

## CSS

什么是CSS

定义CSS的两种方式

内联样式方式

外联样式方式

CSS选择器

ID选择器

元素选择器

类(class)选择器

属性选择器

基于关系的选择器

A E(后代元素选择器)

A>E(子元素选择器)

B+E(相邻兄弟选择器)

伪类

伪元素

选择器优先级别

背景

属性:

文本

属性

字体

属性

列表

属性

表格

属性

CSS常见布局

CSS盒子模型

# CSS

- 本课程将学习的内容
  - CSS基础入门：什么是CSS以及为什么要使用CSS
  - CSS基础语法：CSS声明、规则、引入方式和命名规范等
  - CSS选择器：选择器、伪元素、伪类和选择器的优先级别等
  - CSS样式：背景、文本、链接、列表和表格等
  - CSS布局：三行两列、两行三列

## 什么是CSS

- CSS指层叠样式表(Cascading Style Sheets)样式通常存储在样式表中
- CSS主要目的是 可以让相同的一个页面在不同浏览器中呈现相同的样式
- 样式定义如何显示HTML元素
- 是为了解决内容和表现分离的问题

- 内容： 内容主要指HTML中的元素
- 表现： 表现主要指内容在浏览器中呈现的样式

## 定义CSS的两种方式

### 内联样式方式

- ```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>内联样式</title>
7   </head>
8   <body>
9     <p style="color: lightcoral;font-weight: bolder;">这是一个段落内容</p>
10    <p>这又是一个段落内容</p>
11  </body>
12 </html>
```

- 仅对当前标签有效

### 外联样式方式

- ```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>外联样式</title>
7     <style>
8       p{
9         color: lightcoral;
10        font-weight: bolder;
11      }
12    </style>
13  </head>
14  <body>
15    <p>这是一个段落内容</p>
16    <p>这又是一个段落内容</p>
17  </body>
18 </html>
```

- 引用外联样式

```

1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title></title>
7     <link rel="stylesheet" type="text/css" href="css/04.css"/>
8   </head>
9   <body>
10    <p>这是一个段落内容</p>
11  </body>
12 </html>

```

```

1 p{
2   color: lightblue;
3   font-weight: bolder;
4 }

```

- 如本机中引用 css 文件，link 中的 href 应该是 "css/index.css"

注意：css文件与html文件的路径关系请参考gif图：

名称	修改日期	类型	大小
css	2016/6/2 10:55	文件夹	
index.html	2016/6/2 9:29	360 se HTML Do...	1 KB

- 当同时使用外联样式和内联样式时，优先级别：内联样式 > 外联样式

## CSS选择器

## ID选择器

- id属性的特点就是唯一不可重复的

- ```
1 #p{
2     color: lightblue;
3 }
```

## 元素选择器

- 元素选择器是通过元素名称来定位页面元素

- ```
1 span{
2     color: lightcoral;
3 }
```

## 类(class)选择器

- ```
1 .myclass{
2     color: lightblue;
3 }
```

## 属性选择器

- ```
1 [title] {
2     color: lightblue;
3 }
```

```
1 <!DOCTYPE html>
2 <html lang="en">
3     <head>
4         <meta charset="utf-8" />
5         <meta name="viewport" content="width=device-width, initial-scale=1">
6         <title>CSS选择器</title>
7         <style>
8             #p{
9                 color: lightblue;
10            }
11            .myclass{
12                color: lightblue;
13            }
14            span{
15                color: lightcoral;
16            }
17            [title] {
18                color: lightblue;
19            }
20        </style>
```

```

21     </head>AAA
22     <body>
23
24         <p id="p">id选择器</p>
25         <p class="myclass">类选择器</p>
26         <span>元素选择器</span>
27         <p title="this is p.">属性选择器</p>
28     </body>
29 </html>

```

## 基于关系的选择器

### A E(后代元素选择器)

- 元素A的任一后代元素E(后代节点值A的子节点，子节点的子节点，以此类推),称之为 祖先元素和后代元素
- 如

```

<body>
  <p>
    我是body的后代元素
    <span>我也是body的后代元素</span>
  </p>
</body>

```

- `<p><span>` 都是 `<body>` 的后代元素

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title>后代选择器</title>
7      <style>
8        span{
9          background-color: lightblue;
10       }
11       div span {
12         background-color: lightcoral;
13       }
14     </style>
15   </head>
16   <body>
17     <div>
18       <span>
19         span1
20         <span>span2</span>
21       </span>
22     </div>
23     <span>
24       span3
25     </span>

```

```
26     </body>
27 </html>
```

### A>E(子元素选择器)

- 元素A的任一子元素E(也就是直系后代), 称之为 父元素和子元素
- 如

```
<body>
  <p>
    我是body的后代元素也是子元素
    <span>我只是body的后代元素, 不是子元素</span>
  </p>
</body>
```

- `<p>` 是 `<body>` 的子元素也是后代元素, `<span>` 是 `<p>` 的子元素也是 `<p><body>` 的后代元素

- ```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1">
6      <title></title>
7      <style>
8        span{
9          background-color: blue;
10       }
11
12       div > span {
13         background-color: red;
14       }
15     </style>
16   </head>
17   <body>
18     <div>
19       <span>span1
20         <span>span2</span>
21       </span>
22     </div>
23     <span>span3</span>
24   </body>
25 </html>
```

### B+E(相邻兄弟选择器)

- 元素B的任一下一个兄弟元素E
- 如

```
<body>
  <p>我是span的兄弟元素</p>
  <span>我是p的兄弟元素</span>
</body>
```

- `<span><p>` 互为兄弟元素

- ```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title></title>
7     <style type="text/css">
8       #one + li{
9         color: blue;
10      }
11    </style>
12  </head>
13  <body>
14    <ul>
15      <li id="one">one</li>
16      <li>two</li>
17      <li>thredd</li>
18    </ul>
19  </body>
20 </html>
```

## 伪类

- `:link` :向未被访问的链接添加样式
- `:hover` :当鼠标悬浮在元素上方时, 向元素添加样式
- `:active` :向被激活的元素添加样式
- `:visited` :向已被访问的链接添加样式

## 伪元素

- 伪元素加两个:
- `::before` :会为当前元素创建一个子元素作为伪元素
- `::after` :用来匹配已选中元素的一个虚拟的最后子元素

- ```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title>伪元素</title>
```

```

7      <style type="text/css">
8          q::before{
9              content: "<<";
10             color: lightgreen;
11         }
12         q::after{
13             content: ">>";
14             color: red;
15         }
16     </style>
17 </head>
18 <body>
19     <q>一些引用</q>, 他说, <q>比没有好。</q>
20 </body>
21 </html>

```

## 选择器优先级别

- 内联样式的优先级别高于外联样式
- id选择器>类选择器>元素选择器>伪类

## 背景

### 属性:

- `background`: 简写属性, 作用是将背景属性设置在一个声明中
- `background-color`: 设置元素的背景颜色
- `background-image`: 把图像设置为背景
- `background-position`: 设置背景图像的起始位置
- `background-repeat`: 设置背景图像是否及如何重复

- ```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8" />
5          <meta name="viewport" content="width=device-width, initial-scale=1">
6          <title></title>
7          <style type="text/css">
8              #p1{
9                  background-color: greenyellow;
10             }
11             #p2{
12                 background-image: url("img/car.jpg");
13                 width: 200px;
14                 height: 200px;
15             }
16             #p3{
17                 background-image: url("img/car.jpg");
18                 widows: 200px;

```



```

19         height: 200px;
20         background-position: -60px -80px;
21     }
22     body{
23         background-image: url("img/car.jpg");
24         background-repeat: no-repeat;
25     }
26     </style>
27 </head>
28 <body>
29 <p id="p1">这是一个段落内容</p>
30 <p id="p2"></p>
31 <p id="p3"></p>
32 </body>
33 </html>

```

## 文本

### 属性

- `color`: 设置文本颜色
- `text-align`: 对齐元素中的文本
- `text-decoration`: 向文本添加修饰
- `text-indent`: 缩进元素中文本的首行
- `vertical-align`: 设置元素的垂直对齐

- ```

1 <!DOCTYPE html>
2 <html>
3     <head>
4         <meta charset="utf-8" />
5         <meta name="viewport" content="width=device-width, initial-scale=1">
6         <title>文本</title>
7         <style type="text/css">
8             #p1{
9                 color: lightcoral;
10            }
11            #p2{
12                text-align: left;
13            }
14            #p3{
15                text-align: center;
16            }
17            #p4{
18                text-align: right;
19            }
20            a{
21                text-decoration: none;
22            }
23            #p5{

```

```

24         text-indent: 50px;
25     }
26 </style>
27 </head>
28 <body>
29     <p id="p1">这是一个段落内容</p>
30     <p id="p2">这又是一个段落内容</p>
31     <p id="p3">这又是一个段落内容</p>
32     <p id="p4">这又是一个段落内容</p>
33     <a href="#">这是一个链接</a>
34     <p id="p5">这又是一个段落内容</p>
35 </body>
36 </html>

```

## 字体

### 属性

- `font` :在一个声明中设置所有的字体属性
- `font-family` :指定文本的字体系列
- `font-size` :指定文本的字体大小
- `font-style` :指定文本的字体样式
- `font-weight` :指定字体的粗细

- ```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8" />
5          <meta name="viewport" content="width=device-width, initial-scale=1">
6          <title></title>
7          <style type="text/css">
8              #p1{
9                  font-family: "times new roman";
10             }
11             #p2{
12                 font-family: "microsoft yahei";
13             }
14             #p3{
15                 font-size: large;
16                 font-style: italic;
17                 font-weight: bolder;
18             }
19         </style>
20     </head>
21     <body>
22         <p id="p1">this is text.</p>
23         <p id="p2">这是一个段落内容</p>
24         <p id="p3">这是一个段落内容</p>
25     </body>
26 </html>

```

## 列表

### 属性

- `list-style`: 用于把所有用于列表的属性设置于一个声明中
- `list-style-image`: 将图象设置为列表项标志
- `list-style-type`: 设置列表项标志的类型

- ```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6     <title></title>
7     <style type="text/css">
8       #u1{
9         list-style-type: decimal;
10      }
11      #u2{
12        list-style-image: url("img/car.jpg");
13      }
14    </style>
15  </head>
16  <body>
17    <ul id="u1">
18      <li>西瓜</li>
19      <li>苹果</li>
20      <li>香蕉</li>
21    </ul>
22    <ul id="u2">
23      <li>西瓜</li>
24      <li>苹果</li>
25      <li>香蕉</li>
26    </ul>
27  </body>
28 </html>
```

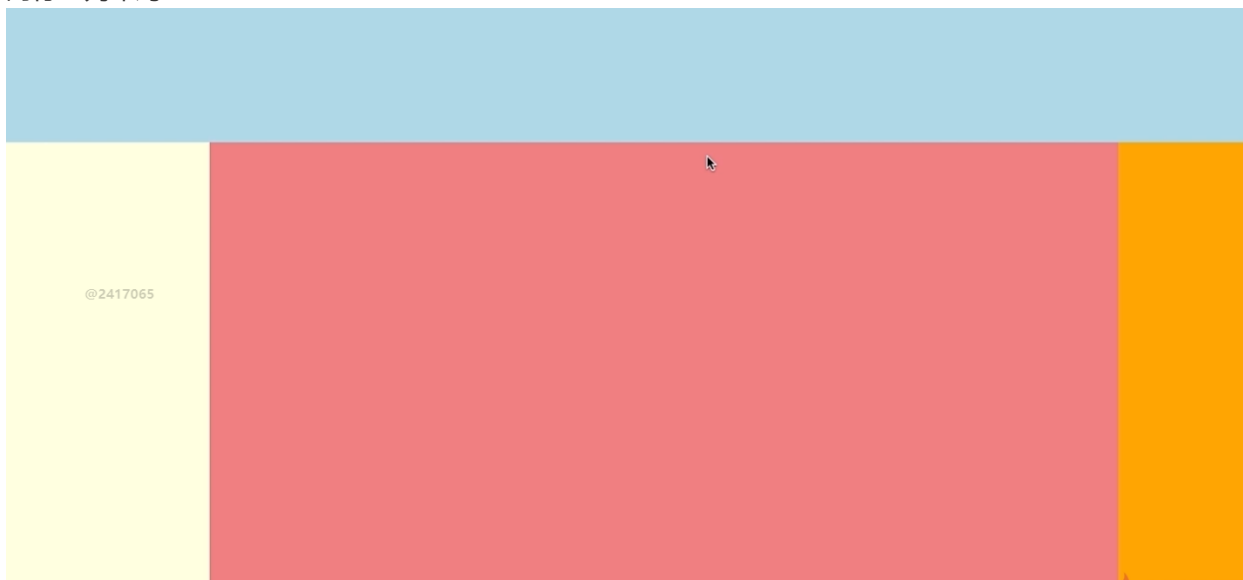
## 表格

### 属性

- `border`: 设置表格边框
- `border-collapse`: 设置表格的边框是否被折叠成一个单一的边框或隔开
- `width`: 定义表格的宽度
- `text-align`: 表格中的文本对齐
- `padding`: 设置表格中的填充

### CSS常见布局

- 两行三列布局



- 三行两列布局



## CSS盒子模型

-

# CSS 盒子模型

## CSS 盒子模型



•

# CSS 盒子模型

## 边框

属性	描述
border	简写属性，用于把针对四个边的属性设置在一个声明
border-style	用于设置元素所有边框的样式，或者单独地为各边设置边框样式
border-width	用于为元素的所有边框设置宽度，或者单独地为各边边框设置宽度
border-color	设置元素的所有边框中可见部分的颜色，或为 4 个边分别设置颜色

•

# CSS 盒子模型

## 内边距

属性	描述
padding	使用缩写属性设置在一个声明中的所有填充属性
padding-bottom	设置元素的底部填充
padding-left	设置元素的左部填充
padding-right	设置元素的右部填充
padding-top	设置元素的顶部填充

# CSS 盒子模型

## 外边距

属性	描述
margin	简写属性。在一个声明中设置所有外边距属性
margin-bottom	设置元素的下外边距
margin-left	设置元素的左外边距
margin-right	设置元素的右外边距
margin-top	设置元素的上外边距

•