

html概述

什么是html?

HTML 是用来描述网页的一种语言。

- HTML 指的是超文本标记语言 (**H**yper **T**ext **M**arkup **L**anguage)
- HTML 不是一种编程语言，而是一种**标记语言**(markup language)
- 标记语言是一套**标记标签** (markup tag)
- HTML 使用**标记标签**来描述网页
- 【超文本】：网页本身是一个文本文件，而超文本指的是这种文件中既可以包含文本信息，又可以包含图片，音频，视频和链接等非文字的信息。
- 【标记语言】：标记，也叫做标签。也就是说HTML这门语言是由标签组成的。
- HTML通过提前约定好的标签来构建我们的网页内容，通过浏览器来进行解释执行，浏览器是一个HTML的解释器，在浏览器中内置了一个解释器，该解释器中包含了所有HTML标签的展示风格

html由什么组成?

html文档==网页

- HTML 文档**描述网页**
- HTML 文档 包含 HTML **标签**和**纯文本**
- HTML 文档也被称为**网页**

Web 浏览器的作用是读取 HTML 文档，并以网页的形式显示出它们。浏览器不会显示 HTML 标签，而是使用标签来解释页面的内容

什么是html标签?

HTML 标记标签通常被称为 HTML 标签 (HTML tag)。

- HTML 标签是由尖括号包围的关键词，比如 `<html>`
- HTML 标签通常是**成对出现**的，比如 `` 和 ``
- 标签对中的第一个标签是**开始标签**，第二个标签是**结束标签**
- 开始和结束标签也被称为**开放标签**和**闭合标签**

用来编写html?

html只是一种标记语言，它的运行依靠浏览器对其的解释，因此编写html只需要像写文档那样即可，并不需要特殊的编译环境

常用的编写html的免费工具有：VScode等

[VScode下载](#)

需要中文版可以下载中文插件，具体参考网络教程

第一次编写html

1. 打开VScode
2. 新建一个文件，将文件命名为 文件名.html
3. 在文件中输入 ! （英文状态下） + Enter此时VScode会自动生成html的必要骨架，如图

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title></title>
8  </head>
9  <body>
10 </body>
11 </html>
```

4. 在间输入任意文字
5. 先Ctrl+S保存 再Alt+B快捷默认浏览器打开，就制作出你的第一个网页了

html骨架

以下就是html5骨架的大致描述

```
1  <!DOCTYPE html><!-- 这是网所使用的html语言类型，这是html5的表示 -->
2  <html lang="en"><!-- 这是网站所使用的语言，这是英语的表示，zh-CN（中文） -->
3  <head>
4      <!-- 这里是网页的头部，用来定义或引用一些东西 -->
5      <meta charset="UTF-8"><!-- 这是使用的文字编码 -->
6      <meta http-equiv="X-UA-Compatible" content="IE=edge"><!-- 用于解决ie浏览器的兼容性 -->
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <!-- 为了和移动端兼容 -->
9      <title>这里是网页的标题</title>
10 </head>
11 <body>
12     这里是网页的主体
13 </body>
14 </html>
```

html注释

在学习一门新的编程语言的时候我们应该先学会的是这个语言的注释格式

从而使我们在学习代码的途中更好的去注释

- 在html里就是注释的格式

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>html注释</title>
8  </head>
9  <body>
```

```
10
11      <!-- 这是html语言的注释格式 -->
12      <!-- 注释一行的快捷键 ctrl+/ -->
13      <!-- 注意!!! 不能使用数字键盘上的斜杠 -->
14  </body>
15  </html>
```

```
1  <html> </html> : 页面中最大的标签, 也称根标签
2  <head> </head> : 文档的头部, 在 head 标签中必须设置 title (标题)
3  <title> </title> : 文档的标题, 让网页拥有一个标题
4  <body> </body> : 文档的主体, 所有内容基本都放在 body 里
```

html标签

标签是干什么的?

在前边我们就说过html语言是有各种标签和纯文本组成的, 其中纯文本就是网页中显示的内容, 而标签则是浏览器需要识别并作出相应布局的东西

都有什么标签?

HTML的标签包含以下两种:

- (1) 带有标签体的标签< a>标签体< /a > (有些标签的标签体可以为空)
- (2) 不带标签体的标签(自结束标签)< br/>,< hr/> (在html中自结束标签中的/可以不写也能够被识别成功, 但是为了标签能够形成都需要结束标签表示结束的规范, 建议都带上/)

常用的html标签有哪些?

- 标题: h1 ~ h6 标签

1. <h1> </h1> : 一级标题
2. <h2> </h2> : 二级标题
3. <h3> </h3> : 三级标题
4. <h4> </h4> : 四级标题
5. <h5> </h5> : 五级标题
6. <h6> </h6> : 六级标题

- 段落: p 标签

<p> </p> : 生成一个段落

- 换行: br 标签

 : break 的缩写, 强制换行

- 水平线: hr标签

<hr/> : 做一条水平线

- 文本格式化

1. 或 : 加粗
2. 或 <i> </i> : 倾斜
3. 或 <s> </s> : 删除线

4. `<ins> </ins>` 或 `<u></u>` : 下划线

- **盒子标签**

1. `<div> </div>` 标签用来布局, 但是一行只能放**一个** div 标签
2. ` ` 标签用来布局, 但是一行可以放**多个** span 标签

- **图像: img 标签**

- 标签格式: ``

- **超链接: a 标签**

标签格式: `文本或图像`

html 元素

html 元素指的是从开始标签 (start tag) 到结束标签 (end tag) 的所有代码。

html元素和标签的区别, 简单的理解:

- ```
1 <p>这就是标签:
2 <p>这是内容</p>这是一个元素, 也就是说元素由一个开始标签和结束标签组成, 用来包含内容
3 有一值得注意的例外,
本身就是开始标签也是结束标签, 不包含内容, 所以这只是一个标
 签。
```

## html元素的语法

- HTML 元素以**开始标签**起始
- HTML 元素以**结束标签**终止
- **元素的内容**是开始标签与结束标签之间的内容
- 某些 HTML 元素具有**空内容** (*empty content*)
- 空元素在**开始标签中进行关闭** (以开始标签的结束而结束)
- 大多数 HTML 元素可拥有**属性**

## 嵌套的html元素

大多数 html 元素可以嵌套 (可以包含其他 HTML 元素) 。

html 文档由嵌套的 html 元素构成。

## 不要忘记结束标签

即使您忘记了使用结束标签, 大多数浏览器也会正确地显示 HTML:

- ```
1 <p>This is a paragraph
2 <p>This is a paragraph
```

上面的例子在大多数浏览器中都没问题, 但不要依赖这种做法。忘记使用结束标签会产生不可预料的结果或错误。

html属性

HTML 标签可以拥有**属性**。属性提供了有关 HTML 元素的**更多的信息**。

属性总是 (但非一定) 以名称/值对的形式出现, 比如: `name="value"`。

属性总是在 HTML 元素的开始标签中规定。

如何去使用属性？

- 1 链接由 `<a>` 标签定义。链接的地址在 `href` 属性中指定：
- 2 `This is a link`

大多数标签都可以使用的属性

属性	值	描述
class	<i>classname</i>	规定元素的类名 (classname)
id	<i>id</i>	规定元素的唯一 id
style	<i>style_definition</i>	规定元素的行内样式 (inline style)
title	<i>text</i>	规定元素的额外信息 (可在工具提示中显示)

至于这些属性的使用我们会在后边涉及

html基础标签

html标签众多，但常用的基础标签并不多，很快就能学完

标题标签

标题标签用来声明标题，利用好标题标签也有利于用在浏览器中更好的发现你的网站

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>html标题标签</title>
8 </head>
9 <body>
10  <h1>1级标题</h1>
11  <h2>2级标题</h2>
12  <h3>3级标题</h3>
13  <h4>4级标题</h4>
14  <h5>5级标题</h5>
15  <h6>6级标题</h6>
16  <!-- 标题特点：1.标题独占一行 2.标题有加粗效果 -->
17  <!-- 注意！！！标题只有一级到六级 -->
18 </body>
19 </html>
```

段落标签

段落标签常被用来填充文本内容

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
```

```

4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>html段落标签</title>
8 </head>
9 <body>
10  <p>This is a paragraph</p>
11  <p>This is another paragraph</p>
12  <!-- 特点: 1.独占一行 2.段落之间有间距 -->
13 </body>
14 </html>

```

换行，水平线标签

换行标签和水平线标签都是比较常用的标签，水平线常用来分割面板

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>html换行，水平线标签</title>
8  </head>
9  <body>
10   <p>这句话后边的话将换行<br/>这行话被换行了</p>
11   <p>
12       春眠不觉晓，
13       处处闻啼鸟。
14       夜来风雨声，
15       花落知多少。
16       <hr/>
17       <!-- <hr/>是一个水平线标签 -->
18   </p>
19
20   <!-- 注意，浏览器会自动忽略源代码中的排版（省略了多余的空格和换行，它们都将值被识别为一个空格）。 -->
21   <!-- 段落的行数依赖于浏览器窗口的大小。如果调节浏览器窗口的大小，将改变段落中的行数。 -->
22 </body>
23 </html>

```

文本格式化标签

对于将文本加粗，斜体，删除线等效果统称为文本格式化

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>html文本格式化标签</title>
8  </head>
9  <body>

```

```

10 <!-- 让文本出现加粗，下划线，倾斜，删除线效果称为文本格式化 -->
11 <!-- 每种格式化的方式都有两种方法，单词形式的具有强调效果（但是外观一样）比如特价 -->
12 <b>b加粗</b>
13 <strong>strong加粗</strong>
14 <u>u下划线</u>
15 <ins>ins下划线</ins>
16 <i>i倾斜</i>
17 <em>em倾斜</em>
18 <s>s删除线</s>
19 <del>删除线</del>
20 </body>
21 </html>

```

媒体标签

从这里开始我们将接触链接和文件路径

因此需要了解文件的先对路径和绝对路径

- 绝对路径：从根目录开始的路径，在win系统下指的是从C盘或D盘等开始的路径，如 C:\Program Files\Adobe\Common。在linux系统中指从./开始的文件路径，显然绝对路径很不方便，不仅不同系统下路径的起始不同，并且当挪动文件时就会造成路径失效等问题
- 相对路径：从当前文件开始的路径，这也是我们最常用的路径，./指当前文件../指返回上一级目录

媒体标签：图片 音频 视频 超链接等

下面代码的相关文件统一打包在webcode文件夹

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>html媒体标签</title>
8 </head>
9 <body>
10  <!-- 媒体标签：图片 音频 视频 超链接等 -->
11  <!-- 媒体标签的属性
12      1.src 链接
13      2.alt 替换文本
14      3.title 提示文本
15      4.width和height 宽高 一般只需要使用其中一个，另一个会同图片等比例缩放 -->
16 <hr/>
17  <!-- 1. 图片标签 img-->
18  
19 <hr/>
20  <!-- 2. 绝对路径和相对路径 -->
21  <!-- 建议使用相对路径 -->
22  <!-- 绝对路径 自盘符开始的文件路径-->

```

```

23 
24 <!-- 相对路径 自当前文件开始的文件路径 -->
25 
26 
27 <hr/>
28 <!-- 3. 音频标签 audio（建议使用mp3格式）-->
29 <!-- 属性：
30     1.src 文件路径
31     2.controls 播放控件
32     3.autoplay 自动播放（有的浏览器不支持）
33     4.loop 循环播放 -->
34 <audio src="audio/7-html媒体标签-1.mp3" controls autoplay loop></audio>
35 <hr/>
36 <!-- 4. 视频标签 video（建议使用MP4格式）-->
37 <!-- 属性：
38     1.src 文件路径
39     2.controls 播放控件
40     3.autoplay 自动播放（谷歌浏览器需要配合muted才能实现静音播放）
41     4.loop 循环播放 -->
42 <video src="video/7-html媒体标签-1.mp4" controls autoplay muted loop>
</video>
43 <hr/>
44 <!-- 5. 链接标签 a -->
45 <!-- 属性：
46     1.href 跳转地址
47     2.target 新页面打开形式 取值 _self 默认值,在当前页面打开 _blank 在新
的页面打开 -->
48 <a href="https://www.w3school.com.cn/index.html">w3school</a>
49 <!-- 注意!!!在网站开发初期不知道跳转地址时写#,#被称为空链接 -->
50 <a href="#">这是一个空链接,点击它会重新跳转至本页面</a>
51 <a href="https://www.w3school.com.cn/index.html"
target="_self">w3school这是一个加了_self的链接</a>
52 <a href="https://www.w3school.com.cn/index.html"
target="_blank">w3school这是一个加了_blank的链接</a>
53
54 </body>
55 </html>

```

!!! 列表，表格，表单的区别

列表：列表常用于列出一系列相似的事物

- 芝麻开门
- 未雨绸缪
- 温文尔雅

1. 芝麻开门
2. 玛卡巴卡
3. 叶斯摩拉

网址之家
百度
腾讯
360

表格：表格常用于显示不同属性的事物间的某种联系

姓名	年龄	性别
小数	20	阿斯頓
整数	21	手动分
负数	22	请问饿

表单：表单常用于让用户提交信息

用户名：

密 码：

男 ☒ 女 ☐ 待定 ☐

请选择你的爱好

吃饭 ☐ 树胶 ☐ 睡觉 ☐

请选择你的家乡

河南 ▼

请键入你的感想：

叮叮当,叮叮当,铃儿响叮当

列表标签

列表分为三种

1. 无序列表

1. 有序列表

1. 自定义列表

- 芝麻开门
- 未雨绸缪
- 温文尔雅

1. 芝麻开门
2. 玛卡巴卡
3. 叶斯摩拉

网址之家
百度
腾讯
360

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>html列表标签</html></title>
8 </head>
9 <body>
10   <!-- 列表 1. 无序列表
11       2. 有序列表
12       3. 自定义列表 -->
13   <!-- 1. 无序列表 <ul>无序列表标签 <li>无序列表的每一项的内容 注意: <ul>标签内只能
14       写<li>标签, <li>标签内可以有其他标签-->
15       <ul>水果
16         <li>苹果</li>
17         <li>西瓜</li>
18         <li>香蕉</li>
19       </ul>
20   <!-- 2. 有序列表 <ol>有序列表标签 <li>无序列表的每一项的内容 注意: <ol>标签内只能
21       写<li>标签, <li>标签内可以有其他标签-->
22       <ol>成绩排名
23         <li>张三</li>
24         <li>李四</li>
25         <li>王五</li>
26       </ol>
27   <!-- 3. 自定义列表 <dl>无序列表标签 <dt>自定义列表的主题 <dd>每一项的内容 注
28       意: <dl>标签内只能写<li>标签, <dt><dd>标签内可以有其他标签-->
```

```

26         <dl>这是一个自定义标签
27             <dt>帮助中心
28                 <dd>账户管理</dd>
29                 <dd>购物指南</dd>
30                 <dd>订单操作</dd>
31             </dt>
32         </dl>
33
34     </body>
35 </html>

```

表格标签

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-
scale=1.0">
7      <title>html 表格标签</title>
8  </head>
9  <body>
10     <!-- 表格标签 <table>创建表格 <tr>创建行 <td>创建列 -->
11     <!-- 属性:
12         1.border 边框宽度
13         2.wight 表格宽度
14         3.height 表格高度
15         注意: 实际开发中一般在css添加外观效果-->
16     <table border="1" width="500" height="300">
17         <tr>
18             <td>姓名</td>
19             <td>成绩</td>
20             <td>评语</td>
21         </tr>
22         <tr>
23             <td>张三</td>
24             <td>88</td>
25             <td>6</td>
26         </tr>
27         <tr>
28             <td>李四</td>
29             <td>58</td>
30             <td>菜</td>
31         </tr>
32     </table>
33     <hr>
34     <!-- 表格标题和表头标签 <caption>表格整体大标题 <th>表示一系列的小标题
35         注意点: <caption>标签书写在table标签内部, <th>标签书写在<tr>标签内部 (用
于替换<td>) -->
36     <table border="1">
37         <caption><strong>学生成绩单</strong></caption>
38         <tr>
39             <th>姓名</th>
40             <th>成绩</th>

```

```

41         <th>评语</th>
42     </tr>
43     <tr>
44         <td>张三</td>
45         <td>88</td>
46         <td>6</td>
47     </tr>
48     <tr>
49         <td>李四</td>
50         <td>58</td>
51         <td>菜</td>
52     </tr>
53 </table>
54 <hr>
55 <!-- 表格的结构标签(没有明显区别,但是方便css注入)
56 1.表格头部 <thead>
57 2.表格主体 <tbody>
58 3.表格底部 <tfoot> -->
59 <table border="1">
60     <caption><strong>学生成绩单</strong></caption>
61     <thead>
62         <tr>
63             <th>姓名</th>
64             <th>成绩</th>
65             <th>评语</th>
66         </tr>
67     </thead>
68     <tbody>
69         <tr>
70             <td>张三</td>
71             <td>88</td>
72             <td>优秀</td>
73         </tr>
74         <tr>
75             <td>李四</td>
76             <td>99</td>
77             <td>优秀</td>
78         </tr>
79     </tbody>
80     <tfoot>
81         <tr>
82             <td>总结</td>
83             <td>完美</td>
84             <td>相亲成功</td>
85         </tr>
86     </tfoot>
87 </table>
88 <hr>
89 <!-- 合并单元格
90 给保留的单元格加代码
91 1.上下合并 跨行合并 保留上边<rowspan>
92 2.左右合并 跨列合并 保留左边<colspan>
93 -->
94 <table border="1">
95     <caption><strong>学生成绩单</strong></caption>

```

```

96     <thead>
97         <tr>
98             <th>姓名</th>
99             <th>成绩</th>
100            <th>评语</th>
101        </tr>
102    </thead>
103    <tbody>
104        <tr>
105            <td>张三</td>
106            <td rowspan="2">88</td>
107            <td>优秀</td>
108        </tr>
109        <tr>
110            <td>李四</td>
111            <!-- <td>99</td> 被合并下的单元格 -->
112            <td>优秀</td>
113        </tr>
114    </tbody>
115    <tfoot>
116        <tr>
117            <td>总结</td>
118            <td colspan="2">完美</td>
119            <!-- <td>相亲成功</td> 被合并的右单元格 -->
120        </tr>
121    </tfoot>
122 </table>
123 <!-- 注意:不能跨表格结构合并单元格!!! -->
124 <hr>
125
126 </body>
127 </html>

```

表单标签

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>html表单标签</title>
8  </head>
9  <body>
10     <!-- 表单标签的使用场景较为局限 常用于搜索,注册等 -->
11     <!-- 表单标签 <input>
12         通过属性值type的不同,表现不同的效果
13         <input> 标签用于搜集用户信息。
14         根据不同的 type 属性值,输入字段拥有很多种形式。输入字段可以是文本字段、复选
            框、掩码后的文本控件、单选按钮、按钮等等。
15
16         type值          描述
17         button          定义可点击按钮（多数情况下,用于通过 JavaScript 启动脚本）。
18         checkbox        定义复选框。
19         file             定义输入字段和 "浏览"按钮,供文件上传。

```

```

20     hidden    定义隐藏的输入字段。
21     image    定义图像形式的提交按钮。
22     password  定义密码字段。该字段中的字符被掩码。
23     radio    定义单选按钮。
24     reset    定义重置按钮。重置按钮会清除表单中的所有数据。
25     submit   定义提交按钮。提交按钮会把表单数据发送到服务器。
26     text     定义单行的输入字段，用户可在其中输入文本。默认宽度为 20 个字符。
27 -->
28 <!-- 表单的父级标签<form>表单域 作用:使表单组成一个体系 -->
29     <!-- sction是提交数据的地址 -->
30     <form action="">
31     <hr>
32         <input type="button" value="value显示表单文字"> 1.button    定义可点
击按钮（多数情况下，用于通过 JavaScript 启动脚本）。
33     <hr>
34         <input type="checkbox" name="" id=""> 2.checkbox    定义复选框。
35     <hr>
36         <!-- multiple多文件选择 -->
37         <input type="file" name="" id="" multiple> 3.file    定义输入字段和
"浏览"按钮，供文件上传。
38     <hr>
39         <input type="hidden" name=""> 4.hidden    定义隐藏的输入字段。
40     <hr>
41         <input type="image" src="./img/12-html表单标签.jpg" alt="当图片无法显
示时出现的替换文本" width="10%"> 5.image    定义图像形式的提交按钮。
42     <hr>
43         <input type="password" name="" id=""> 6.password    定义密码字段。该字
段中的字符被掩码。
44     <hr>
45         <!-- 拥有相同name属性值的为同一组
         checked 默认选中-->
46         <input type="radio" name="性别" id="" checked>男
47         <input type="radio" name="性别" id="">女
48         7.radio    定义单选按钮。
49     <hr>
50         <input type="reset" value=""> 8.reset    定义重置按钮。重置按钮会清除
表单中的所有数据。
51     <hr>
52         <input type="submit" value=""> 9.submit    定义提交按钮。提交按钮
会把表单数据发送到服务器。
53     <hr>
54         <input type="text" name="" id="" placeholder="这里是默认文本">
10.text    定义单行的输入字段，用户可在其中输入文本。默认宽度为 20 个字符。
55     <hr>
56     <!-- <button>可以作为标签名使用
         type属性:1.submit 2.reset 3.button -->
57         <button type="submit">提交按钮</button>
58         <button type="reset">重置按钮</button>
59         <button type="button">默认, 普通按钮</button>
60         <button></button>
61     </form>
62 </body>
63 </html>

```

下拉菜单标签

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>html下拉菜单标签</title>
8 </head>
9 <body>
10   <!-- 定义和用法
11       select 元素可创建单选或多选菜单。
12       <select> 元素中的 <option> 标签用于定义列表中的可用选项。
13
14       属性      值      描述
15       autofocus autofocus 规定在页面加载后文本区域自动获得焦点。
16       disabled  disabled 规定禁用该下拉列表。
17       form      form_id 规定文本区域所属的一个或多个表单。
18       multiple  multiple 规定可选择多个选项。
19       name      name    规定下拉列表的名称。
20       required  required 规定文本区域是必填的。
21       size      number  规定下拉列表中可见选项的数目。
22   -->
23   <select name="选择城市" id="">
24     <option value="">北京</option>
25     <option value="">上海</option>
26     <option value="" selected>深圳 selected默认选中</option>
27   </select>
28 </body>
29 </html>
```

label标签

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>label标签</title>
8 </head>
9 <body>
10   <!-- label标签常用于绑定内容与表单标签的关系
11       两种使用方法，注意两者有什么不同-->
12   性别：
13   <input type="radio" name="sex" id="男" checked> <label for="男">男
14   </label>
15   <label><input type="radio" name="sex"> 女</label>
16 </body>
17 </html>
```

文本域标签

文本域标签被用来让用户输入文本内容，在文本域标签中，文本的内容会按照原有格式显示

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>html 文本域标签</title>
8 </head>
9 <body>
10   <!-- 定义和用法
11       <textarea> 标签定义多行的文本输入控件。
12       文本区中可容纳无限数量的文本，其中的文本的默认字体是等宽字体（通常是
13       Courier）。
14       可以通过 cols 和 rows 属性来规定 textarea 的尺寸，不过更好的办法是使用 CSS
15       的 height 和 width 属性。
16       注释：在文本输入区内的文本行间，用 "%OD%OA" （回车/换行）进行分隔。
17   -->
18   <textarea name="" id="" cols="30" rows="10">文本域 这里是默认的显示
19 </textarea>
20 </body>
21 </html>
```

语义化标签

语义化标签被用来制作网页时区分网页的区域，方便css的注入

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>语义化标签</title>
8 </head>
9 <body>
10   <!-- 没有语义的标签
11       1.div
12       2.span -->
13   <div>这是div标签</div>
14   <span>这是span标签</span>
15   <!-- 有语义的标签(在html5推出,用于手机端网页开发)
16       1.header    网页头部
17       2.nav        网页导航
18       3.footer    网页底部
19       4.aside     网页侧边栏
20       5.section   网页区块
21       6.article   网页文章-->
22   
```



```
23 <header>网页头部</header>
24 <nav>网页导航</nav>
25 <footer>网页底部</footer>
26 <aside>网页侧边栏</aside>
27 <section>网页区块</section>
28 <article>网页文章</article>
29 <!-- 这些标签都没有实际效果,主要是为了方便css注入效果 -->
30 </body>
31 </html>
```

实体字符

由于在html中的文本不会保持原有格式,所以一些特殊字符就需要特殊的表示方式,其中最常用的是空格

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>实体字符</title>
8 </head>
9 <body>
10  <!-- 由于在代码中的多个空格和回车都会被自动识别为一个空格,因此要使用实体字符
11      主要使用空格 &nbsp;-->
12    <p>这中间      打了五个空格,但是只显示一个</p>
13    <p>这中间打了&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;五个'&nbsp;'</p>
14
15 </body>
16 </html>
```

css基础

- 什么是CSS?

CSS: Cascading Style Sheet 层叠样式表

是一组样式设置的规则,用于控制页面的外观样式

- 为什么使用CSS

实现内容与样式的分离,便于团队开发样式复用,便于网站的后期维护页面的精确控制,让页面更精美

- CSS作用

页面外观美化

布局和定位

css体验

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
```

```

4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>css体验</title>
8   <style>
9       /* css注释 在这里写的都是css */
10      /* 格式：
11          选择器{
12              css属性
13          }
14      */
15      /* 选择器的作用是查找标签 */
16      p{
17          /* p标签的文字颜色变成红色 */
18          color: red;
19          /* 字变大 px是像素*/
20          font-size: 30px;
21          /* 背景颜色 */
22          background: blue;
23          /* width height */
24          width: 400px;
25          height: 200px;
26      }
27  </style>
28 </head>
29 <body>
30   <p>这是一个注入了css的段落</p>
31   这是一句没有注入css的话
32 </body>
33 </html>

```

css的引入方式

css的引入方式有很多种，但是在我们学习时都是一个很小的项目，往往只需要一个页面就可以解决

所以在今后的学习中我们常用 **内嵌式** 引入css

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>css的引入方式</title>
8      <!-- rel="stylesheet"关系是样式表 -->
9      <link rel="stylesheet" href="./21-css的引入方式.css">
10     <style>
11         /*
12             css有三种引入方式
13             1.内嵌式: css写在style标签内（style标签可以写在任意地方，但是通常约定卸载title标签下方）
14             常用于小案例

```

```

15         2. 外联式: css写在一个单独的css文件中（需要通过link标签在网页引入）
16             项目中使用
17         3. 行内式: css写在标签的style属性中（配合js使用）
18             一般不会使用，某些时候会配合js使用
19     */
20 </style>
21 </head>
22 <body>
23     <p>这是一个p标签</p>
24     <div style="color: blue; font-size: 30px">这是一个div标签</div>
25 </body>
26 </html>

```

```

1 p{
2     color: red;
3 }

```

css标签选择器

选择器用于区分不同的标签或类等去赋予不同的样式

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>css标签选择器</title>
8     <style>
9         /* 结构:
10             选择器{
11
12             } */
13         /* 1. 标签选择器 即 以标签命名的选择器
14             所有被选中的标签都会生效
15         2. 类选择器 即 以类命名的选择器
16             所有具有同类名的标签都会生效
17             类名通过在标签内定义class属性的值定义
18         3. id选择器 即 以id命名的选择器
19             所有具有相同id名的标签都会生效
20             id名通过在标签内定义id属性的值定义
21             id选择器相同命名只能出现一次, 只有一个标签可以使用, id选择器
22             主要用于js添加动态效果
23         4. 通配符选择器 通配符"*", 常用于清除标签的默认内外边距
24         *{
25             margin: 0;
26             padding: 0;
27         }
28         /* 标签选择器 */
29         p{
30             color: red;

```

```

31     }
32     /* 类名选择器 */
33     .blue{
34         color: blue;
35     }
36     .size{
37         font-size: 30px;
38     }
39     /* id选择器 */
40     #green{
41         color: green;
42     }
43 </style>
44 </head>
45 <body>
46 <p>这是一个p标签，通过标签选择器改变样式</p>
47 <div>这是一个div标签</div>
48 <p>这还是一个p标签，通过标签选择器改变样式</p>
49 <hr>
50 <p class="blue">这是一个p标签，通过类选择器改变样式，显然类选择器的优先级高于标签
51 选择器</p>
52 <p class="blue size">这是一个p标签，通过类选择器改变样式，一个标签可以有多个类
53 名，通过空格隔开</p>
54 <div id="green">这是一个div标签，通过id选择器改变样式</div>
55 </body>
56 </html>

```

css字体样式

通过css改变标签内的字体样式，如大小，颜色等

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>字体样式</title>
8 <style>
9     /* 1. 字体字号 font-size 单位px像素 谷歌默认大小16px*/
10    .one{
11        font-size: 30px;
12    }
13    /* 2. 字体粗细 font-weight 单位可以是关键字或者数字,400和normal是默认效
14    果,700和bold是默认加粗效果*/
15    .two{
16        font-weight: 700 ;
17    }
18    /* 3. 字体倾斜 font-style 取值有normal 和 italic(倾斜) */
19    .three{
20        font-style: italic;
21    }

```

```

21      /* 4. 字体系列 font-family windows系统默认为微软雅黑字体 macOS默认是苹方字
   体 */
22      .four{
23          /* font-family: 宋体; */
24          font-family: 微软雅黑, 黑体, sans-serif;
25      }
26      /* 5. 样式的层叠问题 */
27      .five{
28          /* css注入属性具有后来者居上的特点, 也就是说当注入了两种相冲突的样式效果时
   后边的会覆盖前边的 */
29          color: red;
30          color: green;
31          color: blue;
32      }
33      /* 6. font属性的连写 */
34      .six{
35          /* font-size: ;
36             font-weight: ;
37             font-style: ;
38             font-family: ;
39             等效于
40             font: style weight size family; */
41             font: italic 700 30px 宋体;
42             /* 注意 font属性连写必须有字体大小和字体系列, 并且顺序必须为 style
   weight size family*/
43
44      }
45
46  </style>
47  </head>
48  <body>
49  <p class="one">字体字号 font-size 单位px像素 谷歌默认大小16px</p>
50  <p class="two">字体粗细 font-weight 单位可以是关键字或者数字, 400和normal是默
   认效果, 700和bold是默认加粗效果</p>
51  <p class="three">字体倾斜 font-style 取值有normal 和 italic(倾斜)</p>
52  <p class="four">字体系列 font-family windows系统默认为微软雅黑字体</p>
53  <p class="five">样式的层叠问题 css注入属性具有后来者居上的特点, 也就是说当注入了
   两种相冲突的样式效果时后边的会覆盖前边的</p>
54  <p class="six">font属性的连写</p>
55  </body>
56  </html>

```

css文本样式

通过css改变标签内的文本样式, 如行高, 对齐方式等

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>css文本样式</title>

```

```
8 <style>
9 /* 1.文本缩进 text-indent 单位px(像素)或em(一个字的大小) */
10 .one{
11     text-indent: 2em;
12 }
13 /* 2.文本水平对齐方式 text-align 取值left(左对齐) center(居中对齐)
righe(右对齐)
该标签也能用于图片等 需要注入其父级标签 */
14
15 .two{
16     text-align: left;/* left为默认效果 */
17     text-align: right;
18     text-align: center;
19 }
20 /* 3.文本关于线的修饰 text-decoration
取值:underline(添加下划线)
line-through(添加删除线)
overline(添加上划线,几乎不用)
none(无装饰线,常用于去除超链接的下划线)
*/
21
22 .three1{
23     text-decoration: underline;
24 }
25 .three2{
26     text-decoration: line-through;
27 }
28 .three3{
29     text-decoration: overline;
30 }
31 .three4{
32     text-decoration: none;
33 }
34 /* 4.行高 line-height 单位px或者倍数 行高指的是字体自身以及部分留白距离 */
35 .four1{
36     /* line-height: 40px; */
37     line-height: 0.5;
38 }
39 .four2{
40     /* 注意 行高也可以用于font连写 顺序为 font: style weight size/line-
height family(前两个可以省略) */
41     font: italic 700 30px/0.5 宋体;
42 }
43
44 </style>
45 </head>
46 <body>
47 <p class="one">每一个学校都希望自己的学生做到品学兼优，全面发展，树立良好形象，也
使我们的学校有一个良好形象。每一个同学也
都希望学校给自己一个良好的学习环境来学习，生活。包括我自己也希望可以有一个良好的
学习环境，但是一个好的学习环
境靠的是大家来共同维护来建立起来的，而我自己这次却犯了错误，去破坏了学校的良好环
境，是很不应该的，若每一个同学
都这样犯错，那么是不会有良好的学习环境形成</p>
48 <p class="two">文本水平对齐方式 text-align 取值left(左对齐) center(居中对齐)
righe(右对齐)
该标签也能用于图片等 需要注入其父级标签</p>
```

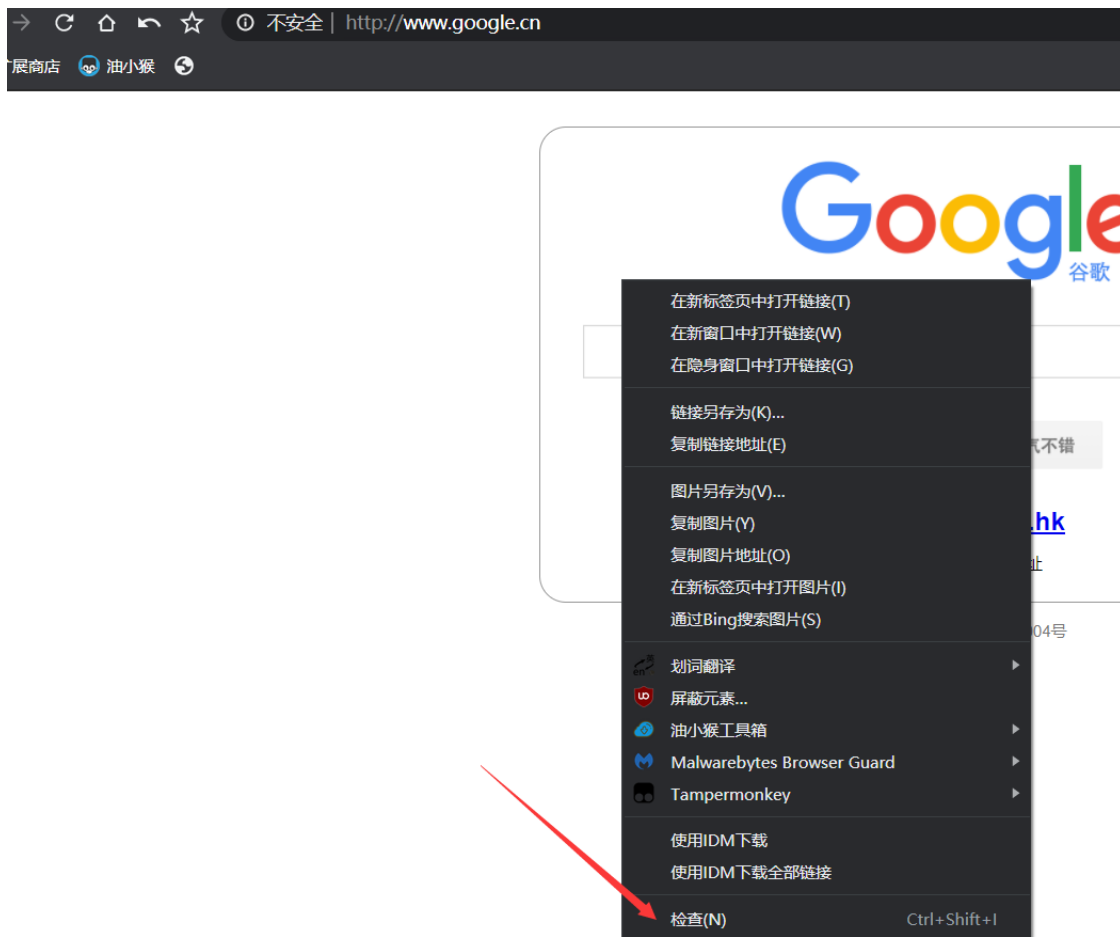
```

57 <div class="two"></div>
58 <p class="three1">underline(添加下划线)</p>
59 <p class="three2">line-through(添加删除线)</p>
60 <p class="three3">overline(添加上划线,几乎不用)</p>
61 <a href="#" class="three4">none(无装饰线)</a>
62 <p class="four1">行高 line-height 单位px或者倍数 行高指的是字体自身以及部分留
白距离 以下文字用于测试0.5倍行高:每一个学校都希望自己的学生做到品学兼优,全面发展,树
立良好形象,也使我们的学校有一个良好形象。每一个同学也
都希望学校给自己一个良好的学习环境来学习,生活。包括我自己也希望可以有一个良好的
学习环境,但是一个良好的学习环
境靠的是大家来共同维护来建立起来的,而我自己这次却犯了错误,去破坏了学校的良好环
境,是很不应该的,若每一个同学
都这样犯错,那么是不会有良好的学习环境形成</p>
65 <p class="four2">注意 行高也可以用于font连写 顺序为 font: style weight
size/line-height family(前两个可以省略) 以下文字用于测试:每一个学校都希望自己的
学生做到品学兼优,全面发展,树立良好形象,也使我们的学校有一个良好形象。每一个同学也
都希望学校给自己一个良好的学习环境来学习,生活。包括我自己也希望可以有一个
良好的学习环境,但是一个良好的学习环
境靠的是大家来共同维护来建立起来的,而我自己这次却犯了错误,去破坏了学校的
良好环境,是很不应该的,若每一个同学
都这样犯错,那么是不会有良好的学习环境形成</p>
70
71 </body>
72 </html>

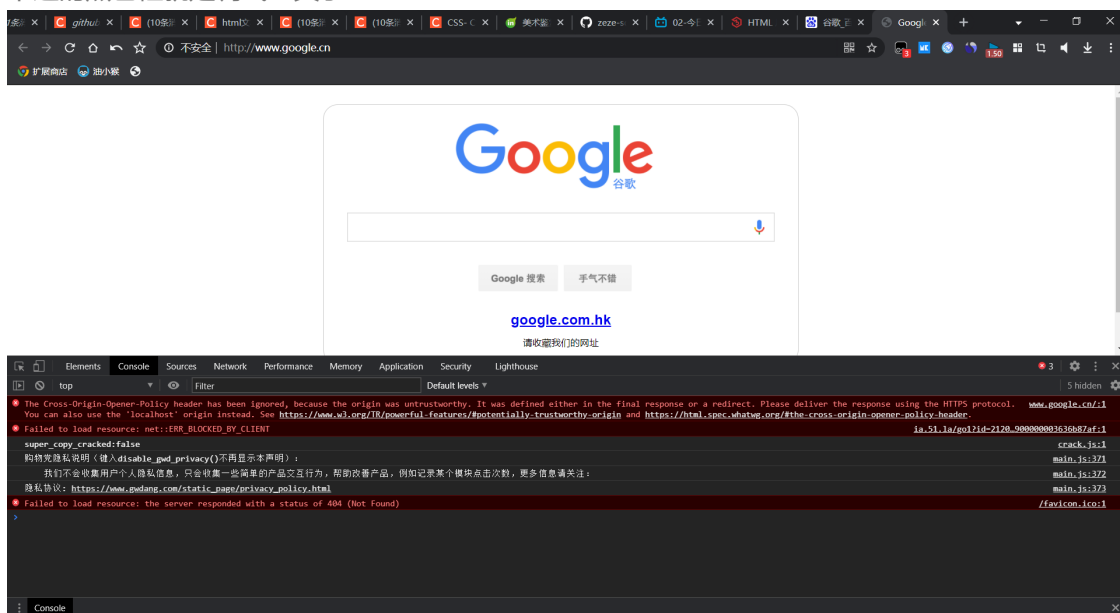
```

浏览器调试工具

每个浏览器都自带了应该调试工具,如谷歌浏览器,启动调试工具的快捷键是F12,或者在页面鼠标右键,点击检查



下边的黑色框就是调试工具了



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>25-css基础-浏览器调试工具</title>
8 </head>
9 <body>
10  <!-- 调试工具快捷键F12 -->
11  <p>具体详细教程参考该视频</p>
```



```
12 <video src="./video/25-css基础-浏览器调试工具.mp4" controls width="100%">
    </video>
13 </body>
14 </html>
```

css选择器进阶

上边的选择器只是挑选了标签，也就是说当标签相互嵌套式，很难甚至是无法挑选出其中某一个标签，因此选择器无疑需要更高级的形式

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>css选择器进阶</title>
8 <style>
9     /* 后代选择器
10         根据html标签的嵌套关系,选择父元素后代中满足的元素
11         语法：选择器1 选择器2{
12             CSS
13         }
14         在选择器1所在的标签内嵌套的标签中找到选择器2的标签
15     */
16     div p{
17         color: red;
18     }
19     /* 子代选择器
20         根据html标签的嵌套关系,选择父元素子代中满足的元素
21         语法：选择器1>选择器2{
22             CSS
23         }
24         在选择器1所在的标签内嵌套的下一级标签中找到选择器2的标签
25     */
26     p>span{
27         color: aqua;
28     }
29     /* 并集选择器
30         多个标签需要的效果相同时
31         语法： 选择器1,
32                选择器2,
33                选择器3{
34             CSS
35         }
36     */
37     h1,
38     h2,
39     h3{
40         color: rgb(230, 17, 190);
41     }
42     /* 交集选择器
```

```

43      一个较为特殊的标签,如在同一页面存在<p> 和 <p class="box">要求只改变<p
class="box">时
44      语法: 选择器1选择器2{
45          CSS
46      }
47      选择器1选择器2同时存在时起作用
48      */
49      p.box{
50          color: rgb(28, 228, 10);
51      }
52      /* 伪类选择器
53          当鼠标选定在元素上时产生效果
54          语法: 选择器:hover{
55              CSS
56          }
57          鼠标悬停在元素上时起作用
58          */
59      a:hover{
60          color: rgb(230, 255, 4);
61          background: rgb(231, 6, 6);
62      }
63  </style>
64  </head>
65  <body>
66  <p>这是一个p标签</p>
67  <div><p>这是一个div标签里嵌套的p标签</p></div>
68  <div><span><p>这是一个div标签里嵌套的span标签里嵌套的p标签</p></span></div>
69  <hr>
70  <p><span>这是一个p标签里嵌套的span标签</span></p>
71  <p><div><span>这是一个p标签里嵌套的p标签里嵌套的span标签</span></div></p>
72  <hr>
73  <h1>这是一个h1标签</h1>
74  <h2>这是一个h2标签</h2>
75  <h3>这是一个h3标签</h3>
76  <hr>
77  <p class="box">这是一个p标签</p>
78  <hr>
79  <a href="#">这是一个a标签</a>
80  </body>
81  </html>

```

css背景属性

用来给一个标签添加不同的背景

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8">
5  <meta http-equiv="X-UA-Compatible" content="IE=edge">
6  <meta name="viewport" content="width=device-width, initial-scale=1.0">
7  <title>css背景属性</title>
8  <style>

```

```

9      /* 1.背景色 background-color */
10     .one{
11         height: 400px;
12         background-color: pink;
13     }
14     /* 2.背景图 background-image */
15     .two{
16         height: 800px;
17         background-color: pink;
18         background-image: url(./img/29-css背景属性.jpg);
19     }
20     /* 3.背景图的背景平铺 background-repeat
21         取值      no-repeat    不平铺
22                  repeat      平铺（默认）
23                  repeat-x    沿x轴平铺
24                  repeat-y    沿y轴平铺
25     */
26     .three{
27         height: 800px;
28         background-color: pink;
29         background-image: url(./img/29-css背景属性.jpg);
30         background-repeat: no-repeat;
31     }
32     /* 4.背景位置 background-position
33         取值  1号位 left center right
34              2号位 top center bottom
35         或者以左上角为原点 单位为像素点的坐标轴摆放
36     */
37     .four{
38         height: 800px;
39         background-color: pink;
40         background-image: url(./img/29-css背景属性.jpg);
41         background-repeat: no-repeat;
42         background-position: 0 bottom;
43     }
44     /* 5.背景属性连写 background
45         连写顺序 background:color image repeat position;(这是建议的顺序，可以打乱)
46     */
47     .five{
48         height: 800px;
49         background: pink url(./img/29-css背景属性.jpg) no-repeat 0
50         bottom;
51     }
52 </style>
53 </head>
54 <body>
55     <div class="one">背景色</div>
56     <hr>
57     <div class="two">背景图</div>
58     <hr>
59     <div class="three">背景图取消平铺效果</div>
60     <hr>
61     <div class="four">背景图位置</div>
62     <hr>

```

```
62 <div class="five">背景图位置</div>
63 </body>
64 </html>
```

元素显示模式的重定义

对于标签来说，有行标签，块标签，行内块标签等，他们各自有各自的特点

- 块标签

特点：

1. 总是在新行上开始。
2. 高度，宽度，内外边距，都可以进行控制。
3. 宽度缺省是他的容器的100%，除非设定一个宽度。
4. 它可以容纳行内元素和其他块元素。

块标签包括：p、div、ul、ol、li、dl、dt、dd、h1~h6、form、table、td、thead、tr

- 行标签

特点：

1. 可以和其它元素共处一行，不用另起一行编写。
2. 元素的高度，宽度以及顶部和底部的内外边距不能进行设置，(因为设置了不起作用)
3. 该行标签的高度，宽度，是它包含的内容(图片，文字)的宽度，行标签不可以嵌套块标签。

行标签包括：a、abbr、b(字体加粗)、br、em、input、select、span、strong、sub、textarea

- 行内块标签

特点：

能设置宽高，还能和其他元素呆在一起

某些时候我们需要将标签的显示模式更改为另外的显示模式，这时就需要显示模式的重定义

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Document</title>
8 <style>
9     div{
10         background: pink;
11         color: red;
12         width: 400px;
13         height: 400px;
14     }
15     span{
16         background: pink;
17         color: red;
18         width: 400px;
19         height: 400px;
20     }
21     textarea{
```

```

22     background: pink;
23     color: red;
24     width: 400px;
25     height: 400px;
26 }
27 </style>
28 </head>
29 <body>
30 <!-- 元素指带<>的东西 -->
31 <!-- 1.块级标签
32     独占一行；默认宽度是父级元素宽度，高度默认由内容撑开；可以自定义宽高；
33     代表: div p h系列 ul li dl dt dd form header nav footer..... --
34 >
35 <div>这是第一个div标签</div>
36 <div>这是第二个div标签</div>
37 <hr>
38 <!-- 2.行内元素
39     一行可以显示多个；宽度、高度默认由内容撑开；不能自定义宽高
40     代表: a span b u i s strong ins em del..... -->
41 <span>这是第一个span标签</span>
42 <span>这是第二个span标签</span>
43 <hr>
44 <!-- 3.行内块元素
45     一行可以显示多个；宽度、高度默认由内容撑开；可以自定义宽高
46     代表: input textarea button select..... -->
47 <textarea name="" id="" cols="30" rows="10">这是第一个textarea标签
48 </textarea>
49 <textarea name="" id="" cols="30" rows="10">这是第二个textarea标签
50 </textarea>
51 <!-- 4.元素显示模式转换
52     display 取值 block（转换成块级标签）
53     inline-block（转换成行内块元素）
54     inline（转换成行内块元素）
55 -->
56 <hr>
57 </body>
58 </html>

```

css的三大特性，继承性，层叠性，优先性

对于权重，简单来说就是限定越少的权重就越大，其次就是数量越多的权重越大

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>css的三大特性</title>
8   <style>
9     /* 1.继承性的介绍
10      特性：子元素有默认继承父元素样式的特点（子承父业）
11      可以继承的常见属性（文字控制属性都可以继承）
12      1.color
13      2.font-style、font-weight、font-size、font-family

```

3.text-indent、text-align

4.line-height

5. ...

注意点：可以通过调试工具判断样式是否可以继承

子级自身的效果优先级高于继承的

*/

/* 2.层叠性，即覆盖，后来的覆盖前边的，

特性：

1.给同一个标签设置不同的样式→此时样式会层叠叠加→会共同作用在标签上

2.给同一个标签设置相同的样式→此时样式会层着覆盖→最终写在最后的样式会生效

效

注意点：

当样式冲突时，只有当选择器优先级相同时才能通过层叠性判断结果

*/

/*3.优先级

特性：

不同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样式

优先级公式：

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

注意点：

1.!important写在属性值的后面，分号的前面！

2.!important不能提升继承的优先级，只要是继承优先级最低！

3.实际开发中不建议使用 !important。

*/

/* css的权重叠加

不同的css选择器，权重的数值不一样，这些权重也可以相加。

下面是不同选择器的权重数值

标签选择器 0, 0, 0, 1

每增加一个标签，数值加一，如：

```
p{
    font-weight: 700;
}
```

```
h1 p{
    font-weight: 500;
}
```

第一个选择器p的权重数值为0, 0, 0, 1

第二个选择器h1 p的权重数值为0, 0, 0, 2 这是因为增加了一个标签选择器，所以数值也加一

类选择器（伪类选择器和类选择器的权重数值一样） 0, 0, 1, 0

和上面一样，每增加一个类选择器，也会加一，如

```
.p{
    font-weight: 700;
}
.h1 .p{
    font-weight: 500;
}
.h1 .p p{
    font-weight: 200;
}
```

第一个选择器.p的权重数值为0, 0, 1, 0

第二个选择器.h1 .p的权重数值为0, 0, 2, 0

```

67     不同选择器的权重数值也可以相加，如
68     第三个选择器.h1 .p p的权重数值为0，0，2，1
69     不过需要注意的是，权重数值不可进位，
70     如标签选择器到达10个不可进一位，应该这样写0，0，1，10
71     id选择器 0，1，0，0          权重叠加和上面的一致
72     行内样式 1，0，0，0
73     !important 无穷大
74     还有一点需要注意，继承的样式权重为0
75     <style>
76         .h1{
77             color: red;
78         }
79         p{
80             color: green;
81         }
82     </style>
83 </head>
84 <body>
85     <h1 class="h1">
86         <p>666</p>
87     </h1>
88 </body>
89 </html>
90     如上面，
91     虽然.h1的权重大，但是是继承到p的，所以权重是0，0，0，0          而p的权重是
92     0，0，0，1
93     因此是第二个选择器p生效。666是绿色的
94     */
95 </style>
96 </head>
97 <body>
98 </body>
99 </html>

```

css盒子模型

在网页布局时我们需要一种模块化的布局思想，添加盒子用以划分区域，盒子其实就是标签，常用的有div和span

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>盒子模型</title>
8      <style>
9          /* 1. 盒子内容 */
10         .one{
11             width: 200px;
12             height: 200px;
13             background-color: pink;
14         }
15         /* 2. 盒子边框 border（复合属性）粗细 线条种类 颜色

```

```

16         solid 实线
17         dashed 虚线
18         dotted 点线
19         拓展: 只为盒子某个方向单独设置边框 border-方位名词
20         注意: border会撑大盒子
21     */
22     .two{
23         width: 200px;
24         height: 200px;
25         background-color: pink;
26         border-bottom: 5px dashed #000;
27     }
28     /* 3.盒子内边距 padding (可视为复合属性)
29         padding最多可以跟四个值
30         四个值: 上 右 下 左
31         三个值: 上 左右 下
32         两个值: 上下 左右
33         注意: padding会撑大盒子
34     */
35     .three{
36         width: 200px;
37         height: 200px;
38         background-color: pink;
39         padding: 20px;
40     }
41     /* 4.盒子外边距 margin (可视为复合属性)
42         margin最多可以跟四个值
43         四个值: 上 右 下 左
44         三个值: 上 左右 下
45         两个值: 上下 左右
46         注意: margin会撑大盒子 */
47     .four{
48         width: 200px;
49         height: 200px;
50         background-color: pink;
51         margin: 20px;
52     }
53 </style>
54 </head>
55 <body>
56 <div class="one">
57     这是第一个div标签
58 </div>
59 <hr>
60 <div class="two">
61     这是第二个div标签
62 </div>
63 <hr>
64 <div class="three">
65     这是第三个div标签
66 </div>
67 <hr>
68 <div class="four">
69     这是第四个div标签
70 </div>

```



```
71 </body>
72 </html>
```

css3盒子模型自动内减

盒子都用两个属性，分别是margin（盒子外边距）和padding（盒子内边距），而在设计盒子时，盒子所占区域是由盒子的宽高加内外边距相加而得，与平常思维相悖，于是添加盒子的内减属性可以使其和平常思维相同，盒子的大小由设置的宽高决定

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>css3盒子模型自动内减</title>
8    <style>
9      /* 在css2中盒子模型的边框border和内边距padding都会撑大盒子
10       但在css3中引入了 box-sizing:border-box;不会撑大盒子 */
11      /* 1.css2盒子模型 */
12      .one{
13        width: 200px;
14        height: 200px;
15        background-color: pink;
16        border: 5px dashed #000;
17        padding: 20px;
18      }
19      .two{
20        width: 200px;
21        height: 200px;
22        background-color: pink;
23        border: 5px dashed #000;
24        padding: 20px;
25        box-sizing: border-box;
26      }
27    </style>
28  </head>
29  <body>
30    <div class="one">这是第一个盒子模型，完全使用css2</div>
31    <hr>
32    <div class="two">这是第二个盒子模型，使用了css3</div>
33  </body>
34  </html>
```

css清除盒子的默认样式

浏览器常会给部分标签设置默认的padding 和 margin，并且有时不同的浏览器还并不相同，因此我们常需要自己手动清除这些默认样式

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```

6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>css清除盒子的默认样式</title>
8   <style>
9       /* 浏览器常会给部分标签设置默认的padding 和 margin
10          我们常需要自己手动清除这些默认，后续自行添加
11          常用写法: *{
12
13              margin: 0;
14              padding: 0;
15          }
16
17      */
18      *{
19          margin: 0;
20          padding: 0;
21      }
22  </style>
23 </head>
24 <body>
25
26 </body>
</html>

```

css版心居中

我们常需要将盒子设置在浏览器的中间，这就是版心居中常用代码。margin: 0 auto;

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>css版心居中</title>
8      <style>
9          /* 版心指网页有效内容
10             常用写法如下 */
11      div{
12          width: 1000px;
13          height: 200px;
14          background-color: pink;
15          margin: 20px auto;
16      }
17  </style>
18
19 </head>
20 <body>
21     <div>版心</div>
22 </body>

```

css盒子模型行内元素垂直内外边距

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>

```

```

4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>盒子模型行内元素垂直内外边距</title>
8   <style>
9       /* 行内元素的上下方向内外边距不会生效，可将其视为一个框，而非块
10          |      |      —
11      框 |      |      块 |      |
12          |      |      —
13      */
14      span{
15          background-color: pink;
16          margin: 100px;
17          /* 要想实现行内元素标签增加边距可通过增加行高 */
18          line-height: 100px;
19      }
20  </style>
21 </head>
22 <body>
23   <span>这是一个行内元素标签span</span>
24   <span>这是一个行内元素标签span</span>
25
26 </body>
27 </html>

```

css结构伪类选择器

结构伪类选择器，可以根据元素在文档中所处的位置，来动态选择元素，从而减少HTML文档对ID或类class的依赖，有助于保持代码干净整洁。

```

1   <!DOCTYPE html>
2   <html lang="en">
3   <head>
4       <meta charset="UTF-8">
5       <meta http-equiv="X-UA-Compatible" content="IE=edge">
6       <meta name="viewport" content="width=device-width, initial-scale=1.0">
7       <title>结构伪类选择器</title>
8       <style>
9           /* 选中第一个li */
10          li:first-child{
11              background-color: red;
12          }
13          /* 选中最后一个 */
14          li:last-child{
15              background-color: blue;
16          }
17          /* 以下两种的小括号中可以填写公式
18              2n 取偶数
19              2n+1 取奇数
20              -n+5 取前五个
21              n+5 取五个后边的 */
22          /* 选中任意一个 */
23          li:nth-child(4){
24              background-color: green;
25          }

```

```

26     li:nth-last-child(2){
27         background-color: black;
28     }
29 </style>
30 </head>
31 <body>
32 <!-- ul>li{这是第$个li}*8 -->
33 <ul>
34     <li>这是第1个li</li>
35     <li>这是第2个li</li>
36     <li>这是第3个li</li>
37     <li>这是第4个li</li>
38     <li>这是第5个li</li>
39     <li>这是第6个li</li>
40     <li>这是第7个li</li>
41     <li>这是第8个li</li>
42 </ul>
43
44
45 </body>
46 </html>

```

css伪元素

概念：CSS 在渲染文档的时候，伪元素可以通过 css 给 HTML 添加一个元素（叫标签也行），这个元素在文档树中是找不到的。伪元素被当做 CSS 的样式来进行展现，用法和普通的元素用法是一样的。

特点：

优点：

不占用 DOM 节点，减少 DOM 节点数。

让 CSS 帮助解决了一部分 JavaScript 问题，简化了开发。

避免增加毫无意义的页面元素。

缺点：

不利于调试。

伪元素不真正在文档内容中体现，只在视觉效果上体现，所以不能给伪元素添加具有实际意义的内容，这部分内容不会搜索引擎抓取。

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8     <style>
9         .father{
10             width: 300px;
11             height: 300px;
12             background-color: pink;
13         }
14         .father::before{
15             /* 内容

```

```

16         content属性必须存在
17         生成的默认为行内元素*/
18         color: blue;
19         content: "老鼠";
20         width: 100px;
21         height: 100px;
22         background-color: green;
23         display: block;
24     }
25     .father::after{
26         /* 内容 */
27         content: "大米";
28     }
29 </style>
30 </head>
31 <body>
32     <!-- 伪元素 通过css创建标签，用于装饰 生成的默认为行内元素
33         找一个标签作为父级 在父级标签内创建-->
34     <div class="father">爱</div>
35 </body>
36 </html>

```

css浮动

浮动的详细内容请参考: [浮动详解](#)

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>浮动</title>
8     <style>
9         div{
10             width: 100px;
11             height: 100px;
12             background-color: pink;
13             display: inline-block;
14         }
15     </style>
16 </head>
17 <body>
18     <!-- 在浏览器解析行内块或行内标签的时候，如果两标签换行书写会产生一个空格间距
19         浮动用于解决该问题-->
20     <div>这是第一个div标签</div>
21     <div>与上一个换行书写</div>
22     <hr>
23     <div>这是第一个div标签</div><div>与上一个同一行书写</div>
24 </body>
25 </html>

```

css浮动体验

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <!-- 浮动现在用于布局网页 -->
9   <style>
10     img{
11       /* 过去浮动用于网页布局 */
12       float: left;
13     }
14     div{
15       width: 200px;
16       height: 200px;
17     }
18     .one{
19       background-color: blue;
20       float: left;
21     }
22     .two{
23       background-color: pink;
24       float: left;
25     }
26   </style>
27 </head>
28 <body>
29   <!-- 1. 图文环绕 -->
30   
31   二位好肉欸发我i发货欧维hi哦额分为i发货wife我黑哦ihf哦第三个是的士大夫大师傅士大夫瑟夫色的士大夫晒等发士大夫算法色服务费啊啊个人得分散发顺丰啊啊额发违法违法二发违法阿尔菲安哇分啊法文俄文法文发而非我啊万丰奥威二位好肉欸发我i发货欧维hi哦额分为i发货wife我黑哦ihf哦第三个是的士大夫大师傅士大夫瑟夫色的士大夫晒等发士大夫算法色服务费啊啊个人得分散发顺丰啊啊额发违法违法二发违法阿尔菲安哇分啊法文俄文法文发而非我啊万丰奥威二位好肉欸发我i发货欧维hi哦额分为i发货wife我黑哦ihf哦第三个是的士大夫大师傅士大夫瑟夫色的士大夫晒等发士大夫算法色服务费啊啊个人得分散发顺丰啊啊额发违法违法二发违法阿尔菲安哇分啊法文俄文法文发而非我啊万丰奥威二位好肉欸发我i发货欧维hi哦额分为i发货wife我黑哦ihf哦第三个是的士大夫大师傅士大夫瑟夫色的士大夫晒等发士大夫算法色服务费啊啊个人得分散发顺丰啊啊额发违法违法二发违法阿尔菲安哇分啊法文俄文法文发而非我啊万丰奥威二位好肉欸发我i发货欧维hi哦额分为i发货wife我黑哦ihf哦第三个是的士大夫大师傅士大夫瑟夫色的士大夫晒等发士大夫算法色服务费啊啊个人得分散发顺丰啊啊额发违法违法二发违法阿尔菲安哇分啊法文俄文法文发而非我啊万丰奥威
32   <hr>
33   <div class="one">这是第一个div</div>
34   <div class="two">这是第二个div</div>
35 </body>
36 </html>

```

css浮动特点

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Document</title>
8    <style>
9      /* 浮动会使元素脱离标准流，在标准流不占位置
10         浮动元素比标准流高半个级别，可以覆盖标准流中的元素，但不能覆盖内容
11         浮动找浮动，浮动一起排，下一个浮动的元素会在上一个浮动的后边排
12         浮动的标签 顶对其 */
13      .one{
14        width: 200px;
15        height: 200px;
16        background-color: pink;
17        float: left;
18      }
19      .two{
20        width: 300px;
21        height: 300px;
22        background-color: blue;
23        float: left;
24      }
25      .three{
26        width: 400px;
27        height: 400px;
28        background-color: red;
29      }
30      /* css的书写顺序
31         1.浮动/display
32         2.盒子模型
33         3.文字样式 */
34    </style>
35  </head>
36  <body>
37    <div class="one">one</div>
38    <div class="two">two</div>
39    <div class="three">three</div>
40  </body>

```

css清除浮动

清除浮动原因及原理请参考：[清除浮动](#)

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Document</title>
8    <style>
9      /* 清除浮动
10         1.直接加高度

```

```

11      2.加标签法 clear:both 给父元素内容的最后添加一个块级元素
12          给添加的块级元素设置clear:both
13          缺点：添加了标签，使结构更加复杂
14
15      3.单伪元素法
16      .clearfix::after{
17          content: '';
18          display:block;
19          clear:both;
20      }
21      4.双伪元素清除法
22      .clearfix::before,
23      .clearfix::after {
24          content: '';
25          display: table;
26      }
27
28      .clearfix::after {
29          clear: both;
30      }
31      5.给父级标签加overflow:hidden;
32      */
33      </style>
34      </head>
35      <body>
36
37      </body>
38      </html>

```

css定位-相对定位

定位可以使盒子放在我们想要的位置

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8      <style>
9          /* 相对定位
10          1.占有原来的位置
11          2.仍然具有标签原有的显示模式特点
12          3.改变位置参照自己原来的位置
13          */
14          .box{
15              position: relative;
16              left: 100px;
17              top:200px;
18              /* bottom: 200px;
19              right: 100px;
20              注意：以left top 为准*/
21
22              width: 200px;
23              height: 200px;

```



```

24     background-color: pink;
25 }
26
27 </style>
28 </head>
29 <body>
30 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
31 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
32 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
33 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
34 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
35 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
36 <div class="box">box</div>
37 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
38 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
39 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
40 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
41 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
42 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
43 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
44 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
45 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
46
47 </body>
48 </html>

```

css定位-绝对定位

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Document</title>
8 <style>
9     /* 绝对定位
10     先找已经定位的父级，如果有这样的父级，则以该父级为参照物
11     父级没有定位则以父级为参照物
12     1.脱标，不占位

```

```
13 2.改变了标签显示特点，具备了行内块的特性
14 3.建议子绝父相
15 4.绝对定位查找父级，逐层向外查找
16 */
17 .box{
18     position: absolute;
19     left: 0;
20     top: 0;
21     /* 注意：以left top 为准*/
22
23     width: 200px;
24     height: 200px;
25     background-color: pink;
26 }
27
28 .father{
29     width: 400px;
30     height: 400px;
31     background-color: pink;
32 }
33 .son{
34     position: relative;
35     width: 200px;
36     height: 200px;
37     background-color: skyblue;
38 }
39 .sun{
40     position: absolute;
41     /* left: 20px;
42     top: 30px; */
43
44     right: 20px;
45     bottom: 20px;
46
47     width: 100px;
48     height: 100px;
49     background-color: green;
50 }
51
52
53 </style>
54 </head>
55 <body>
56 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
57 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
58 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
59 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
60 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
61 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
```

```

62 <div class="box">box</div>
63 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
64 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
65 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
66 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
67 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
68 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
69 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
70 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
71 <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
72 <div class="father">
73     <div class="son">
74         <div class="sun">
75
76         </div>
77     </div>
78 </div>
79
80 </body>
81 </html>

```

css定位-居中

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8     <style>
9         .box{
10             /* 绝对定位的盒子不能使用margin: auto;居中 */
11             position: absolute;
12             left: 50%;
13             /* margin-left: -100px; */
14             top: 50%;
15             /* margin-top: -100px; */
16             transform: translate(-50%,-50%);
17
18             width: 200px;
19             height: 200px;
20             background-color: pink;
21
22         }
23     </style>

```

```
24 </head>
25 <body>
26   <div class="box"></div>
27 </body>
28 </html>
```

css定位-固定定位

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <style>
9     .box{
10       /* 固定定位
11        1.脱标-不占位置
12        2.改变位置参考浏览器的窗口
13        3.具有行内块特点
14        z-index: 数字;决定显示层级
15        */
16       position: fixed;
17       left: 0;
18       top: 0;
19
20       width: 200px;
21       height: 200px;
22       background-color: pink;
23     }
24   </style>
25 </head>
26 <body>
27   <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
28   <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
29   <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
30   <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
31   <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
32   <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
33   <div class="box">box</div>
34   <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
35   <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
36   <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
37   <p>相对定位1. 占有原来的位置2. 仍然具有标签原有的显示模式特点3. 改变位置参照自己原来的位置</p>
```

[illegible]

```
65 <p>相对定位1.占有原来的位置2.仍然具有标签原有的显示模式特点3.改变位置参照自己原来的位置</p>
66 <p>相对定位1.占有原来的位置2.仍然具有标签原有的显示模式特点3.改变位置参照自己原来的位置</p>
67 <p>相对定位1.占有原来的位置2.仍然具有标签原有的显示模式特点3.改变位置参照自己原来的位置</p>
68 </body>
69 </html>
```

css装饰-垂直

某些时候我们会发现标签并不会按照我们想的那样去对其，这是因为浏览器识别行内和行内块标签按照文字识别

文字按照基线对齐，那么什么是基线对齐？



因此有时会出现对不齐的情况，



这时我们需要设置 `vertical-align: top;`



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Document</title>
8 <style>
9     /* 浏览器识别行内和行内块标签按照文字识别
10     文字按照基线对齐
11     vertical-align: ;取值top middle bottom
12     */
```

```

13     div{
14         background-color: pink;
15     }
16     input{
17         vertical-align: top;
18     }
19 </style>
20 </head>
21 <body>
22     <div>
23         <input type="text">
24     </div>
25 </body>
26 </html>

```

css装饰-光标

用于更改鼠标的样式，通常不需要更改

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8     <style>
9         /* 改变光标样式
10         cursor 取值 default 小手
11                     text 工字形
12                     move 十字型 可移动 */
13     </style>
14 </head>
15 <body>
16
17 </body>
18 </html>

```

css装饰-边框圆角

用于使边框变得圆滑

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8     <style>
9         /* 使盒子四个角变得圆滑
10         border-radius */
11     <div>
12         margin: 100px auto;
13         width: 200px;

```

```

14         height: 200px;
15         background-color: pink;
16         /* 左上 右上 右下 左下 */
17         border-radius: 10px 40px 80px 100px;
18     }
19     .one{
20         margin: 400px auto;
21         width: 50px;
22         height: 50px;
23         background-color: pink;
24         /* 取值最高50% */
25         border-radius: 50%;
26     }
27 }
28 </style>
29 </head>
30 <body>
31     <div></div>
32     <div class="one"></div>
33 </body>
34 </html>

```

css装饰-溢出显示

当我们不限定盒子的大小时，盒子的大小会由内容撑开，但是当盒子的大小限时，盒子的内容超出时，我们需要将其限定成我们需要的样子。

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8      <style>
9          div{
10             /* 溢出
11             overflow
12             取值visible 默认，溢出部分可见
13                 hidden 溢出部分隐藏
14                 scroll  无论是否溢出，均显示滚动条
15                 auto   根据是否溢出，显示滚动条
16             */
17             overflow: auto;
18             width: 200px;
19             height: 200px;
20             background-color: pink;
21         }
22     </style>
23 </head>
24 <body>
25     <div>

```


[illegible]

CSS装饰-元素本身的隐藏

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Document</title>
8    <style>
9      div{
10        width: 200px;
11        height: 200px;
12        background-color: pink;
13      }
14      .one{
15        /* 占位隐藏 */
16        visibility: hidden;
17      }
18      .two{
19        /* 不占位隐藏 */
20        display: none;
21      }
22
23      body:hover .two{

```

```

24         display: block;
25     }
26 </style>
27 </head>
28 <body>
29     <div>这是一个div</div>
30     <div class="one">这是one</div>
31     <div>这是一个div</div>
32     <div class="two">这是two</div>
33     <div>这是一个div</div>
34 </body>
35 </html>

```

css装饰-元素整体透明度

将元素变得透明从而使用户有更高级的体验

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8      <style>
9          /* opacity */
10         div{
11             opacity: 0.5;
12             width: 200px;
13             height: 200px;
14             background-color: pink;
15         }
16     </style>
17 </head>
18 <body>
19     <div>这个字和背景色是不是有点透明？ </div>
20 </body>
21 </html>

```

css精灵图

- 什么是精灵图？

精灵图是单个图像中包含的图像集合。

包含许多图像的网页可能需要很长时间才能加载，同时会生成多个服务器请求。

使用精灵图将减少服务器请求的数量并节约带宽。

- 例子：

我们使用以下单幅图像（"navsprites.gif"）而不是使用三幅单独的图像：



通过使用 CSS，我们可以仅显示所需图像的某个部分。

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <style>
9     .phone{
10       display: inline-block;
11       width: 27px;
12       height: 35px;
13       background-color: pink;
14       background-image: url(./img/55-css精灵图.jpg);
15       background-position: -11px -7px;
16     }
17   </style>
18 </head>
19 <body>
20   <!-- 简单来说就是为了提升用户打开网页的速度，于是将多个小图合为一张大图方便传输
21       用户的图标等只取这个大图的一小部分，这样的图就叫精灵图 -->
22   <!-- 精灵图的使用步骤
23       1. 创建一个盒子，设置盒子的尺寸和小图尺寸相同(一般精灵图都用行内标签)
24       2. 将精灵图设置为背景图
25       3. 修改背景图位置(通过PxCook测量小图片左上角坐标，分别取负值设置给盒子的
26       background-position:x y;) -->
27   <span class="phone"></span>
28 </body>
29 </html>

```

css背景图缩放

当我们所拥有的图片和设计的盒子大小不匹配，我们就需要将背景图进行缩放等操作

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8   <style>
9     .box{
10       width: 400px;
11       height: 300px;
12       background-color: pink;
13       /* 背景图不重复平铺 */
14       background-repeat: no-repeat;
15       background-image: url(./img/56-css背景图缩放.jpg);
16       /* background-size: 300px; */
17       /* background-size: 50%; */
18       /* 一般在工作中盒子和图的比例是一样的，所以contain cover的效果相同 */
19       /* background-size: contain; */
20       background-size: cover;
21     }

```

```

22     }
23 </style>
24 </head>
25 <body>
26   <div class="box"></div>
27 </body>
28 </html>

```

css盒子阴影

给盒子添加阴影效果

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Document</title>
8    <style>
9      .box{
10         width: 200px;
11         height: 200px;
12         background-color: pink;
13         margin: 100px auto;
14         /* 水平偏移量 垂直偏移量 模糊度 阴影扩大 阴影颜色 inset */
15         box-shadow: 5px 10px 20px 10px green inset;
16      }
17    </style>
18  </head>
19  <body>
20    <div class="box"></div>
21  </body>
22  </html>

```

css过渡

SS 过渡允许您在给定的时间内平滑地改变属性值。

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Document</title>
8    <style>
9      /* 过渡的作用：让元素的样式慢慢变化，长配合hover使用，增强网页的交互体验 */
10     /* transition
11        过渡的属性 all(所有可以过渡的属性都将过渡)，具体属性名(写了那个属性哪个属
12        性能过渡)
13        过度的时长 数字+s(秒)
14        注意：过渡的效果加给默认样式中，鼠标移入移出都会有效果
15        加给hover只有鼠标移入才有效果,简单来说就是给hover加，鼠标移出时很突兀
16     */

```

```

15     .box{
16         width: 200px;
17         height: 200px;
18         background-color: pink;
19
20         transition: width 1s,height 1s;
21
22     }
23
24     .box:hover{
25         width: 600px;
26         height: 400px;
27         background-color: blue;
28         /* transition: width 1s,height 1s; */
29
30     }
31 </style>
32 </head>
33 <body>
34     <div class="box"></div>
35 </body>
36 </html>

```

css搜索引擎优化

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9  <body>
10     <!-- SEO: 优化网站在搜索引擎的排名
11         常见方法:
12         1. 竞价排名
13             前端能做的:
14         2. 将网页做成html后缀
15         3. 标签语义化
16         4. ... -->
17     <!-- title
18         desription
19         keywords
20         不要自己写, 有人准备好-->
21 </body>
22 </html>

```

标题图标

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">

```

```

5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0">
7 <title>Document</title>
8
9 <!-- <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
-->
10
11 </head>
12 <body>
13 <!-- 需要图片为ico -->
14 </body>
15 </html>

```

清除默认样式

在实际开发中，我们往往需要做一些准备工作，清楚默认样式就是一个准备工作

```

1  /* 清除浏览器默认样式 */
2  * {
3      margin: 0;
4      padding: 0;
5      /* 内减模式 */
6      box-sizing: border-box;
7  }
8
9  li {
10     /* 清除li自带的点 */
11     list-style: none;
12 }
13
14 a {
15     /* 清除链接自带的下划线 */
16     text-decoration: none;
17 }
18
19 /* 清除浮动 */
20 .clearfix::before,
21 .clearfix::after {
22     content: '';
23     display: table;
24 }
25 .clearfix::after {
26     clear: both;
27 }
28 body{
29     background-color: #f3f5f7;
30 }
31 /* 版心居中 */
32 .wrapper{
33     width: 1200px;
34     margin: 0 auto;
35 }

```