

CSS

01CSS 样式表

cascading style sheet 层叠样式表

简单说就是如何修饰网页信息的显示样式

用来展示XHTML或者XML等样式的计算机语言

02CSS语法

1. 每个CSS 样式由两部分组成，即选择符和声明符，声明又分为属性和属性值
2. 属性必须放在花括号中，属性与属性值用冒号链接
3. 每条声明用分号结束
4. 当一个属性有多个属性值的时候，属性值与实行值不分先后顺序，用空格隔开
5. 在书写样式过程中，空格、换行等操作

01内部样式

```
1  <head>
2      <meta charset="UTF-8">
3      <title>Lightly-HTML-Project</title>
4      <style>
5      h1{
6          color:red;
7      }
8      h2{
9          color:yellow;
10     }
11     h3{
12         color:blue;
13     }
14     </style>
15 </head>
16 <body>
17     <h1>11111111</h1>
18     <h2>22222222</h2>
19     <h3>33333333333333</h3>
20 </body>
```

02外部样式

2.1 link

属于XHTML标签，link引用的C S S会同时被加载

```

1 <head>
2     <meta charset="UTF-8">
3     <title>Lightly-HTML-Project</title>
4     <link rel="stylesheet" href="../css/02.1.css"
5         " type="text/css">
6 </head>

```

2.2 import

@import 完全是CSS提供的一种方式，@import 引用的CSS会等到页面全部被下载完再被加载

```

1
2 <head>
3     <meta charset="UTF-8">
4     <title>Lightly-HTML-Project</title>
5
6     <style>
7         @import url("../css/02.1.css");
8     </style>
9 </head>

```

03行内样式

style作为属性直接写在标签后面

```

1 <body>
2     <h1 style="color: gold;">11111</h1>
3     <h2 style="color: blue;">122222</h2>
4     <div style="width:200px;height:300px;color:gold;">我是div</div>
5 </body>
6
7

```

04样式表优先级

- !important优先级最高，
- 就近原则
- 优先级针对用一个标签，同一个属性，如果一个标签多了一个属性，那就会保留下来，

```

1 <head>
2     <meta charset="UTF-8">
3     <title>Lightly-HTML-Project</title>
4     <link rel="stylesheet" href="../css/02.1.css">
5     <style>
6         div{
7             color: hotpink;
8         }
9     </style>
10 </head>
11 <body>
12     <!--!important>行内>内部>外部--->
13     <div style="color: green;">人生苦短，我学Python</div>
14 </body>

```

!important放在哪，哪个优先级就最高

```
1  <head>
2    <meta charset="UTF-8">
3    <title>Lightly-HTML-Project</title>
4    <link rel="stylesheet" href="../css/02.1.css">
5    <style>
6      div{
7        color: hotpink!important;
8      }
9    </style>
10 </head>
11 <body>
12   <!--!important>行内>内部>外部--->
13   <div style="color: green;">人生苦短，我学Python</div>
14 </body>
```

03CSS选择器

要使用CSS对HTML页面中的元素实现一对一，一对多或者多对一的控制，就需要用到CSS选择器

3.1标签选择器

```
1  <head>
2    <meta charset="UTF-8">
3    <title>Lightly-HTML-Project</title>
4    <style>
5      div{
6        background-color: aqua;
7        color: blue;
8      }
9    </style>
10 </head>
11 <body>
12   <div>11111111</div>
13   <h1>222222222</h1>
14   <p>1222222131321321</p>
15   <div>78984464645</div>
16 </body>
```

3.2类选择器

一个div可以设置多个类名

看style标签里面的顺序，然后就近原则，显示的颜色是brown

```
1  <head>
2    <meta charset="UTF-8">
3    <title>Lightly-HTML-Project</title>
4    <style>
5      .ibm{
6        background-color: blueviolet;
7      }
```

```

8         .qianfeng{
9             color: brown;
10        }
11    </style>
12 </head>
13 <body>
14     <div class="ibm qianfeng">zhaosong1111111</div>
15     <div>zhaosng222222</div>
16     <div>zhaosong 3333333</div>
17     <div class ="ibm">zhaosng44444444</div>
18     <div class="ibm">zhaosng5555555555</div>
19     <div>zhaosong5666666</div>
20 </body>

```

3.3id选择器

一个id名称只能对应文档中的一个具体的元素对象（唯一性）

起名字要起英文，不能用关键字，（所有标记，和属性都是关键字）如：head标记

```

1    <head>
2        <meta charset="UTF-8">
3        <title>Lightly-HTML-Project</title>
4        <style>
5            #box{
6                background-color: burlywood;
7                color: blueviolet;
8            }
9            #box1{
10                background-color: chartreuse;
11            }
12        </style>
13    </head>
14    <body>
15        <div id="box1">人生苦短，我学Python</div>
16        <div id="box">三国演义</div>
17    </body>

```

3.4通配符选择器

*：含义就是所有的元素

```

1    <head>
2        <meta charset="UTF-8">
3        <title>Lightly-HTML-Project</title>
4        <style>
5            *{
6                /* 外边距*/
7                margin: 0;
8                padding: 0;
9                /*内边距 */
10            }
11        </style>
12    </head>

```

```
13 <body>
14     <h1>标题</h1>
15     <div>晒斑</div>
16     <ul>
17         <li>11111</li>
18         <li>22222</li>
19         <li>323333</li>
20     </ul>
21 </body>
```

3.5伪类选择器

主要应用于a链接上

语法：

```
1 a:link{属性:属性值;}超链接的起始状态
2 a:visited{属性:属性值;}超链接被访问后的状态
3 a:hover{属性:属性值;}鼠标悬停，即鼠标划过超链接时的状态
4 a:active{属性:属性值;}超链接被激活时的状态，即鼠标按下时超链接的状态
```

顺序：Link ----> Visited ----> Hover ----> Active

顺序错误会使样式失效

```
1 <head>
2     <meta charset="UTF-8">
3     <title>Lightly-HTML-Project</title>
4     <style>
5         /*未访问*/
6         a:link{
7             color: yellow;
8         }
9         /*访问之后*/
10        a:visited{
11            color: red;
12        }
13        /*鼠标移动到上面*/
14        a:hover{
15            color:orange;
16        }
17        /*点击，激活*/
18        a:active{
19            color: orchid;
20        }
21    </style>
22 </head>
23 <body>
24     <a href="https://www.baidu.com">百度</a>
25 </body>
```

3.6 群组选择器

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <title>Lightly-HTML-Project</title>
6   <style>
7   /*
8       div{
9         background-color: greenyellow;
10      }
11      h1{
12        background-color: greenyellow;
13      }
14      p{
15        background-color: greenyellow;
16      }*/
17
18      /* 群组选择器，提出公共代码，节约代码量 */
19      div,p,h1,.boxp{
20        background-color: greenyellow;
21      }
22  </style>
23 </head>
24 <body>
25   <div>1111111111</div>
26   <p>11111111111111</p>
27   <h1>111111111111</h1>
28   <div class="boxp">1212313212</div>
29 </body>
30 </html>
31
32
```

3.7 后代选择器

从右往左匹配

```
1 <head>
2   <meta charset="UTF-8">
3   <title>Lightly-HTML-Project</title>
4   <style>
5   /* 后代选择器， 包含选择器 */
6   /*从右到左匹配，先匹配p便签，在匹配包含p 的div标签*/
7   div p{
8     color:blue;
9     background-color: greenyellow;
10  }
11
12  p b{
13    color: palevioletred;
14  }
15 </style>
```

```
16     </head>
17     <body>
18         <div>1111111</div>
19         <p>
20             <b>4546545</b>
21         </p>
22         <div>
23             <p>1231314546545</p>
24         </div>
25
26     </body>
27
28
```

3.8选择器的权重

当多个选择器，选中的是同一个元素，且都为他们定义了样式，如果属性发生冲突，会**选择权重高的来执行**

选择器	权重，CSS中用四位数表示权重，权重表达方式如：0,0,0,0,
类型(元素)选择器	0001
Class选择器(类选择器)	0010
id选择器	0100
包含选择符	为包含选择符的权重之和 (div p)
内联样式	1000
!important	10000
CSS选择器解析规则1:	当不同选择符的样式设置有冲突的时候，高权重选择符的样式会覆盖低权重符的样式
CSS选择器解析规则2:	相同权重的选择符，样式遵循就近原则，哪个选择符最后定义，就选择哪个选择符样式。

04CSS属性

4.0CSS属性和属性值的定义

4.1CSS文本属性

属性	描述	说明
font_size	字体大小	单位是px,浏览器默认是16px，设计图常用的是12px

属性	描述	说明
font_family	字体	当字体是中文字体、英文字体中有空格时，需要加 双引号 ； 多个字体中间用 逗号 链接，先解析出第一个字体，如果没有解析第二个字体，以此类推
color	颜色	color:red; color:#ff0; color:rgb(255,0,0); 0-255
font-weight	加粗	font-weight:bloder(更粗的)/bold(加粗)/normal(常规) font-weight:100-900; 100-500不加粗 600-900加粗
font-style	倾斜	font-style:italic(斜体字)/oblique(倾斜的文字)/normal(正常显示)
text-align	文本水平对齐	text-align:left;水平靠左 text-align:right;水平靠右 text-align:center;水平居中 text-align:justify;水平两端对齐，但是只对多行起作用
line-height	行高	line-height的数据=height的数据，可实现单行文本垂直居中
text-indent	首行缩进	text-indent可以取负值；text-indent属性只对第一行起作用
letter-spacing	字间距	控制文字和文字之间的间距
text-decoration	文本修饰	text-decoration:none没有/underline下划线/overline上划线/line-through删除线
font	文件简写	font是font-style font-weight font-size /line-height font-family的缩写。
font:italic 800 30px/80px "宋体"; 顺序不能变，必须同时指定font-size和font-family属性时才起作用
text-transform	检索大小写	text-transform: capitalize;首字母大写 text-transform:lowercase;全部小写 text-transform:uppercase;全部大写 text-transform:none; 没效果



4.2CSS列表属性

属性	描述	说明
list-style-type	定义列表符合样式	list-style-type:disc;(实心圆)/circle(空心圆)/square(实心方块)/none(去掉符号)
list-style-image	将图片设置为列表符合样式	list-style-image:url();
list-style-position	设置列表项标记的放置位置	list-style-position:outside;列表的外面默认值 list-style-position:inside;列表的里面
list-style	简写	list-style:none url() inside ;

4.3CSS背景属性

属性	描述	说明
background-color	背景颜色	background-color:red; background-color:rgb(0,0,255); background-color:#fff;
background-image	背景图片	background-image:url();

属性	描述	说明
background-repeat	背景图片的平铺	background-repeat:no-repeat（不平铺）/repeat(平铺)/repeat-x(x轴平铺)/repeat-y(y轴平铺);
background-position	背景图片的定位	background-positon:水平位置, 垂直位置; 可以给负值 水平位置: left center right 垂直位置: top center bottom
background-attachment	背景图片的固定	background-attachment:scroll(滚动)/fixed(固定, 固定在浏览器窗口里面, 固定之后就相对于浏览器窗口了)
background-size	设置图片大小	background-size: 400px 400px 可以设置跟墙一般大 background-size: cover;将背景图片扩展至足够大, 以使背景图像完全覆盖背景区域, 可能有一部分会被裁掉 (cover 覆盖 contain包含) background-size:contain;把图片图像扩展到最大尺寸, 以使其宽度和高度完全适应内容区, 铺不满盒子, 留白
background	复合属性	background:yellow url() center fixed no-repeat 顺序随便 background-size只能单独用, 不能复合

4.4CSS边框属性

- 1 一 边框属性
- 2 1. 上边框:
- 3
- 4 border-top-style: 样式
- 5 border-top-width: 宽度
- 6 border-top-color: 颜色
- 7 border-top: 宽度, 样式, 颜色
- 8
- 9 2. 下边框:
- 10
- 11 border-bottom-style: 样式

12 border-bottom-width:宽度
13 border-bottom-color:颜色
14 border-bottom:宽度, 样式, 颜色
15
16 3. 左边框:
17
18 border-left-style: 样式
19 border-left-width:宽度
20 border-left-color:颜色
21 border-left:宽度, 样式, 颜色
22
23 4. 右边框:
24
25 border-right-style:样式
26 border-right-width:宽度
27 border-right-color:颜色
28 border-right: 样式, 宽度, 颜色
29
30 二:(1)设置边框样式(border-style)常用属性值如下
31 ·none:没有边框,即忽略所有边框的宽度(默认值)
32 ·solid:边框为单实线
33 ·dashed:边框为虚线
34 ·dotted:边框为点线
35 ·double:边框为双实线
36 ·border-top-style:上边框样式
37 ·border-right-style:右边框样式
38 ·border-bottom-style:下边框样式
39 ·border-left-style:左边框样式
40 ·border-style:上边框样式 右边框样式 下边框样式 左边框
41 ·border-style:上边框样式 左右边样式 下边框样式
42 ·border-style:上下边框样式 左右边框样式
43 ·border-style:上下边框样式
44
45 (2)设置边框宽度(border-width)
46
47 ·border-top-width:上边框宽度
48 ·border-right-width:右边框宽度
49 ·border-bottom-width: 下边框宽度
50 ·border-left-width:左边框宽度
51 ·border-width:上边框宽度[右边框宽度,下边框宽度,左边框宽度]
52
53 (3)设置边框颜色(border-color)
54
55 ·border-top-color:上边框颜色
56 ·border-right-color:右边框颜色
57 ·border-bottom-color:下边框颜色
58 ·border-left-color:左边框颜色
59 ·border-color:上边框颜色[右边框颜色 下边框颜色 左边框]
60
61 (4)综合设置边框
62
63 border:宽度 样式 颜色;
64 P{border-top:2px solid #ccc;}//定义上边框, 各个值顺序任意
65 例如将二级标题的边框设置为双实线,红色,3像素宽,代码如下: h2{border:3px double red;}
66 常用的复合属性有font-字体,border-边框,margin-外边距和background-背景等。

4.5 CSS浮动属性

- 浮动

属性	描述	说明
float	float:left;	元素靠左边浮动
float	float:right;	元素靠右边浮动
float	float:none;	元素不浮动

- 浮动的作用：

1. 定义网页中其他文本如何环绕该元素显示
2. 就是让竖着的东西横起来

- 清除浮动：

属性	描述	说明
clear	clear:none;	允许有浮动对象
clear	clear:right;	不允许右边有浮动对象
clear	clear:left;	不允许左边有浮动对象
clear	clear:both;	不允许有浮动对象

清除浮动只是改变元素 排序方式，该元素还是漂浮着 的，不占据文档位置。

05 盒子模型

盒模型是css布局的基石，它规定了网页元素如何显示以及元素之间相关关系，css定义所有的元素都可以拥有像盒子一样的外形和平面空间，即都包含边框、边界、补白、内容区，这就是盒模型

5.1 内边距padding

```

1  /* padding:30px;*/ /*内边距*/
2      /* padding:30px 30px;*/
3      padding-right: 10px;
4
5
6      /*
7      padding:
8      1个值，四个方向都一样
9      2个值，上下，左右
10     3个值，上 左右 下
11     4个值，上 右 下 左*/
12
13
14     /*padding-方向:top bottom left right*/

```

5.2边框border

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <title>Lightly-HTML-Project</title>
7    <style>
8      .box1 {
9        width: 100px;
10       height: 100px;
11       background: yellow;
12       border: 10px solid red
13     }
14
15     /*border 粗细 样式 颜色*/
16     /*样式: solid double dashed dotted*/
17
18     /*  border-top
19         border-bottom
20         border-left
21         border-right
22     */
23     .box2{
24       width: 100px;
25       height: 100px;
26       background: green;
27       border-top: 5px solid blue;
28       border-right: 10px dashed yellow;
29
30     }
31     .box3{
32       width: 100px;
33       height: 100px;
34       background: pink;
35       border-width:10px 20px 30px 40px ;
36       border-color: yellow tomato black blue;
37       border-style: solid dashed double dotted;
38
39       /*
40       border-width
41       border-style
42       border-color
43       */
44     }
45   </style>
46 </head>
47
48 <body>
49   <div class="box1"></div>
50   <div class="box2"></div>
51   <div class="box3"></div>
52 </body>
53
```

5.3外间距margin

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <title>Lightly-HTML-Project</title>
7    <style>
8      div {
9        width: 200px;
10       height: 200px;
11     }
12
13     .box1 {
14       background: red;
15       border: 1px solid yellow;
16       margin: 200px;
17       /*margin:50px 40px 20px 40px*/
18       /*
19         margin-top:10px;
20         margin-bottom:20;
21         margin-left:30px;
22         margin-right:40px;
23         */
24
25       float: left;
26     }
27
28     .box2 {
29       background: green;
30       border: 1px solid blue;
31       margin-left: 100px;
32       margin-bottom: 60px;
33     }
34
35     .box3 {
36       background-color: black;
37       margin: 0 auto; /*居中 上下是零, 左右自动*/
38     }
39   </style>
40 </head>
41
42 <body>
43   <div class="box1"></div>
44   <div class="box2"></div>
45   <div class="box3"></div>
46 </body>
47
48 </html>
```

5.3外间距特性问题

```
1 特性问题：
2 1. 兄弟关系，两个盒子垂直外边距与水平外边距问题？
3   - 垂直方向，外边距取最大值
4   - 水平方向，外边距会进行合并处理？
5 2. 父子关系，给子加外边距，但作用在父身上，怎么解决：
6
7   - 从子的margin-top===》父的padding-top，注意高度计算
8   - 给父盒子设置边框
9   - 加浮动
10  - overflow:hidden
11
```

• 兄弟关系

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <title>Lightly-HTML-Project</title>
7    <style>
8      /*
9      特性问题：
10     1. 兄弟关系，两个盒子垂直外边距与水平外边距问题？
11        - 垂直方向，外边距取最大值
12        - 水平方向，外边距会进行合并处理？
13     2. 父子关系，给子加外边距，但作用在父身上，怎么解决：
14     */
15
16
17     .box1,.box2,.box3,.box4{
18       width: 200px;
19       height: 200px;
20     }
21
22     .box1 {
23       background: red;
24       margin-bottom: 100px;
25     }
26     .box2 {
27       background: yellow;
28       margin-top: 50px;
29     }
30     .box3 {
31       float:left;
32       background: black;
33       margin-right: 50px;
34     }
35     .box4 {
36       float:left;
37       background: blue;
38       margin-left: 150px;
```

```

39     }
40
41     </style>
42 </head>
43
44 <body>
45     <div class="box1"></div>
46     <div class="box2"></div>
47     <div class="box3"></div>
48     <div class="box4"></div>
49
50 </body>
51
52 </html>

```

• 父子关系

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <title>Lightly-HTML-Project</title>
7      <style>
8          /*
9              1.从子的margin-top===》父的padding-top，注意高度计算
10             2.给父盒子设置边框
11             3.加浮动
12             4.overflow:hidden
13          */
14      .contain {
15          width: 500px;
16          height: 500px;
17          background-color: yellow;
18          /* padding-top:100px;*/
19          border: 1px solid transparent;
20
21      }
22
23      .box {
24          width: 200px;
25          height: 200px;
26          background-color: tomato;
27          margin-top: 100px;
28      }
29  </style>
30 </head>
31
32 <body>
33     <div class="contain">
34         <div class="box"></div>
35     </div>
36 </body>
37
38 </html>

```


06溢出

6.1溢出属性

属性	属性值
overflow	visible:默认值，溢出内容会显示在元素外面
overflow	hidden:溢出隐藏内容
overflow	scroll:滚动，溢出内容以滚动方式显示
overflow	auto:如果有溢出会添加滚动条，没有溢出则正常显示
overflow	inherit:规定应该遵从父元素继承overflow属性的值
overflow-x	x轴溢出
overflow-y	y轴溢出

```
1  <head>
2      <style>
3          div {
4              width: 200px;
5              height: 200px;
6              background: yellow;
7              /* overflow: visible; 显示溢出*/
8              /* overflow: hidden; 溢出隐藏，文本裁剪 */
9              /* overflow: scroll;不管有屋溢出，都会有滚动条 */
10
11              /* overflow: auto; 文本内容少就正常显示，内容多就出现滚动条 */
12
13              /* overflow: inherit;继承父元素的效果 */
14          }
15          .box{
16              /* overflow: auto; */
17              overflow-x:auto ;
18              overflow-y: hidden;
19          }
20      </style>
21  </head>
22
23  <body>
24      <div>
25          Lorem, ipsum dolor sit amet consectetur adipisicing elit. Quae
26          necessitatibus maxime error, tempore voluptatum
27          dignissimos, unde iste nam aliquid alias nostrum sapiente, ipsa
28          dolor exercitationem enim totam! Inventore,
29          accusamus optio.
30      </div>
31      <div class="box">
32          
33      </div>
34  </body>
```

6.2空余空间

该属性用来设置如何处理元素内的空白

属性	属性值
white-space	normal: 默认值, 空白会被浏览器忽略
	nowrap: 文本会在同一行上继续, 知道遇到 标签

```

1  /* nowrap : 不换行
2      pre : 显示空格, 回车, tab 不换行
3      pre-wrap: 显示空格, 回车, tab 换行
4      pre-line: 显示回车, 不显示换行
5      */

```

6.3省略号显示

```

1  text-overflow: clip/ellipsis;
2  clip: 默认值, 不显示省略号
3  ellipsis: 显示省略号
4

```

当单行文本溢出显示省略号需要同时设置以下声明:

1. 容器宽度: `width:200px;`
2. 强制文本在一行显示: `white-space:nowrap;`
3. 溢出内容为隐藏: `overflow:hidden;`
4. 溢出文本显示省略号: `text-overflow:ellipsis;`

07元素显示类型

元素类型的分类, 依据CSS的显示		
块元素(block element)	行内(内联)元素	行内块元素
<p>A) 块状元素在网页中就是以块的形式显示, 所谓块状就是元素显示为矩形区域,</p> <p>B) 默认情况下, 块状元素都会占据一行; 默认情况下, 块状元素会按顺序自上而下排列。</p> <p>C) 块状元素都可以定义自己的宽度和高度。</p> <p>D) 块状元素一般都作为其他元素的容器, 它可以容纳其它内联元素和其它块状元素。我们可以把这种容器比喻为一个盒子</p> <p>例如: <code>div p ul li ol li dl dt dd h1-h6</code>等</p>	<p>A) 内联元素的表现形式是始终以行内逐个进行显示; 横着排</p> <p>B) 内联元素没有自己的形状, 不能定义它的宽和高, 它显示的宽度、高度只能根据所包含内容的高度和宽度来确定, 它的最小内容单元也会呈现矩形形状;</p> <p>C) 内联元素也会遵循盒模型基本规则, 但是对于margin和padding个别属性值不生效</p> <p>例如: <code>a b em i span strong</code>等</p>	<p>内联块状元素 (inline-block) 就是同时具备内联元素、块状元素的特点</p> <p>例如: <code>img input</code>等</p>

块元素

```
1 display: block;
2 display: list-item;
3 p标签里放文本可以，不能放块级元素
```

行内元素

```
1 display: inline;
```

行内块元素

```
1 display: inline-block;
```

元素类型互相转换

```
1
```

08定位

定位 <u>position</u>					
个数	书写语法	说明	文档流	偏移位置 (top left right bottom) 时候的参照物	层叠顺序 (z-index)
1	position: static;	默认值	默认	默认	z-index属性是不带单位的，并且可以给负值，没有设置z-index时，最后写的对象优先显示在上层，设置后，数值越大，层越靠上；
2	position: absolute;	绝对定位	脱离	A) 当没有父元素或者父元素没有定位，参照物是浏览器窗口的 第一屏 B) 有父元素且父元素有定位， 父元素	
3	position: relative;	相对定位	不脱离	自己的初始位置	
4	position: fixed;	固定定位	脱离	浏览器的当前窗口	
5	position: sticky;	粘性定位	可以做吸顶效果，粘性定位是css3.0新增加的，兼容不好		

8.1静态定位

```
1 <style>
2     div{
3         width: 200px;
4         height: 200px;
5     }
6     .box1{
7         background-color: red;
8     }
9     .box2{
10        background-color: yellow;
11        position: relative;
12        top: 100px;
13        left: 100px;
14    }
15    .box3{
```

```

16         background-color: blue;
17     }
18 </style>
19 </head>
20 <body>
21     <div class="box1"></div>
22     <div class="box2"></div>
23     <div class="box3"></div>
24 </body>
25

```

8.2绝对定位，没有父元素

```

1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <title>Document</title>
9      <style>
10         * {
11             margin: 0;
12             padding: 0;
13         }
14
15         .box {
16             width: 200px;
17             height: 200px;
18             background-color: yellow;
19             position: absolute;
20             top: 100px;
21             left: 100px;
22         }
23     </style>
24 </head>
25
26 <body>
27
28     <div class="box"></div>
29     <div>Lorem ipsum, dolor sit amet consectetur adipisicing elit. At
aperiam repellat dolore totam mollitia, ullam
exercitationem repudiandae id atque vitae in qui a rLorem ipsum,
dolor sit amet consectetur adipisicing elit. At aperiam repellat dolore
totam mollitia, ullam
exercitationem repudiandae id atque vitae in qui a repellendus
quLorem ipsum, dolor sit amet consectetur adipisicing elit. At aperiam
repellat dolore totam mollitia, ullam
exercitationem repudiandae id atque vitae in qui a repellendus
quLorem ipsum, dolor sit amet consectetur adipisicing elit. At aperiam
repellat dolore totam mollitia, ullam
exercitationem repudiandae id atque vitae in qui a repellendus
quepellendus quod veritatis rem, expedita veniam officiis.
</div>
34

```

```
35 </body>
36
37 </html>
```

8.3绝对定位，有父元素

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta http-equiv="X-UA-Compatible" content="IE=edge">
7     <meta name="viewport" content="width=device-width, initial-scale=1.0">
8     <title>Document</title>
9     <style>
10         * {
11             margin: 0;
12             padding: 0;
13         }
14
15         .box {
16             width: 500px;
17             height: 500px;
18             background-color: yellow;
19             margin: 0 auto;
20             position: relative;
21         }
22
23         .child{
24             width: 200px;
25             height: 200px;
26             background-color: red;
27             left:100px;
28             top:100px;
29             position: absolute;
30         }
31     </style>
32 </head>
33
34 <body>
35
36     <div class="box">
37         <div class="child"></div>
38     </div>
39
40 </body>
41
42 </html>
```

8.4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             width: 100%;
11             height: 1500px;
12             background-color: yellow;
13         }
14         .ad{
15             width: 100px;
16             height: 200px;
17             background-color: red;
18             position: fixed;
19             right: 0;
20             bottom: 200px;
21         }
22     </style>
23 </head>
24 <body>
25     <div class="box"></div>
26     <div class="ad"></div>
27 </body>
28 </html>
```

8.5粘性定位

```
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         *{
10             margin: 0;
11             padding: 0;
12         }
13         .header{
14             background-color: yellow;
15             height: 100px;
16             width: 1000px;
17         }
18         .nav{
19             background-color: red;
20             width: 500px;
21             height: 50px;
```

```

22         margin: 0 auto;
23         position: sticky;
24         top:0px;
25     }
26     .body{
27         height: 1000px;
28         background-color: green;
29     }
30 }
31 </style>
32 </head>
33 <body>
34     <div class="header"></div>
35     <div class="nav"></div>
36     <div class="body"></div>
37 </body>
38 </html>

```

8.6定位的层级

- z-index:值越大，层级就越大，越靠上显示 可以设置负值

```

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          div{
10             width: 200px;
11             height: 200px;
12         }
13         .box1{
14             background-color: yellow;
15             position: relative;
16             left: 100px;
17             top:100px;
18             z-index: 1;/*层级*/
19         }
20         .box2{
21             background-color: red;
22             position: relative;
23             z-index: 100;
24         }
25         /* z-index值越大，层级就越大，越靠上显示 可以设置负值 */
26     </style>
27 </head>
28 <body>
29     <div class="box1"></div>
30     <div class="box2"></div>
31
32 </body>
33 </html>

```

09锚点

10精灵图

11宽高自适应

CSS3

一、CSS3的概念和优势

CSS3是css技术的升级版，CSS3语言开发是朝着模块化发展的。以前的规范作为一个模块实在是太庞大而且比较复杂，所以，把它分解为一些小的模块，更多新的模块也被加入进来。这些模块包括：盒子模型、列表模块、超链接方式、语言模块、背景和边框、文字特效、多栏布局等。

css3的优点：CSS3将完全向后兼容，所以没有必要修改现在的设计来让它们继续运作。网络浏览器也还将继续支持CSS2。对我们来说，CSS3主要的影响是将会使用新的可用的选择器和属性，这些会允许实现新的设计效果（譬如动态和渐变），而且可以很简单的设计出现在的设计效果（比如说使用分栏）

二、渐进增强和优雅降级

渐进增强 progressive enhancement:

针对低版本浏览器进行构建页面，保证最基本的功能，然后再针对高级浏览器进行效果、交互等改进和追加功能达到更好的用户体验。

优雅降级 graceful degradation:

一开始就构建完整的功能，然后再针对低版本浏览器进行兼容。

区别:

优雅降级是从复杂的现状开始，并试图减少用户体验的供给，而渐进增强则是从一个非常基础的，能够起作用的版本开始，并不断扩充，以适应未来环境的需要。降级（功能衰减）意味着往回看；而渐进增强则意味着朝前看，同时保证其根基处于安全地带。

层级选择器

```
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>li{
10              border: 1px solid red;
11          }
12          /*当前元素后边的第一个兄弟*/
13          /* .child+li{
14              background-color: yellow;
15          }
16          */
17          /*当前元素后面所有的亲兄弟*/
18          .child~li{
19              background-color: yellow;
20          }
21          .container+p{
22              background-color: blue;
23          }
24
25
26      </style>
27 </head>
28 <body>
```



```

29     <ul class="box">
30         <li>111
31             <ul>
32                 <li>1111-111</li>
33                 <li>22222-222222</li>
34                 <li>333-3333</li>
35             </ul>
36         </li>
37         <li class="child">222</li>
38         <li>333</li>
39         <li>444</li>
40         <li>555</li>
41     </ul>
42     <div class="container">1111</div>
43     <p>p-11111111</p>
44     <p>p-22222222</p>
45     <div>
46         <p>div-33333</p>
47     </div>
48 </body>
49 </html>

```

属性选择器

二、属性选择器

- 1、E[attr]: 只使用属性名，但没有确定任何属性值；□
- 2、E[attr="value"]: 指定属性名，并指定了该属性的属性值；□
- 3、E[attr~="value"]: 指定属性名，并且具有属性值，此属性值是一个词列表，并且以空格隔开，其中词列表中包含了一个value词，而且等号前面的“~”不能不写□

扩展知识

- 4、E[attr^="value"]: 指定了属性名，并且有属性值，属性值是以value开头的；□
- 5、E[attr\$="value"]: 指定了属性名，并且有属性值，而且属性值是以value结束的□
- 6、E[attr*="value"]: 指定了属性名，并且有属性值，而且属性值中包含了value；□

```

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          /* [class]{
10             background-color: yellow;
11         } */
12         /* div下所有的class */
13         div[class]{
14             background-color: blue;
15         }
16         p[class]{
17             background-color: rebeccapurple;
18         }
19         /*完全匹配*/
20         /* div[class=box1]{
21             border: 1px solid pink;

```

```

22         } */
23         /*包含就匹配*/
24         div[class~=box1]{
25             border: 1px solid pink;
26         }
27         input[name]{
28             background-color: yellow;
29         }
30         input[type=email]{
31             background-color: aqua;
32         }
33         /* 模糊匹配 */
34         /* class^=b 以b开头
35         class$b 以b结尾的
36         class*=b 包含某个字符 */
37         div[class^='b'],p[class*='1']{
38             color:white
39         }
40     </style>
41 </head>
42 <body>
43     <div class="box1">div1111</div>
44     <div class="box2">div222222222222</div>
45     <div >333333</div>
46     <div class="box1 box3"> div555555 </div>
47     <p class="p1">p-1111</p>
48     <p class="p2">p-222222</p>
49     <p>p-33363333</p>
50     <div>
51         用户名: <input type="text" name="username"><br>
52         密码:<input type="password" name="password"><br>
53         年龄: <input type="number" name="age"><br>
54         邮箱: <input type="email">
55     </div>
56 </body>
57 </html>

```

结构伪类选择器

三、伪类选择器

1、结构性伪类选择器

X:first-child 匹配子集的第一个元素。IE7就可以支持

X:last-child匹配父元素中最后一个X元素

X:nth-child(n)用于匹配索引值为n的子元素。索引值从1开始

X:only-child这个伪类一般用的比较少，比如上述代码匹配的是div下的有且仅有一个的p，也就是说，如果div内有多个p，将不匹配。

X:root匹配文档的根元素。在HTML（标准通用标记语言下的一个应用）中，根元素永远是HTML

X:empty匹配没有任何子元素（包括包含文本）的元素X

```

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: 940px;
11         height: 100px;
12         margin:0 auto;
13         background: yellow;
14     }
15     .box div{
16         float: left;
17         width: 300px;
18         height: 100px;
19         background-color: red;
20         margin-right: 20px;
21     }
22
23     .box div:last-child{
24         margin-right: 0;
25     }
26     /* li:first-child{
27         background-color: yellow;
28     }
29     li:last-child{
30         background-color: brown;
31     }
32     /*第几个*/
33     /* li:nth-child(2){
34         background-color: blueviolet;
35     } */
36     /*偶数 nth-child(2n),nth-child(even)*/
37     li:nth-child(2n){
38         background-color: red;
39     }
40     /*奇数 nth-child(2n-1),nth-child(odd)*/
41     li:nth-child(odd){
42         background-color: burlywood;
43     }
44     .icon1{
45         border: 1px solid black;
46     }
47     .icon1 p{
48         background-color: antiquewhite;
49     }
50     /*独生子女*/
51     .icon1 p:only-child{
52         color: blueviolet;
53     }
54     /*没有孩子的, */
55     .empty1:empty{
56         width: 200px;
57         height: 200px;
58         background-color: black;
59     }
```

```

60      /*根*/
61      :root,body{
62          background-color: black;
63      }
64  </style>
65  </head>
66  <body>
67      <div class="box">
68          <div></div>
69          <div></div>
70          <div class="last"></div>
71      </div>
72
73      <ul>
74          <li>1111</li>
75          <li>2222</li>
76          <li>3333</li>
77          <li>4444</li>
78          <li>5555</li>
79      </ul>
80
81      <div class="icon1">
82          <p>1111</p>
83          <p>2222</p>
84      </div>
85      <div class="icon1">
86          <p>1111</p>
87      </div>
88      <div class="empty1"></div>
89      <div class="root1"></div>
90  </body>
91  </html>
92

```

目标伪类选择器

2、目标伪类选择器

E:target 选择匹配E的所有元素，且匹配元素被相关URL指向

```

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          *{
10             margin:0;
11             padding: 0;
12         }
13         ul{
14             list-style: none;
15             position: fixed;
16             right: 0px;
17             top:100px
18         }
19         li{
20             width: 100px;
21             height: 50px;
22             line-height: 50px;
23             border: 1px solid black;
24         }
25         div{
26             height: 600px;
27             border: 1px solid #ccc;
28         }
29         div:target{
30             background-color: yellow;
31         }
32     </style>
33 </head>
34 <body>
35     <ul>
36         <li>
37             <a href="#a">京东秒杀</a>
38         </li>
39         <li>
40             <a href="#b">双11</a>
41         </li>
42         <li>
43             <a href="#c">频道优选</a>
44         </li>
45         <li>
46             <a href="#d">特色广场</a>
47         </li>
48     </ul>
49
50     <div id="a">京东秒杀</div>
51     <div id="b">双11</div>
52     <div id="c">频道优选</div>
53     <div id="d">特色广场</div>
54 </body>
55 </html>

```

UI元素状态伪类选择器

3、UI 元素状态伪类选择器

E:enabled 匹配所有用户界面（form表单）中处于可用状态的E元素

E:disabled 匹配所有用户界面（form表单）中处于不可用状态的E元素

E:checked 匹配所有用户界面（form表单）中处于选中状态的元素E

E:selection 匹配E元素中被用户选中或处于高亮状态的部分

```
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          input:enabled{
10              /* background-color: rebeccapurple; */
11          }
12          input:disabled{
13              background-color: yellow;
14          }
15          /*焦点会匹配此选择器*/
16          input:focus{
17              background-color: blue;
18          }
19          input[type=checkbox]{
20              /*去掉默认样式*/
21              appearance:none;
22              width: 20px;
23              height: 20px;
24              border: 1px solid black;
25          }
26          input:checked{
27              background-color: red;
28          }
29          div::selection{
30              background-color: yellow;
31              color: red;
32          }
33      </style>
34  </head>
35  <body>
36      <form action="">
37          用户名<input type="text" name="username" ><br>
38          密码<input type="password" name="password" ><br>
39          记住用户名<input type="checkbox"><br>
```

```

40         <input type="submit" disabled>
41
42     </form>
43     <div>Lorem ipsum dolor, sit amet consectetur adipisicing elit. Deserunt
dolorem reprehenderit beatae magni, quas numquam accusantium optio explicabo
corrupti aspernatur, fugiat, id repudiandae. Corrupti maiores recusandae eos
ex placeat molestiae.</div>
44 </body>
45 </html>

```

否定伪类选择器

4、否定伪类选择器

E:not(s) (IE6-8浏览器不支持:not()选择器。)

匹配所有不匹配简单选择符s的元素E

5、动态伪类选择器

E:link

链接伪类选择器

选择匹配的E元素，而且匹配元素被定义了超链接并未被访问过。常用于链接描点上

E:visited

链接伪类选择器

选择匹配的E元素，而且匹配元素被定义了超链接并已被访问过。常用于链接描点上

E:active

用户行为选择器

选择匹配的E元素，且匹配元素被激活。常用于链接描点和按钮上

E:hover

用户行为选择器

选择匹配的E元素，且用户鼠标停留在元素E上。IE6及以下浏览器仅支持a:hover

```

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          /* li:not(:first-child){
10              background-color: yellow;
11          } */
12          li:not(:nth-child(2n+1)){
13              background-color: blue;
14          }
15          div:not(:empty){
16              width: 200px;
17              height: 200px;
18              background-color: greenyellow;
19          }

```

```
20     </style>
21 </head>
22 <body>
23     <ul>
24         <li>1111</li>
25         <li>2222</li>
26         <li>3333</li>
27     </ul>
28     <div>
29         <ul><li>121</li></ul>
30     </div>
31     <div></div>
32 </body>
33 </html>
```

WebApp布局

渐变动画变形

Grid网格布局
