

```
1  ""
2  1. 含有某个属性
3  2. 含有某个属性并且有某个值
4  3. 含有某个属性并且有某个值的某个标签
5
6  ""
7  # 属性选择器是以[]来
8
9  [username] { background-color: red;} /*将所有含有属性名是username的标签背景色改为红色*/
10 [username='zhao']{color:orange;}
11 input[username='zhao']{background-color: yellow;}
```

分组与嵌套

```
1  div,span,p{    /*逗号表示并列关系*/
2      color: tomato;
3  }
4
5  #d1,.c1,span {
6      color: greenyellow;
7  }
8
9  #d1 .c2{
10     color: lime;
11 }
12
```

伪类选择器

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <style>
7          a:link { /*访问之前的状态*/
8              color: red;
9          }
10
11         a:hover {
12             color: blue; /*鼠标悬浮态*/
13         }
14
15         a:active {
16             color: black; /*鼠标点击不松开的状态  激活态*/
17         }
18
19         a:visited {
20             color: darkgreen; /*访问之后的状态*/
21         }
22
23         p {
24             color: lime;
25             font-size: 20px;
26         }
27
28         p:hover {
29             color: #5bc0de;
30         }
31
32         input:focus { /*input框获取焦点  （鼠标点击了input框）*/
33             background-color: mintcream;
34         }
35
36     </style>
37 </head>
```

```

38 <body>
39 <a href="https://blog.csdn.net/">拉钩网</a>
40 <p>点我</p>
41 <input type="text">
42 </body>
43 </html>

```

伪元素选择器

```

1  p:first-letter{
2      font-size: 48px;
3      color: orange;
4  }
5  p:before{
6      content: '..?';
7      color: blue;
8  }
9  p:after{
10     content: '雨露均沾';
11     color: tomato;
12 }
13
14
15 ps: before和after通常用来清除浮动带来的影响：父标签塌陷的问题

```

选择器优先级

```

1  ""
2  id选择器
3  类选择器
4  标签选择器
5  行内式
6  ""
7
8      1. 选择器相同，书写顺序不同
9          # 就近原则
10
11      2. 选择器相同，书写顺序不同
12          # 行内 > id选择器 > 类选择器 > 标签选择器
13
14      # !important （优先级最高）强制让标签采用你的样式

```

CSS属性相关

设置长宽

```

1  <style>
2      p {
3          background-color: red;
4          height: 200px;
5          width: 400px;
6      }
7
8      span {
9          background-color: green;
10         height: 200px;

```

```

11     width: 400px;
12     /*行内标签无法设置长宽，就算写了，也不生效*/
13 }
14 </style>
15
16
17 /*
18     块级标签可以改长宽
19     行内标签不可以改行宽
20     块级标签的宽度不修改的情况下，默认占浏览器的一整行，块级标签的高度也就是取决于标签内部
    的文本高度
21     行内标签宽度和高度都是由内部文本决定的，行内标签是无法设置长宽的
22
23 */

```

字体属性

```

1  p {
2      font-family: 'Arial Black', '微软雅黑'; /*第一个不生效，就用后面的，*/
3      font-size: 24px;
4      font-weight: bold; /*bold lighter 100-900 inherit*/
5      /*color: red; /*直接写英文!*/
6      /*color: #ffff00; /*十六进制，颜色编号!*/
7      /*color: rgb(128,23,45); /*三基色 范围都是0-255!*/
8      color: rgba(128,23,45,0.1); /*第四个参数是颜色的透明度 范围是0-1*/
9
10 }

```

文字属性

```

1  p {
2      /*text-align: center; /* center right left */
3      /*text-align: justify; /*两端对齐!*/
4      /*text-decoration: underline; /*下划线!*/
5      /*text-decoration: overline; /*上划线!*/
6      text-decoration: line-through; /*删除线*/
7      /*text-decoration: none;*/
8      /*在HTML中，有很多标签渲染出来的样式效果是一样的*/
9      font-size: 16px;
10     text-indent: 32px; /*缩进32px*/
11
12 }
13
14 a {
15     text-decoration: none; /* 用于给a标签去掉自带的 下划线*/
16 }
17
18
19 line-height 行高

```

背景图片

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <style>
7          #d1{
8              height: 500px;
9              background-color: red;
10         }
11         #d2{
12             height: 500px;
13             background-color: greenyellow;
14         }
15         #d3{
16             height: 500px;
17             background-image: url("img.png");
18             background-position:
19
20             background-attachment: fixed;
21         }
22         #d4{
23             height: 500px;
24             background-color: aqua;
25         }
26     </style>
27 </head>
28 <body>
29 <div id="d1"></div>
30 <div id="d2"></div>
31 <div id="d3"></div>
32 <div id="d4"></div>
33 </body>
34 </html>
35
36
37
38 /*background-repeat: repeat-x;!* x轴铺 *!*/
39 /*background-repeat: repeat-y;!* y轴铺*!*/
40 /*background-position: center center;*/
41
42 background: red url("./dog.jpeg") no-repeat;
```

边框

```
1  # 任何一个标签都有上下左右四个不同的边框
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <title>Title</title>
7      <style>
```

```

8      p {
9          /*border: 1px;*/
10         border-width: 6px;
11         border-style: solid;
12         border-color: rebeccapurple;
13         background-color: lime;
14
15
16     }
17
18     div {
19         /*border-left-width: 5px;*/
20         /*border-left-color: red;*/
21         /*border-left-style: solid;*/
22
23         /*border-right-width: 10px;*/
24         /*border-right-color: greenyellow;*/
25         /*border-right-style: dotted;*/
26
27         /*border-top-width: 15px;*/
28         /*border-top-color: deeppink;*/
29         /*border-top-style: solid;*/
30
31         /*border-bottom-width: 5px;*/
32         /*border-bottom-color: blue;*/
33         /*border-bottom-style: dashed;*/
34
35         border: 3px solid darkgreen;
36     }
37     #d1{
38         background-color: mediumpurple;
39         width: 400px;
40         height: 400px;
41         border-radius: 50%; /*圆，直接写50%，长宽一致就是圆*/
42     }
43     </style>
44 </head>
45 <body>
46 <p></p>
47 <div>python</div>
48 <div id="d1"></div>
49 </body>
50 </html>

```

display属性

```

1  # 能够让标签具有自身没有的属性和特征
2  <!DOCTYPE html>
3  <html lang="en">
4  <head>
5      <meta charset="UTF-8">
6      <title>Title</title>
7      <style>
8          /*#d1 {*/

```



```

9      /*      !*display: none; !*隐藏标签，不显示在前端页面，并且原来的位置也不占有
      了，但是还存在文档上*!\*!*/
10      /*      display: inline;*/
11      /*}*/
12      /*#d2{*/
13      /*      display: inline;*/
14      /*}*/
15      /*#d3{*/
16      /*      display: block; !*将标签设置成块儿级标签的特点*!*/
17      /*}*/
18      /*#d4{*/
19      /*      display: block;*/
20      /*}*/
21
22      /*#d1{*/
23      /*      display: inline-block;  !* 让标签既可以在一行显示，又可以设置长宽
      *!*/
24      /*}*/
25      /*#d2{*/
26      /*      display: inline-block;*/
27      /*}*/
28  </style>
29 </head>
30 <body>
31 <div style="display: none">div1</div>
32 <div>div2</div>
33 <div style="visibility: hidden">单纯的隐藏，位置还在</div>
34 <div>div4</div>
35
36 <!--<div id="d1" style="height: 100px ;width: 100px; background-color:
      blue">01</div>-->
37 <!--<div id="d2" style="height: 100px ;width: 100px; background-color:
      orange">02</div>-->
38 <!--<span id="d3" style="height: 100px ;width: 100px; background-color:
      grey" ></span>-->
39 <!--<span id="d4" style="height: 100px ;width: 100px; background-color:
      yellow"></span>-->
40
41 <!--<div id="d1" style="height: 100px ;width: 100px; background-color:
      blue">01</div>-->
42 <!--<div id="d2" style="height: 100px ;width: 100px; background-color:
      orange">02</div>-->
43 </body>
44 </html>
45

```

盒子模型

```

1  # margin 外边距（标签与标签之间的距离）
2  # border 边框
3  # padding 内填充/内边距
4  # content 内容
5
6  ""
7  body默认自带8px 的margin

```

```

8  ""
9
10 <!DOCTYPE html>
11 <html lang="en">
12 <head>
13     <meta charset="UTF-8">
14     <title>Title</title>
15     <style>
16         /**{*/
17         /* margin: 0 ;*/
18         /* padding:0;*/
19         /**}*/
20         #d1{
21             margin-bottom: 50px;
22         }
23         #d2{
24             margin-top: 20px; /*不叠加，只取最大的*/
25         }
26         #dd{
27             margin:0 auto; /*只能做到标签的 水平居中，*/
28         }
29         p{
30             border:3px solid red;
31             /*padding-left: 30px;*/
32             /*padding-top: 20px;*/
33             /*padding-right: 20px;*/
34             /*padding-bottom: 20px;*/
35
36             /*padding: 10px;*/
37             /*padding:10px 20px;*/
38             /*padding:10px 20px 30px;*/
39             padding: 10px 20px 30px 40px;
40         }
41     </style>
42 </head>
43 <body>
44 <!-- <p style="border:1px solid red" id="d1">ppp</p>-->
45 <!-- <p style="border:1px solid blue" id="d2">ppp</p>-->
46 <!--<div style="border:5px solid red;height: 400px;width:400px">-->
47 <!-- <div id="dd" style="border:1px solid orange ;width:50px;height:
48 50px;background-color: blue "></div>-->
49 <!--</div>-->
50 <p>ppp</p>
51 </body>
52 </html>
53
54
55
56

```

浮动

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6   <style>
7     body{
8       margin:0;
9     }
10    #d1{
11      width: 20%;
12      height: 1000px;
13      background-color: darkgray;
14      float: left;
15    }
16    #d2{
17      width: 80%;
18      height: 2000px;
19      background-color: wheat;
20      float: right;
21    }
22  </style>
23 </head>
24 <body>
25 <div id="d1"></div>
26 <div id="d2"></div>
27 </body>
28 </html>
```

浮动带来的影响

```
1 # 浮动带来的影响
2 #1 自己加一个div设置高度
3 #2 利用clear属性
4   #d4{
5
6     clear: left;    /*该标签的左边不能有浮动的元素*/
7   }
8
9 #3 通用的解决浮动带来的影响方法
10 再写HTML代码之前，先提前写好处理了浮动带来的影响 css代码
11 .clearfix:after {
12     content: '';
13     display: block;
14     clear: both;
15 }
16 之后只要标签出现了塌陷问题，就给该标签加一个clearfix属性即可
17 上述的解决方式是通用的，到哪里有一样，并且名字就叫clearfix
```

溢出属性

```
1  /*overflow:visible;!*默认就是可见的**/  
2      /*overflow: hidden;!*溢出部分隐藏**/  
3      overflow:scroll ;/* 设置成上下滚动条的形式 */  
4      overflow:auto;/* 自动 */
```

定位

- 相对定位
相对于标签原来的位置做移动 relative
- 绝对定位
相对于已经定位过的父标签做移动 absolute (没有父标签那么就以body 为参照)
- 固定定位 fix
相对于浏览器窗口固定在某个位置
- 静态
所有的标签都是静态的 static，无法改变位置

ps:浏览器优先展示文本内容，如果发现文本内容被挡住了，会想方设法的找个位置展示出来

```
1  <!DOCTYPE html>  
2  <html lang="en">  
3  <head>  
4      <meta charset="UTF-8">  
5      <title>Title</title>  
6      <style>  
7          #d1 {  
8              height: 100px;  
9              width: 100px;  
10             background-color: red;  
11             left: 50px; /*距离左边50px*/  
12             top: 50px; /* 距离上面50px*/  
13             /*position: static;!*默认是static，无法修改位置**/  
14             /*position: relative;*/  
15             /* 相对定位：  
16                 标签由static变为relative 它的性质就从原来没有定位的标签变成了已  
17                 经定位过的标签  
18                 虽然位置没有动，但是性质已经发生了改变*/  
19             }  
20             #d2 {  
21                 height: 100px;  
22                 width: 200px;  
23                 background-color: yellowgreen;  
24                 position: relative;  
25             }  
26             #d3 {  
27                 height: 200px;
```

```

30         width: 400px;
31         background-color: rosybrown;
32         position: absolute;
33         left: 200px;
34         top: 100px;
35     }
36 </style>
37 </head>
38 <body>
39 <!--<div id="d1"></div>-->
40 <!--<div id="d2">python-->
41 <!--    <div id="d3"></div>-->
42 <!--</div>-->
43 <div style="height: 500px;background-color: red"></div>
44 <div style="height: 500px;background-color: blueviolet"></div>
45 <div style="height: 500px;background-color: darkcyan"></div>
46 <div style="position: fixed;bottom: 10px ;right:20px; height: 50px;width:
100px;background-color: white;border: 5px solid black"></div>
47 </body>
48 </html>

```

验证浮动和定位是否脱离文档流（原来的位置是否还保留）

```

1  """
2  浮动
3  相对定位
4  固定定位
5  绝对定位
6  """
7
8  # 不脱离文档流
9      1.相对定位
10
11 # 脱离文档流
12     1. 浮动
13     2.绝对定位
14     3. 脱离文档流
15

```

z-index模态框

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <style>
7          body{
8              margin: 0;
9          }
10         .cover{
11             position: fixed;
12             left: 0;
13             top:0;

```

```

14     right: 0;
15     bottom: 0;
16     background-color: rgba(0,0,0,0.4);
17     z-index: 99;
18 }
19 .modal{
20     background-color: white;
21     width: 400px;
22     height: 200px;
23     position: fixed;
24     left: 50%;
25     top: 50%;
26     z-index: 100;
27     margin-left:-200px ;
28     margin-top: -100px;
29
30 }
31
32 </style>
33 </head>
34 <body>
35 <div>这是最底层的页面内容</div>
36 <div class="cover"></div>
37 <div class="modal">
38     <h1>登录页面</h1>
39     <p>username:
40         <input type="text">
41     </p>
42     <p>password:
43         <input type="text">
44     </p>
45     <button>Click Me!!</button>
46 </div>
47 </body>
48 </html>

```

透明度opacity

```

1  ""不仅可以修改颜色的透明度，也可以修改字体的透明度""
2  rgba只能影响颜色
3  而opacity可以修改颜色和字体
4  <style>
5      #d1{
6          background-color: rgba(0,0,0,0.5);
7      }
8      #d2{
9
10         opacity: 0.3;
11         color: black;
12     }
13 </style>

```