

HTML+CSS笔记

HTML笔记

1、初识HTML

- HyperTextMarkupLanguage (超文本标记语言)
- <body>、</body>等成对的标签，分别叫做开放标签和闭合标签，
- 单独呈现的标签（空元素），如<hr/>意为用/来关闭空元素。
- html注释：<!--注释内容-->

```
1  <!--DOCTYPE: 告诉浏览器使用什么规范 (默认是html) -->
2  <!DOCTYPE html>
3  <!-- 语言 zh中文 en英文 -->
4  <html lang="zh">
5  <!-- head标签代表网页头部 -->
6  <head>
7  <!-- meta 描述性标签, 表示用来描述网站的一些信息 -->
8  <!-- 一般用来做SEO -->
9      <meta charset="UTF-8">
10     <meta name="keywords" content="hyx的java学习">
11     <meta name="description" content="一起来学习java">
12     <!-- 网站标题 -->
13     <title>Title</title>
14 </head>
15 <!-- body代表主体 -->
16 <body>
17 Hello World!
18 </body>
19 </html>
```

2、网页基本标签

- 标题标签
- 段落标签
- 换行标签
- 水平线标签
- 字体样式标签
- 注释和特殊符号

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>基本标签</title>
6  </head>
7  <body>
8  <h1>一级标签</h1><!-- 标题标签 -->
9  <h2>二级标签</h2>
10 <h3>三级标签</h3>
11 <h4>四级标签</h4>
12 <h5>五级标签</h5>
13 <h6>六级标签</h6>
14
15 <!-- 段落标签 -->
16 <p>p换行1</p>
17 <p>p换行2</p>
18
19 <hr/><!-- 水平线标签 -->
20
```

```

21 换行1 <br/><!-- 换行标签-->
22 换行2 <br/>
23
24 <!-- 换行标签比较紧凑，段落标签有明显段间距-->
25 <!-- 粗体 斜体-->
26 <h1>字体样式标签</h1>
27 粗体: <strong>I love you </strong><br/>
28 斜体: <em>I love you </em><br/>
29 <!-- 特殊符号-->
30 空格: 1&nbsp;2&nbsp;3&nbsp;4<br/>
31 空格: 1 2 3 4<br/>
32 大于号&gt;<br/>
33 小于号&lt;<br/>
34 版权符号&copy;<br/>
35 <!-- 特殊符号记忆: &开头;结尾, 只要在idea中&敲出后就有提示-->
36 </body>
    </html>

```

- 图像标签

- 链接标签

href: 必填, 表示要跳转到那个页面

target: 表示窗口在那里打开

_blank: 在新标签中打开

_self: 在自己的网页中打开

```

<!DOCTYPE html>
<html lang="en">
1  <head>
2      <meta charset="UTF-8">
3      <title>图像和链接标签</title>
4  </head>
5  <body>
6  <!--src: 资源地址
7      相对地址, 绝对地址
8      ../上级地址
9      alt: 在图片加载失败的时候, 就会用文字代替
10     title: 鼠标悬停在图片上时, 所显示的名字
11     width height 图片的高和宽-->
12  
13  <br/>
14  <!--href: 跳转页面的地址
15     a标签内文字: 即显示的文字
16     可以把图片放在a标签内, 点击图片跳转网页
17     target: 表示在哪打开新网页_self: 当前标签打开 _blank: 新的页面中打开-->
18  <a href="https://www.baidu.com" target="_blank" title="123">你xxxx不会百度吗</a>
19  <br/>
20  <a href="https://www.baidu.com"></a>
21
22  <!-- 锚链接
23     1. 需要一个标记锚
24     2. 跳转到标记
25     #页面内跳转-->
26  <a name="top"></a>
27  <a href="#top">回到顶部</a>
28  <br/>
29  <!-- 也可以在网址后添加#号跳到对应网站的对应位置-->
30  <a href="https://www.baidu.com#down">百度底部</a> <br/>
31
32  <!-- 功能性链接
33  邮箱链接: mailto
34  qq链接
35  -->
36  <a href="mailto:xxxxxxqq.com">点击联系我</a>
37  <a target="_blank"
38     href="http://wpa.qq.com/msgrd?v=xxx&uin=&site=qq&menu=yes"/>
39     <a target="_blank"
40

```

```

41 |
42 |
43 | href="http://wpa.qq.com/msg?d=v=xxx&uin=&site=qq&menu=yes">
44 | 
46 | </body>
47 | </html>

```

行内元素和块元素

- 块元素

- 无论内容多少，该元素独占一行

1 | 例如：<p></p><hr/> <h1>...<h6>

- 行内元素：内容撑开宽度，左右都是行内元素的可以排在同行

1 | 例如：<a>

3、列表标签

什么是列表

列表就是信息资源的一种展示形式。它可以使信息结构化和条理化，并以列表的样式显示出来，以便浏览者能更快捷地获得相应的信息。

列表的分类：

- 无序列表

```

1 | <ol>
2 |   <li>1</li>
3 |   <li>2</li>
4 |   <li>3</li>
5 | </ol>

```

- 有序列表

```

1 | <ul><!-- 无序列表 -->
2 |   <li>123
3 |     <ul>
4 |       <li>1</li>
5 |       <li>2</li>
6 |       <li>3</li>
7 |     </ul>
8 |   </li>
9 |   <li>2</li>
10 |  <li>3</li>
11 | </ul>

```

- 定义列表

```

1 | <!-- 自定义列表
2 | dl: 标签
3 | dt: 列表名称
4 | dd: 列表内容 -->
5 | <dl>
6 |   <dt>学科</dt>
7 |   <dd>语文</dd>
8 |   <dd>数学</dd>
9 |   <dd>英语</dd>
10 |  <dt>语言</dt>
11 |  <dd>中文</dd>
12 |

```

```
12 | <dd>英语</dd>
13 | <dd>日语</dd>
14 | </dl>
```

4、表格

表格的基本结构：

- 单元格
- 行
- 列
- 跨行
- 跨列

```
1 | <!DOCTYPE html>
2 | <html lang="en">
3 | <head>
4 |   <meta charset="UTF-8">
5 |   <title>表格</title>
6 | </head>
7 | <body>
8 |   <!-- 表格table
9 |   行 tr
10 |   列 td
11 |   -->
12 | <table border="1px">
13 |   <tr><!-- 跨列-->
14 |     <td colspan="3">1-1</td>
15 |   </tr>
16 |   <tr><!-- 跨行-->
17 |     <td rowspan="2">2-1</td>
18 |     <td>2-2</td>
19 |     <td>2-3</td>
20 |   </tr>
21 |   <tr>
22 |     <td>3-2</td>
23 |     <td>3-3</td>
24 |   </tr>
25 | </table>
26 | </body>
27 | </html>
```

5、视频和音频

- video
- audio

```
1 | <!DOCTYPE html>
2 | <html lang="en">
3 | <head>
4 |   <meta charset="UTF-8">
5 |   <title>媒体元素</title>
6 | </head>
7 | <body>
8 |   <!-- 视频
9 |     src 资源路径
10 |     controls 控制面板
11 |     autoplay 自动播放
12 |   -->
13 | <video src="xxx/xxx/xxx" controls autoplay></video>
14 |   <!-- 音频-->
15 | <audio src="xxx/xxx/xxx" controls autoplay></audio>
16 | </body>
17 | </html>
```

6、页面结构

元素名	描述
header	标题头部区域的内容（用于页面或者页面中的一块区域）
footer	标记脚部区域的内容（用于整个页面或页面的一块区域）
section	Web页面中的一块独立区域
article	独立的文章内容
aside	相关内容或应用
nav	导航类辅助内容

```
1 | <!DOCTYPE html>
2 | <html lang="en">
3 | <head>
4 |     <meta charset="UTF-8">
5 |     <title>页面结构</title>
6 | </head>
7 | <body>
8 | <!-- 页面头部 -->
9 | <header>
10 |     <h2>网页头部</h2>
11 | </header>
12 | <section>
13 |     <h2>网页主体</h2>
14 | </section>
15 | <footer>
16 |     <h2>网页脚部</h2>
17 | </footer>
18 | </body>
19 | </html>
```

7、iframe内联框架

```
1 | <iframe src="path" name="mainFrame" ></iframe>
```

- ifram标签，必须要有src属性即引用页面的地址
- 给标签加上name属性后，可以做a标签的target属性，即在内联窗口中打开链接

8、表单语法(重点)

from标签，action属性为所提交的目的地址，method选择提交方式
可以选择使用post或者get方式提交

- get效率高，但在url中可以看到提交的内容，不安全，不能提交大文件
- post比较安全且可以提交大文件

标签	说明
input标签	大部分表单元素对应的标签有text、password、checkbox、radio、submit、reset、file、hidden、image和button，默认为text，可以提交用户名、等等
select标签	下拉选择框
textarea标签	文本域

属性	说明
type	指定元素的类型。text、password、checkbox、radio、submit、reset、file、hidden、image和button，默认为text
name	指定表单元素的名称（提交时所对应的key）


```

42
43
44
45 <p><!-- 文本域-->
46     反馈:
47     <textarea name="text" id="10" cols="30" rows="10" >文本内容</textarea>
48 </p>
49 <p><!-- 文件域-->
50     <input type="file" name="files">
51     <input type="button" name="upload" value="上传">
52 </p>
53 <!-- 邮件: 会简单验证是否是邮箱地址
54     url: 会简单验证是否是网络地址
55     number: 数字验证-->
56 <p>邮箱: <input type="email" name="email">
57     url: <input type="url"></p>
58
59 <!-- 数字验证
60     max最大数量
61     min 最小数量
62     step 每次点击增加或减少的数量-->
63 <p>商品数量<input type="number" name="num" max="100" min="1" step="1"></p>
64 <!-- 滑块-->
65 <p>音量: <input type="range" min="0" max="100" name="voice" step="2"></p>
66 <!-- 搜索框-->
67 <p>搜索: <input type="search"></p>
68
69 <p><!-- 增强鼠标可用性-->
70     <label for="mark">你点我试试</label>
71     <input type="text" id="mark">
72 </p>
73 </form>
74 </body>
</html>

```

CSS3学习

1.css是什么

2.CSS怎么用（快速入门）

3.CSS选择器（重点 + 难点）

4.美化页面（文字、阴影、超链接、列表、渐变...）

5.盒子模型

6.浮动

7.定位

8.网页动画（特效）

1.什么是CSS

1.1、什么是CSS

Cascading Style Sheet 层叠样式表

CSS：表现（美化网页）

字体，颜色，边距，高度，宽度，背景图片，网页定位，网页浮动

1.2、发展史

CSS1.0

CSS2.0: DIV（块）+CSS，HTML与CSS结构分离的思想，网页变得简单，SEO

CSS2.1: 浮动，定位

CSS3.0: 圆角、阴影、动画...浏览器兼容性~

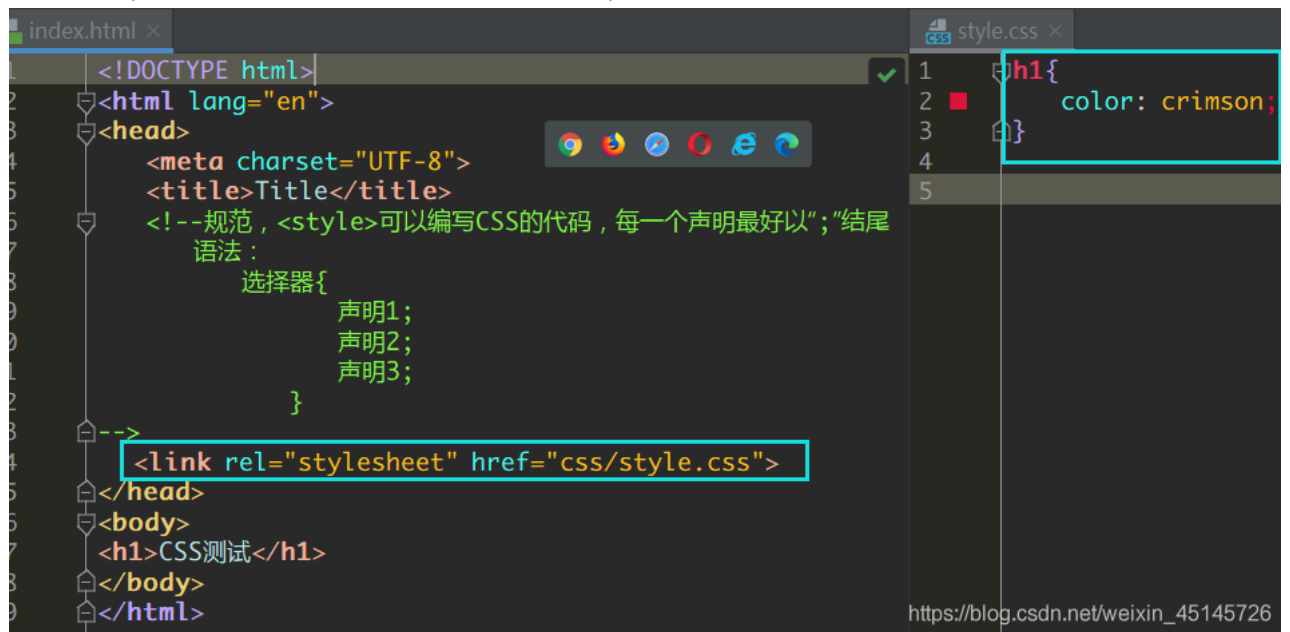
2、快速入门

2.1.1、练习格式：



```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4     <meta charset="UTF-8">
5     <title>Title</title>
6     <!-- 规范, <style>可以编写CSS的代码, 每一个声明最好以";"结尾
7         语法:
8             选择器{
9                 声明1;
10                声明2;
11                声明3;
12            } -->
13     <style>
14         h1{
15             color: crimson;
16         }
17     </style>
18 </head>
19 <body>
20     <h1>CSS测试</h1>
21 </body>
22 </html>
23
```

建议使用这种规范（单独写一个css文件，用link标签引入css文件效果）：



2.1.2、CSS的优势：

- 1、内容和表现分离；
- 2、网页结构表现统一，可以实现复用
- 3、样式十分的丰富
- 4、建议使用独立于html的css文件
- 5、利用SEO，容易被搜索引擎收录！

2.1.3、CSS的3种常用导入方式

```
<!DOCTYPE html>
<html lang="en">
<head>
1     <meta charset="UTF-8">
2     <title>Title</title>
3
```



```

4
5
6
7     <style><!-- 内部样式-->
8         h1{
9             color: green;
10        }
11    </style>
12
13    <!-- 外部样式-->
14    <link rel="stylesheet" href="css/style.css" />
15</head>
16<body>
17
18<!-- 优先级: 就近原则 行内样式 > 内部样式 > 外部样式-->
19<!-- 行内样式: 在标签元素中, 编写一个style属性, 编写样式即可-->
20<h1 style="color: red">这是标签</h1>
21</body>
22</html>

```

拓展：外部样式两种方法

- 链接式

html

```

1 <!-- 外部样式-->
2 <link rel="stylesheet" href="css/style.css" />

```

- 导入式

@import是CSS2.1特有的!

```

1 <!-- 导入式-->
2 <style>
3     @import url("css/style.css");
4 </style>

```

2.2选择器

作用：选择页面上的某一个后者某一类元素

2.2.1、基本选择器

1、标签选择器

选择一类标签

格式： 标签 { }

```

1 <head>
2     <meta charset="UTF-8">
3     <title>Title</title>
4     <style>
5         h1{
6             color: orange;
7             background: blue;
8             border-radius: 10px;
9         }
10        h3{
11            color: orange;
12            background: blue;
13            border-radius: 10px;
14        }
15        p{
16            font-size: 80px;
17        }
18    </style>
19</head>

```

```

20 <body>
21 <h1>标签选择器</h1>
22 <p>我爱学习</p>
23 <h3>学习JAVA</h3>
24 </body>

```

2、类选择器class

选择所有class一致的标签，跨标签

格式：.类名{}

```

1 <head>
2   <meta charset="UTF-8">
3   <title>Title</title>
4   <style>
5     /*类选择器的格式 .class的名称{}
6     好处：可以多个标签归类，是同一个class，可以复用*/
7     .demo1{
8       color: blue;
9     }
10    .demo2{
11      color: red;
12    }
13    .demo3{
14      color: aqua;
15    }
16  </style>
17 </head>
18 <body>
19 <h1 class = "demo1">类选择器: demo1</h1>
20 <h1 class="demo2">类选择器: demo2</h1>
21 <h1 class="demo3">类选择器: demo3</h1>
22 <p class="demo3">p标签</p>
23 </body>

```

3、id 选择器

全局唯一

格式：#id名{}

```

      <head>
        <meta charset="UTF-8">
        <title>Title</title>
        <style>
          /*id选择器: id必须保证全局唯一
          #id名称{}
          不遵循就近原则，优先级是固定的
          id选择器 > class类选择器 > 标签选择器
          */
          #demo1{
            color: red;
          }
          .demo2{
            color: green;
          }
          #demo2{
            color: orange;
          }
          h1{
            color: blue;
          }
        </style>
      </head>
      <body>
        <h1 id="demo1" class="demo2">id选择器: demo1</h1>
        <h1 class="demo2" id = "demo2">id选择器: demo2</h1>
        <h1 class="demo2">id选择器: demo3</h1>
        <h1 >id选择器: demo4</h1>
      </body>
    </html>

```

```

28 |
29 |
30 |     <h1>id选择器: demo5</h1>
    |     </body>

```

优先级: id > class > 标签

2.2.2、高级选择器

1. 层次选择器

- 后代选择器: 在某个元素的后面

```

1 | /*后代选择器*/
2 | <style>
3 | body p{
4 |     background:red;
5 | }
6 | </style>

```

- 子选择器, 一代

```

1 | /*子选择器*/
2 | <style>
3 | body>p{
4 |     background:orange;
5 | }
6 | </style>

```

- 相邻的兄弟选择器 同辈

```

1 | /*相邻兄弟选择器: 只选择一个, 相邻 (向下) */
2 | <style>
3 | .active+p{
4 | background: red
5 | }
6 | </style>
7 | <body>
8 |     <p class="active">p1<p>
9 |     <p>p2</p>
10 | </body>

```

- 通用选择器

```

1 | <style>
2 | /*通用兄弟选择器, 当前选中元素的向下的所有兄弟元素*/
3 | .active~p{
4 |     background:red;
5 | }
6 | </style>
7 | <body>
8 |     <p class="active">p1<p>
9 |     <p>p2</p>
10 | </body>

```

2. 结构伪类选择器

伪类

```

1 | <style>
2 |     ul li:first-child{/*ul的第一个子元素*/
3 |         background: aqua;
4 |     }
5 |     ul li:last-child{/*ul的最后一个子元素*/
6 |         background: blue;
7 |     }

```

```

8         }
9         /*选中p1: 定位到父元素, 选择当前的第一个元素
10         选择当前p元素 的父级元素, 选中父级元素的第一个, 并且是当前元素才生效! */
11         p:nth-child(1){
12             background: orange;
13         }
14         p:nth-of-type(2){/*选中父元素下的, 第2个p元素*/
15             background: red;
16         }
17         a:hover{
18             color: green;
19         }
20     </style>
21 </head>
22 <body>
23     <a href="">123</a>
24     <p>p1</p>
25     <p>p2</p>
26     <p>p3</p>
27     <h3>h3</h3>
28     <ul>
29         <li>1li1</li>
30         <li>1li2</li>
31         <li>1li3</li>
32     </ul>
33     <ul>
34         <li>2li1</li>
35         <li>2li2</li>
36         <li>2li3</li>
37     </ul>
38     <a href="www.baidu.com">百度</a>
</body>

```

3.属性选择器 (常用)

id + class结合

```

<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <style>
        .demo a{
            float: left;
            display: block;
            height: 50px;
            width: 50px;
            border-radius: 10px;
            background: aquamarine;
            text-align: center;
            color: gray;
            text-decoration: none;
            margin-right: 5px;
            /*line-height:50px;*/
            font: bold 20px/50px Arial;
        }
        /*属性名, 属性名=属性值 (正则)
        = 表示绝对等于
        *= 表示包含
        ^= 表示以...开头
        $= 表示以...结尾
        存在id属性的元素
        a[]{}
        */
        a[id]{
            background: yellow;
        }
        a[id=first]{/*id=first的元素*/
            background: green;
        }

        a[class*=links]{/*class 中有links的元素*/

```

```

33
34
35         background: bisque;
36     }
37
38     a[href^=http]{/*选中href中以http开头的元素*/
39         background: aquamarine;
40     }
41     a[href$=pdf]{/*选中href中以http开头的元素*/
42         background: aquamarine;
43     }
44 </style>
45 </head>
46 <body>
47     <p class="demo">
48         <a href="http:www.baidu.com" class="links item first" id="first">1</a>
49         <a href="" class="links item active" target="_blank" title="test">2</a>
50         <a href="images/123.html" class="links item">3</a>
51         <a href="images/1.png" class="links item">4</a>
52         <a href="images/1.jpg" class="links item">5</a>
53         <a href="abc" class="links item">6</a>
54         <a href="/a.pdf" class="links item">7</a>
55         <a href="/abc.pdf" class="links item">8</a>
56         <a href="abc.doc" class="links item">9</a>
57         <a href="abcd.doc" class="links item last">10</a>
58     </p>
59 </body>

```

效果：



3、美化网页元素

3.1、为什么要美化网页

1. 有效的传递页面信息
2. 美化网页，页面漂亮才能吸引客户
3. 凸显页面的主题
4. 提高用户的体验

span标签：重点要突出的字，使用span标签套起来

```

1 <head>
2     <meta charset="UTF-8">
3     <title>Title</title>
4     <style>
5         #title1{
6             font-size: 50px;
7         }
8     </style>
9 </head>
10 <body>
11 学习语言<span id="title1">JAVA</span>
12 </body>

```

3.2、字体样式

- font-family: 字体
- font-size: 字体大小
- font-weight: 字体粗细
- color: 字体颜色

```

1  <head>
2    <meta charset="UTF-8">
3    <title>Title</title>
4    <style>
5      body{
6        font-family:"Arial Black", 楷体;
7        color: red;
8      }
9      h1{
10       font-size: 50px;
11     }
12     .p1{
13       font-weight: 600;
14       color: gray;
15     }
16   </style>
17 </head>
18 <body>
19   <h1>标题</h1>
20   <p>正文11111</p>
21   <p class="p1">正文222222</p>
22   <p>i love study java</p>
23 </body>

```

常用写法:

```

1  font-weight:bolder;/*也可以填px, 但不能超过900, 相当于bolder*/
2  /*常用写法: */
3  font:oblique bolder 12px "楷体"

```

3.3、文本样式

1. 颜色→color:rgb / rgba()
2. 文本对齐方式→text-align: center
3. 首行缩进→text-indent: 2em
4. 行高→line-height: 300px;
5. 下划线→text-decoration

```

1  text-decoration:underline/*下划线*/
2  text-decoration:line-through/*中划线*/
3  text-decoration:overline/*上划线*/
4  text-decoration:none/*超链接去下划线*/

```

1. 图片、文字水平对齐

```

1  img,span{vertical-align:middle}

```

3.4、文本，阴影和超链接伪类

```

<style>
  a{/*超链接有默认的颜色*/
1    text-decoration:none;
2    color:#000000;
3  }
4  a:hover{/*鼠标悬浮的状态*/
5    color:orange;
6  }
7  a:active{/*鼠标按住未释放的状态*/
8    color:green
9  }
10 a:visited{/*点击之后的状态*/
11   color:red
12 }
13 a:link{
14   background: bisque;
15 }

```

```

15
16
17     }
18 </style>

```

阴影：

```

1  /* 第一个参数：表示水平偏移
2     第二个参数：表示垂直偏移
3     第三个参数：表示模糊半径
4     第四个参数：表示颜色
5  */
6  text-shadow:5px 5px 5px 颜色

```

3.6、列表ul li

主页index.html代码：

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6      <link href="css/style.css" rel="stylesheet" type="text/css">
7  </head>
8  <body>
9      <div id="nav">
10         <h2 class="title">全部商品分类</h2>
11         <ul>
12             <li>
13                 <a href="#">图书</a>
14                 <a href="#">音像</a>
15                 <a href="#">数字商品</a>
16             </li>
17             <li>
18                 <a href="#">家用电器</a>
19                 <a href="#">手机</a>
20                 <a href="#">数码</a>
21             </li>
22             <li>
23                 <a href="#">电脑</a>
24                 <a href="#">办公</a>
25             </li>
26             <li>
27                 <a href="#">家居</a>
28                 <a href="#">家装</a>
29                 <a href="#">厨具</a>
30             </li>
31             <li>
32                 <a href="#">服饰鞋帽</a>
33                 <a href="#">个性化妆</a>
34             </li>
35             <li>
36                 <a href="#">礼品箱包</a>
37                 <a href="#">钟表</a>
38                 <a href="#">珠宝</a>
39             </li>
40             <li>
41                 <a href="#">食品饮料</a>
42                 <a href="#">保健食品</a>
43             </li>
44             <li>
45                 <a href="#">彩票</a>
46                 <a href="#">旅行</a>
47                 <a href="#">充值</a>
48                 <a href="#">票务</a>
49             </li>
50         </ul>
51     </div>
52
53

```

```
-- |</body>
    </html>
```

css代码:

```
1  #nav{
2      width: 300px;
3      background: antiquewhite;
4  }
5  .title{
6      font-size: 18px;
7      font-weight: bold;
8      text-indent: 1em;/*缩进*/
9      line-height: 35px;
10     background: red;
11 }
12 /*ul li*/
13 /*
14 list-style:
15     non 去掉实心圆
16     circle 空心圆
17     square 正方形
18 */
19 /*ul{!*nav替换效果*!
20     background: antiquewhite;
21 }*/
22 ul li{
23     height: 30px;
24     list-style: none;
25     text-indent: 1em;
26 }
27 a{
28     text-decoration: none;
29     font-size: 14px;
30     color: black;
31 }
32 a:hover{
33     color: burlywood;
34     text-decoration: underline;
35 }
```

3.7、背景

1. 背景颜色: background
2. 背景图片

```
1  background-image:url("");/*默认是全部平铺的*/
2  background-repeat:repeat-x/*水平平铺*/
3  background-repeat:repeat-y/*垂直平铺*/
4  background-repeat:no-repeat/*不平铺*/
```

3.8、渐变

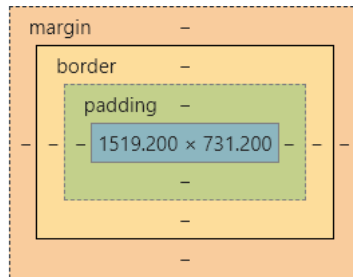
渐变背景网址: <https://www.grabient.com>

径向渐变、圆形渐变

```
1  body{
2      background-color: #4158D0;
3      background-image: linear-gradient(43deg, #4158D0 0%, #C850C0 46%, #FFCC70 100%);
4  }
```

4、盒子模型

4.1什么是盒子模型



1. margin: 外边距
2. padding: 内边距
3. border: 边框

4.2、边框

border: 粗细 样式 颜色

1. 边框的粗细
2. 边框的样式
3. 边框的颜色

4.3、外边距----妙用：居中

margin-left/right/top/bottom→表示四边，可分别设置，也可以同时设置如下

```

1 | margin:0 0 0 0/*分别表示上、右、下、左; 从上开始顺时针*/
2 | /*例1：居中*/
3 | margin:0 auto /*auto表示左右自动*/
4 | /*例2：*/
5 | margin:4px/*表示上、右、下、左都为4px*/
6 | /*例3*/
7 | margin:10px 20px 30px/*表示上为10px, 左右为20px, 下为30px*/

```

盒子的计算方式:

margin+border+padding+内容的大小

总结:

body总有一个默认的外边距 margin:0

常见操作：初始化

4.4、圆角边框----border-radius

```

1 | <style>
2 |     div{
3 |         width: 100px;
4 |         height: 100px;
5 |         border: 10px solid red;
6 |         /*一个border-radius只管一个圆的1/4*/
7 |         border-radius: 50px 20px 20px 30px;/*左上 右上 右下 左下 ,顺时针方向*/
8 |     }
9 | </style>

```

4.5、盒子阴影

```

1 | box-shadow: 10px 10px 1px black;

```

5、浮动

5.1标准文档流

块级元素：独占一行 h1~h6、p、div、列表...

行内元素：不独占一行 span、a、img、strong

注：行内元素可以包含在块级元素中，反之则不可以

5.2、display (重要)

1. block：块元素
2. inline：行内元素
3. inline-block：是块元素，但是可以内联，在一行

这也是一种实现行内元素排列的方式，但是我们很多情况用float

1. none：消失

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6   <!--block 块元素
7       inline 行内元素
8       inline-block 是块元素，但是可以内联，在一行
9   -->
10  <style>
11    div{
12      width: 100px;
13      height: 100px;
14      border: 1px solid red;
15      display: inline-block;
16    }
17    span{
18      width: 100px;
19      height: 100px;
20      border: 1px solid red;
21      display: inline-block;
22    }
23  </style>
24 </head>
25 <body>
26 <div>div块元素</div>
27 <span>span行内元素</span>
28 </body>
29 </html>
```

5.3、float: left/right左右浮动

clear: both

5.4、overflow及父级边框塌陷问题

clear:

right: 右侧不允许有浮动元素

left: 左侧不允许有浮动元素

both: 两侧不允许有浮动元素

none:

解决塌陷问题方案:

方案一: 增加父级元素的高度;

方案二: 增加一个空的div标签, 清除浮动

```
<div class = "clear"></div>
<style>
1   .clear{
2     clear:both;
3     margin:0;
4     padding:0;
5
6
```

```

7 |
8 |
  }
</style>

```

方案三：在父级元素中增加一个overflow属性

```

1 | overflow: hidden /*隐藏超出部分*/
2 | overflow: scroll /*滚动*/

```

方案四：父类添加一个伪类:after

```

1 | #father:after{
2 |     content:'';
3 |     display: block;
4 |     clear: both;
5 | }

```

小结：

1. 浮动元素增加空div----> 简单、代码尽量避免空div
2. 设置父元素的高度-----> 简单，但是元素假设有了固定的高度，可能就会超出范围
3. overflow-----> 简单，下拉的一些场景避免使用
4. 父类添加一个伪类:after（推荐）-----> 写法稍微复杂，但是没有副作用，**推荐使用**

display与float对比

1. display：方向不可以控制
2. float：浮动起来的话会脱离标准文档流，所以要解决父级边框塌陷的问题。

6、定位

6.1、相对定位

相对定位：position: relative;

相对于原来的位置，进行指定的偏移，相对定位的话，它仍然在标准文档流中！原来的位置会被保留

```

1 | top: -20px; /*向上偏移20px*/
2 | left: 20px; /*向右偏移20px*/
3 | bottom: 10px; /*向上偏移10px*/
4 | right: 20px; /*向左偏移20px*/

```

代码：

```

1 | <!DOCTYPE html>
2 | <html lang="en">
3 | <head>
4 |     <meta charset="UTF-8">
5 |     <title>相对定位</title>
6 |     <!-- 相对定位
7 |         相对于自己原来的位置进行偏移
8 |     -->
9 |     <style>
10 |         body{
11 |             padding: 20px;
12 |         }
13 |         div{
14 |             margin: 10px;
15 |             padding: 5px;
16 |             font-size: 12px;
17 |             line-height: 25px;
18 |         }
19 |         #father{
20 |             border: #ffa538 1px solid;
21 |             padding: 0;
22 |

```

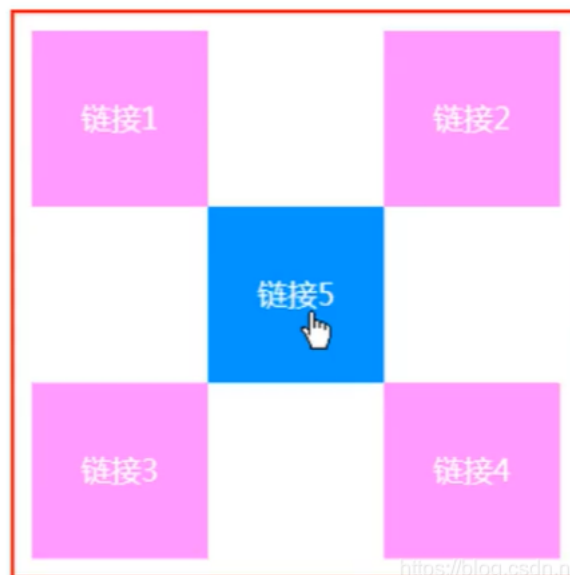
```

23     }
24     #first{
25         border: #b3ff38 1px solid;
26         background-color: #ffa538;
27         position: relative; /*相对定位: 上下左右*/
28         top: -20px; /*向上偏移20px*/
29         left: 20px; /*向右偏移20px*/
30     }
31     #second{
32         border: #427b11 1px solid;
33         background-color: #66c77f;
34     }
35     #third{
36         background-color: #b3ff38;
37         border: #38d7ff 1px solid;
38         position: relative;
39         bottom: 10px; /*向上偏移10px*/
40     }
41 </style>
42 </head>
43 <body>
44 <div id="father">
45     <div id="first">第一个盒子</div>
46     <div id="second">第二个盒子</div>
47     <div id="third">第三个盒子</div>
48 </div>
49 </body>
</html>

```

练习:

- 使用<div>和超链接<a>布局页面
- 每个超链接宽度和高度都是100px，背景颜色是粉色，鼠标指针移上去时变为蓝色
- 使用相对定位改变每个超链接的位置



https://blog.csdn.net/weixin_45145726

实现代码:

```

<style>
    #box{
1         height: 300px;
2         width: 300px;
3         border: 2px red solid;
4         padding: 10px;
5     }
6     a{
7         height: 100px;
8         width: 100px;
9         background-color: #ee73b7;
10        color: white;

```

```

11
12
13         text-align: center;
14         text-decoration: none;
15         line-height: 100px; /*设置行距100px*/
16         display: block; /*设置方块*/
17     }
18     a:hover{
19         background: #4158D0;
20     }
21     .a2{
22         position: relative;
23         left: 200px;
24         top: -100px;
25     }
26     .a4{
27         position: relative;
28         left: 200px;
29         top: -100px;
30     }
31     .a5{
32         position: relative;
33         left: 100px;
34         top: -300px;
35     }
36 </style>
37 </head>
38 <body>
39 <div id="box">
40     <div class="a1"><a href="" >连接1</a></div>
41     <div class="a2"><a href="" >连接2</a></div>
42     <div class="a3"><a href="" >连接3</a></div>
43     <div class="a4"><a href="" >连接4</a></div>
44     <div class="a5"><a href="" >连接5</a></div>
45 </div>
46 </body>

```

6.2、绝对定位-absolute和固定定位-fixed

定位：基于xxx定位，上下左右~

- 1、没有父级元素定位的前提下，相对于浏览器定位
- 2、假设父级元素存在定位，我们通常会相对于父级元素进行偏移
- 3、在父级元素范围内移动

总结：相对一父级或浏览器的位置，进行指定的偏移，绝对定位的话，它不在标准文档流中，原来的位置不会被保留

```

1 <head>
2     <meta charset="UTF-8">
3     <title>Title</title>
4     <style>
5         body{
6             height: 1000px;
7         }
8         div:nth-of-type(1){
9             width: 100px;
10            height: 100px;
11            background-color: red;
12            position: absolute; /*absolute 绝对定位*/
13            right: 0;
14            bottom: 0;
15        }
16        div:nth-of-type(2){
17            width: 50px;
18            height: 50px;
19            background-color: #b3ff38;
20            position: fixed; /*fixed 固定定位*/
21            right: 0;
22            bottom: 0;
23        }
24     </style>
25 </head>
26

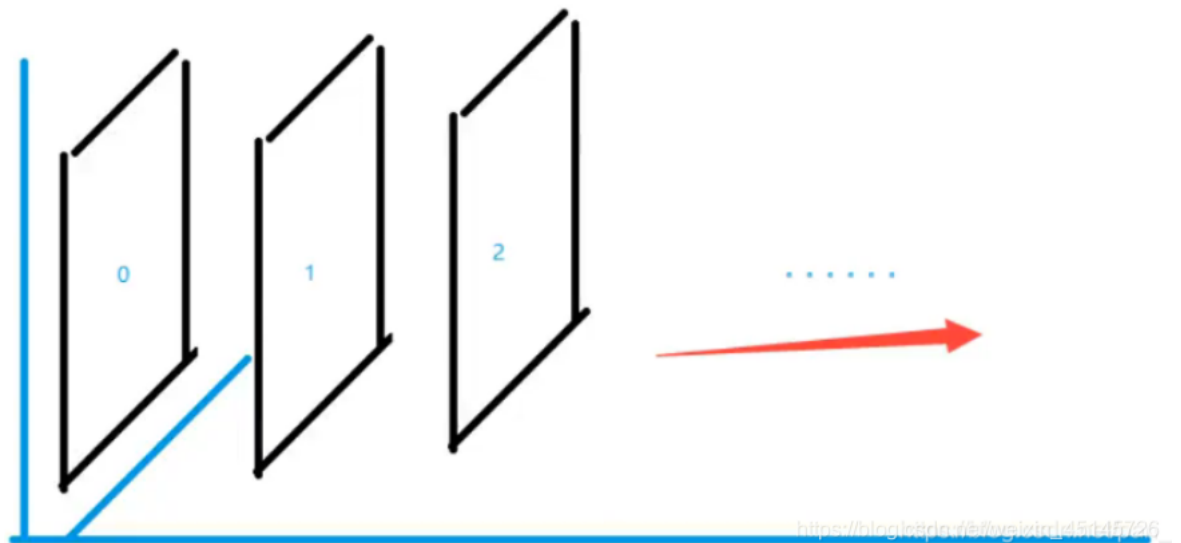
```

```

26 </body>
27 </div>div1</div>
28 </div>div2</div>
29 </body>

```

6.3、z-index



图层-z-index：默认是0，最高无限~999

index.html代码：

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Title</title>
6   <link rel="stylesheet" href="css/style.css" type="text/css">
7   <style></style>
8 </head>
9 <body>
10 <div id="content">
11   <ul>
12     <li></li>
13     <li class="tipText">学习微服务，找狂神</li>
14     <li class="tipBg"></li>
15     <li>时间：2099-01-01</li>
16     <li>地点：月球一号基地</li>
17   </ul>
18 </div>
19 </body>
20 </html>

```

css代码：

```

#content{
  width: 380px;
  padding: 0px;
  margin: 0px;
  overflow: hidden;
  font-size: 12px;
  line-height: 25px;
  border: 1px solid yellow;
}
ul,li{
  padding: 0px;
  margin: 0px;
  list-style: none;
}
/* 父级元素相对定位*/
#content ul{

```

```
16
17
18     position: relative;
19 }
20 .tipText,.tipBg{
21     position: absolute;
22     width: 380px;
23     height: 25px;
24     top:216px
25 }
26 .tipText{
27     color: white;
28     z-index: 999;
29 }
30 .tipBg{
31     background: orange;
32     opacity: 0.5;/*背景透明度*/
33     filter: alpha(opacity=50);
34 }
```

7、动画及视野拓展

css做动画过于繁琐，已有很多工具间接性做出

百度搜索canvas动画、卡斯基监控站（仅作了解）

CSS3

初识及入门



美化网页



盒子模型



