# 3.css

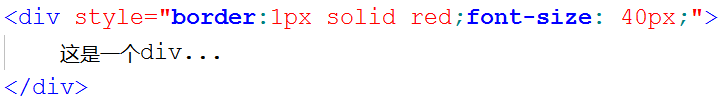
## 3.1.css概述

css：层叠样式表，可以用来美化html页面，就比如html是盖房子，而css则是装修。可以实现展示数据的html代码和设置样式的CSS代码的分离，增强了网页的显示能力！

## 3.2.在html页面中引入css样式的方式

1. **通过标签上的style属性引入css样式（案例1）**

大多数标签都可以具有style属性, 可以通过style属性为当前标签设置样式



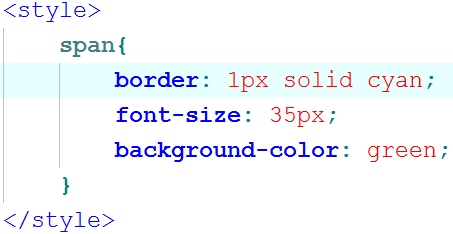
这种方式设置样式不推荐大量使用, 会造成页面结构的混乱, 不利于后期的维护及扩展.

**案例1.属性引入css**

|  |
| --- |
| <div style="border: 2px solid red; font-size: 30px; font-family: 微软雅黑;">div111</div>  <span style="border: 2px solid green; font-size: 30px; font-style: italic;">span111</span>  <p style="border: 2px solid blue; font-size: 30px; font-weight: bold;">p1111</p> |

1. **通过style标签引入css样式（案例2）**

在head标签内部, 可以提供一个style标签, 在style标签内部可以选中元素进行修饰!



这种方式是将所有的css代码放在一个style标签内部统一进行管理, 初步的实现了html代码和css代码的分离.

**案例2. \_css通过style标签引入css样式**

|  |
| --- |
| <style>  td{  text-align: left/right/center/justify;  }  </style> |

1. **通过链接引入外部的css文件（案例3）**

通过head标签下的link标签可以引入外部的css文件



这种方式是将所有的css代码放在一个独立的文件中进行统一管理, 真正的实现了html代码和css代码的分离.

**案例3.外部引入css**

|  |
| --- |
| css文件内容：  p{  border: 1px blue solid;  font-size:30px;  color:green;  }  头文件中引入css文档  <link rel="stylesheet" href="demo.css"/> |

## 3.3.选择器

选择器可以帮助我们在html中选中想要修饰的标签进行修饰

### 3.3.1.基本选择器（案例4）

(1) 标签名(元素名)选择器

通过标签的名称选中指定名称的标签进行样式的修饰

格式: 元素名{}, 如: span{}

<span></span>

<span></span>

(2) 类(class)选择器

通过标签上的class属性, 可以为标签设置所属的类, 所有class值相同的标签则为一类, 可以通过标签名选择器选中这一类的标签进行修饰.

格式: .类名{ css属性 }, 如: .c1{}

<div class=”c1”></div>

<p class=”c1”></p>

<span class=”c1”></span>

(3) ID选择器

通过标签上的id属性, 可以为标签设置所属的编号, id值的特点是在整个html中是独一无二的. 可以通过id值选中想要修饰的标签

格式: #id值{ css属性 }, 如: #d1{}

<div id=”d1”></div>

**案例4.基本选择器**

|  |
| --- |
| /\* 1.标签名选择器 \*/  span{  background-color: #DDA0DD;  font-size: 16px;  }  /\* 2.类选择器 \*/  .c1{  background-color: #8FBC8F;  color: green;  }  .c2{  background-color: #F0E68C;  color: blue;  }  /\* 3.id选择器 \*/  #p1{  font-weight: bold;  text-indent: 20px;  }  //在需要选择的标签上加上class属性或者id属性  <span class="c1 c2">  <p id="p1"> |

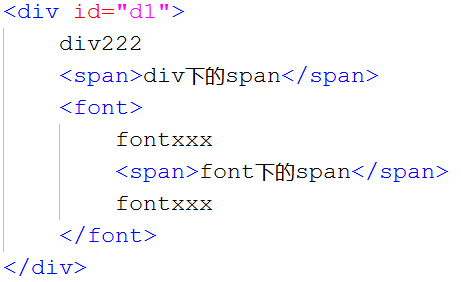
### 3.3.2.扩展选择器（案例5）

1. **后代选择器**

在父选择器选中的元素内部, 再选中指定的后代元素进行修饰

格式: 父选择器 后代选择器{ }

如: 选中div下所有的后代span 🡪 #d1 span{}

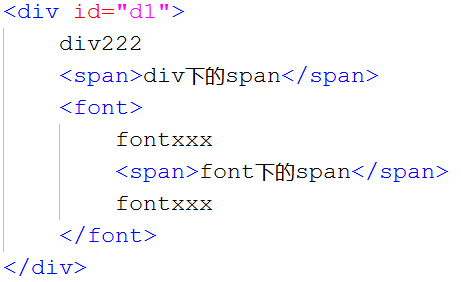


1. **子元素选择器**

在父选择器选中元素的内部, 选中指定的子元素进行样式的修饰

格式: 父选择器>子元素选择器{ }

如: 选中div下所有的子元素span 🡪 #d1>span{}



1. **分组选择器**

可以将多个选择器选中的元素统一来设置样式

格式:选择器1,选择器2,选择器3,…选择器n{}

1. **属性选择器**

可以通过属性条件选中元素进行修饰

格式: 选择器[属性条件..]{}

如: 选中所有的普通文本输入框 🡪 input[type=’text’]{}

<input type=”text” name=”xx”/>

<input type=”submit” name=”xx”/>

1. **相邻兄弟选择器**

如果两个元素具有相同的父元素, 并且是紧挨着的, 这两个元素就是相邻兄弟, 可以通过相邻兄弟选择器选中A元素后的B元素.

格式: 大哥+小弟{ }

1. **伪元素选择器**

伪元素选择器选中的不仅仅是元素本身, 还包括元素的状态

状态分为如下几种:

:link 表示元素未被点击时的状态

:hover 表示光标(鼠标)悬停时的状态(掌握!)

:active 表示元素被点击时的状态

:visited 表示元素被点击之后的状态

**案例5.** **扩展选择器**

|  |
| --- |
| /\* 1.后代选择器 \*/  #d1 span{  background-color: #DDA0DD;  font-size: 22px;  }  /\* 2.子元素选择器 \*/  #d1>span{  background-color: #DEB887;  font-size: 16px;  }  /\* 3.分组选择器 \*/  .c1,.c2,#p1 span{  background-color: #F4A460;  }  /\* 4.属性选择器 \*/  \*[class]{  border: 2px solid gray;  }  input[type="text"]{  background-color: red;  text-indent: 10px;  }  /\* 5.相邻兄弟选择器 \*/  #p1+p{  background-color: #DB7093;  }  /\* 6.伪元素选择器 \*/  a:link{  font-size:24px;  text-decoration: none;/\* 无下划线 \*/  }  a:hover{  color:green;  font-size:36px;  text-decoration: underline;/\* 有下划线 \*/  }  a:active{  color:#8FBC8F;  font-size:50px;  text-decoration: none;/\* 无下划线 \*/  }  a:visited{  color:gray;  font-size:24px;  } |

**案例6.easymall注册页面练习**

可以先将html中的内容写出来

|  |
| --- |
| <!DOCTYPE HTML>  <html>  <head>  <title>注册页面</title>  <meta charset=UTF-8 />  <link rel="stylesheet" href="css/regist.css" />  </head>  <body>  <h1>欢迎注册EasyMall</h1>  <form>  <table>  <tr>  <td class="left\_td">用户名：</td>  <td><input type="text" name="username"/></td>  </tr>  <tr>  <td class="left\_td">密码：</td>  <td><input type="password" name="password"/></td>  </tr>  <tr>  <td class="left\_td">确认密码：</td>  <td><input type="password" name="password2"/></td>  </tr>  <tr>  <td class="left\_td">昵称：</td>  <td><input type="text" name="nickname"/></td>  </tr>  <tr>  <td class="left\_td">邮箱：</td>  <td><input type="text" name="email"/></td>  </tr>  <tr>  <td class="left\_td">验证码：</td>  <td>  <input type="text" name="valistr"/>  <img src="img/regist/yzm.jpg" width="" height="" alt="" />  </td>  </tr>  <tr>  <td class="sub\_td" colspan="2"><input type="submit" value="注册用户"/></td>  </tr>  </table>  </form>  </body>  </html> |

通过html写出css内容

|  |
| --- |
| body{  font-family:"微软雅黑";  /\*加入背景图片\*/  background-image: url("../img/regist/zc.jpg");  background-repeat: no-repeat;  background-position: center 0;  }  h1{  text-align:center;  /\*外边距\*/  margin-top: 190px;  margin-bottom: 30px;  color: #990000;  }  table{  margin: 0 auto;  }  .left\_td{  font-size: 18px;  text-align:right;  }  input{  width:150px;  height:22px;  border: 1px solid #ccc;  }  td{  /\*td属性不能设置margin\*/  padding-top: 10px;  }  input[name='valistr']{  width:80px;  }  input[type="submit"]{  border:none;  width:127px;  height:44px;  font-size: 20px;  color: #fff;  font-weight: bold;  background-image: url("../img/regist/zc\_btn01.jpg");  }  .sub\_td{  text-align: center;  }  input[type="submit"]:hover{  background-image: url("../img/regist/zc\_btn02.jpg");  }  /\*resource中有vertical-align详解\*/  input[name="valistr"],input[name="valistr"]+img{  vertical-align: middle;  } |

## 3.4.外边距/边框/内边距

（案例7）

margin – 外边距

margin-top: 20px;

margin-bottom: 20px;

margin-left: 20px;

margin-right: 20px;

margin: 10px 20px 40px 60px;/\*上、右、下、左\*/

margin: 20px 10px;/\*上下、左右\*/

margin: 20px;/\*上下左右\*/

注意: 垂直外边距合并的现象:

外边距合并指的是，当两个垂直外边距相遇时，它们将形成一个外边距。合并后的外边距的高度等于两个发生合并的外边距的高度中的较大者。

border – 边框

border: 5px solid green;

border-top:5px solid green;

border-bottom: 5px solid black;

border-left: 5px solid gray;

border-right: 5px solid yellow;

padding – 内边距

padding-top: 20px;

padding-bottom: 20px;

padding-left: 20px;

padding-right: 20px;

padding: 10px 20px 40px 60px;/\*上、右、下、左\*/

padding: 20px 10px;/\*上下、左右\*/

padding: 20px;/\*四个边距\*/

**案例7. css盒子模型**

|  |
| --- |
| #div1{  width: 200px;  height: 200px;  background-color: red;  /\*外边距\*/  margin-top: 20px;  margin-bottom: 20px;  margin-left: 20px;  margin-right: 20px;  margin: 10px 20px 40px 60px;/\*上、右、下、左\*/  margin: 20px 10px;/\*上下、左右\*/  margin: 20px;/\*四个边距\*/  /\*边框\*/  border: 5px solid green;  border-top:5px solid green;  border-bottom: 5px solid black;  border-left: 5px solid gray;  border-right: 5px solid yellow;    }  #div2{  width: 200px;  height: 200px;  background-color: blue;    margin: 20px;  /\*内边距\*/  padding-top: 20px;  padding-bottom: 20px;  padding-left: 20px;  padding-right: 20px;  padding: 10px 20px 40px 60px;/\*上、右、下、左\*/  padding: 20px 10px;/\*上下、左右\*/  padding: 20px;/\* 上下左右\*/  } |

## 3.5.常用的css属性

### 3.5.1 元素类型的补充

（案例8）

块级元素:

默认情况下, 块级元素可以独占一行

块级元素可以设置宽和高

如果不设置宽度: 其中宽是默认填满父元素, 高是由内容决定(由内容撑起来)

如果设置, 就是设置的宽和高.

给块级元素设置的外边距/边框/内边距都会起作用

行内元素:

默认情况下, 多个行内元素可以处在同一行

不可以设置宽和高, 其中宽和高由内容决定

给行内元素设置的左右外边距/边框/内边距都会起作用, 但是上下外边距将不会起作用!

### 3.5.2.display属性 -- 设置元素的显示方式

block: 块级元素的默认值

inline: 行内元素的默认值

inline-block: 既具备行内元素的特性(可以同行显示), 也具备块级元素的特性(可以设置宽高)

none: 用来隐藏一个元素, 这种方式的隐藏是真的隐藏了, 元素在原位置不再占用空间

visibility: hidden 这种方式也可以隐藏一个元素, 但是这种隐藏只是元素不显示了, 在原位置仍然占用空间（很少用到）

### 3.5.3. 其他属性

**text-align 设置元素中文本水平对齐方式**

left: 默认值。左对齐

right: 右对齐

center: 居中对齐

justify: 两端对齐

**list列表属性**

list-style-type: none/disc/square/circle...

**字体属性**

font-size： 字体尺寸

font-weight： 字体加粗

font-family： 字体样式

color： 字体颜色

line-height： 行高

text-decoration： 下划线

**背景属性**

background-color： 背景颜色设置

background-image： 背景图片设置

background-repeat：设置背景图像是否及如何铺排

background-position：设置或检索对象的背景图像位置

**其他**

width 设置宽度

height 设置高度

**案例8. 元素类型补充**

|  |
| --- |
| 练习1：  **.**c1{  width:100px;  height:100px;  display: inline-block;  /\*display: none;\*/  visibility: hidden;/\*一般不用\*/  }  .c2{  width:80px;  height:80px;  display: inline-block;  }  练习2：  <ul style="list-style-type: decimal;">  练习3：  <div style="height:100px;line-height:100px;">  练习4：  <div style="height:60px;background-color: #DDA0DD; background-image: url('2.gif'); background-repeat: no-repeat; background-position:20px 10px;"> |

## 3.6.框架标签

（案例9）

frameset – 定义一个框架集

frame – 定义框架集中的一个窗口

**案例9**

**9.1.框架页面**

|  |
| --- |
| <head>  <meta http-equiv="Content-type" content="text/html; charset=UTF-8" />  <title>后台管理页面</title>  </head>  /\* noresize ：规定用户无法调整框架的大小  frameborder ：是否显示框架周围的边框，0为不显示\*/  <frameset rows="13%,87%" noresize="noresize" frameborder="0">  <frame src="\_top.html" />  <frameset cols="14%,86%">  <frame src="\_left.html" />  <frame src="\_right.html" />  </frameset>  </frameset>  <body>  </body> |

**9.2.\_top页面（通过f12选择器选择样式）**

|  |
| --- |
| <head>  <meta http-equiv="Content-type" content="text/html; charset=UTF-8" />  <title>\_top</title>  <style>  body{  background-color: #1172B5;  font-family: "微软雅黑";  font-size: 14px;  margin: 0;  }  h1{  padding-left: 30px;  color: #ffffff;  /\* letter-spacing:设置字符之间的空白\*/  letter-spacing: 5px;  /\* text-shadow:设置阴影，分别是颜色，左右移动距离，上下移动距离，阴影模糊程度\*/  text-shadow: #000 5px 5px 5px;  }  </style>  </head>  <body>  <h1>EasyMall商城管理后台</h1>  </body> |

**9.3.\_left页面（直接查看源代码即可）**

|  |
| --- |
| <head>  <meta http-equiv="Content-type" content="text/html; charset=UTF-8" />  <title>\_top</title>  <style>  body{  background-color: #32323A;  font-family: "微软雅黑";  font-size: 14px;  }  body,div{  margin: 0px;  padding:0px;  }  #menu\_bar{  font-size: 20px;  color:#FFFFFF;  }  #menu\_bar div{  border-top: 1px solid #cccccc;  padding: 5px 0;  /\* text-indent:首行缩进\*/  text-indent: 15px;  letter-spacing: 3px;  }  #menu\_bar div:last-child{  border-bottom: 1px solid #cccccc;  }  #menu\_bar div:hover{  background-color: #797981;  }  #menu\_bar div a{  font-size: 20px;  color:#FFFFFF;  /\* text-decoration:是否添加下划线\*/  text-decoration: none;  }  </style>  </head>  <body>  <div id="menu\_bar">  <div><a href="#">> 商品管理</a></div>  <div><a href="#">> 用户管理</a></div>  <div><a href="#">> 权限管理</a></div>  <div><a href="#">> 订单管理</a></div>  <div><a href="#">> 销售榜单</a></div>  </div>  </body> |

**9.4.\_right页面（直接查看源代码即可）**

|  |
| --- |
| <head>  <meta http-equiv="Content-type" content="text/html; charset=UTF-8" />  <title>\_top</title>  <style>  body{  background-color: #EDEDED;  font-family: "微软雅黑";  font-size: 14px;  margin: 0px;  }  #wel{  text-align: center;  padding: 180px 0px;  font-size: 36px;  color: #686868;  text-shadow: 5px 5px 5px #444444;  }  </style>  </head>  <body>  <div id="wel">  欢迎登陆EasyMall商城后台管理系统 ...  </div>  </body> |

## 3.7.作业练习

1. 块级元素和行内元素的区别？

答：块级元素：默认独占一行，可以包含任何元素。

行内元素：会在水平方向排列，不能包含快级元素，设置width无效，height无效，margin上下无效