# css

用途：美化html页面

学什么：

选择器

修饰的属性

## 1.css概述

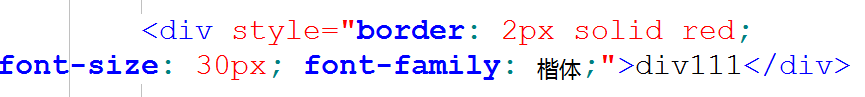
css:层叠样式表，用来美化html页面的，比如html为盖房子，css则为装修，

css增强了网页的显示能力，并且css代码可以和html代码完全分离

## 2.在html页面引入css

### 2.1.通过标签上的style属性引入

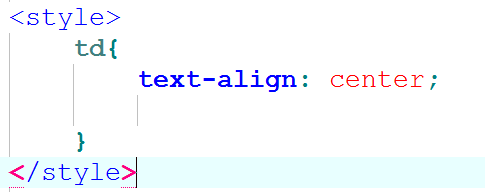
绝大部分标签都提供style属性，可以通过style属性设置当前标签的css样式



这种引入方式不推荐大量使用，会造成页面的混乱

## 2.2.通过style标签引入

在head中，可以提供一个style标签，标签内部可以写入css样式



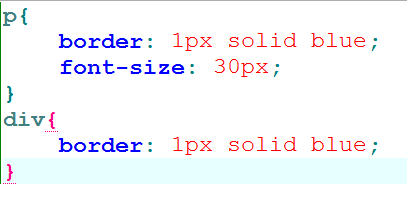
初步实现了css代码和html代码的分离

### 2.3.引入外部的css文件

通过在head标签内添加link标签引入外部的css文件



demo.css代码：



这种方式真正的实现了css代码和html代码的分离

## 3.div、span、p标签

这三个标签都可以看作是一个容器, 可以用来包裹其他的 html内容, 被包裹起来的内容形成了一个组, 可以通过css样式对这一组的内容统一设置样式.

div: 独占一行，标签可以把文档分割为独立的、不同的部分。

span：并非独占一行，多个span会并排排列，直到一行排满才会换行，标签被用来组合文档中的行内元素。

p：独占一行，但是元素会自动在其前后创建一些空白，是段落标签。

元素的类型:

块级元素: 默认情况下独占一行的元素, 如: div p h1~h6 br hr

行内元素: 默认情况下, 多个行内元素可以处在同一行。如：span input font…

## 4.选择器

选择器可以帮助我们选中想要修饰的标签进行修饰

### 4.1.基本选择器(案例4)

#### 4.1.1.标签名选择器

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>

#### 4.1.2.类选择器

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>

#### 4.1.3.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>

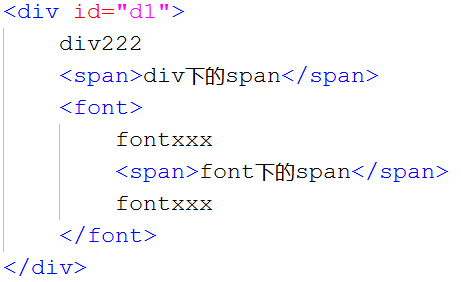
### 4.2.扩展选择器(案例5)

#### 4.2.1.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{}

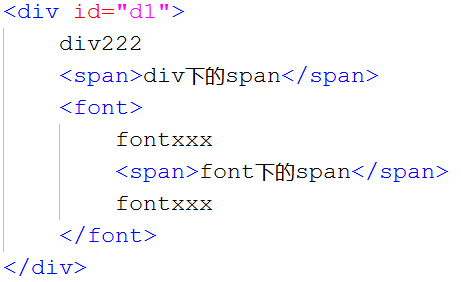


#### 4.2.2.子元素选择器

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{}



#### 4.2.3.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{}

#### 4.2.4.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”/>

#### 4.2.5.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{ }

#### 4.2.6.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

|  |
| --- |
| /\* 1.后代选择器 \*/  #d1 span{  font-size: 22px;  background-color: #DDA0DD;  }  /\* 2.子元素选择器 \*/  #d1>span{  font-size: 16px;  background-color: #DEB887;  }  /\* 3.分组选择器 \*/  .c1,.c2,#p1 span{  background-color: #F4A460;  }  /\* 4.属性选择器 \*/  \*[class]{  border: 2px solid grey;  }  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;  }  a:hover{  color:green;  font-size:36px;  text-decoration: underline;  }  a:active{  color:#8FBC8F;  font-size:50px;  text-decoration: none;  }  a:visited{  color:gray;  font-size:24px;  } |

## 案例6：easymall注册页面

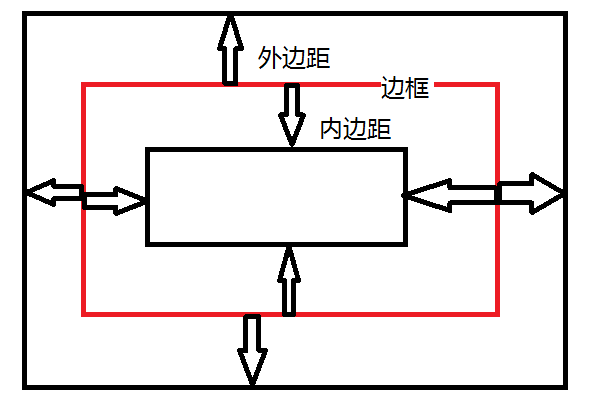
regist.html代码：

|  |
| --- |
| <!DOCTYPE HTML>  <html>  <head>  <meta http-equiv="Content-type" content="text/html; charset=UTF-8" />  <title>用户注册</title>  <link rel="stylesheet" href="css/regist.css"/>    </head>  <body>  <h1>欢迎注册EasyMall</h1>  <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" alt="验证码" />  </td>  </tr>  <tr>  <td colspan="2" id="sub\_td">  <input type="submit" value="注册用户" />  </td>  </tr>  </table>  </body>  </html> |

regist.css代码

|  |
| --- |
| body{  font-family: 微软雅黑;  /\*背景图片\*/  background-image: url("../img/regist/zc.jpg");  /\*背景铺排方式\*/  background-repeat: no-repeat;  /\*背景图片的位置，第一个参数为左右，第二个为上下\*/  background-position: center 0;  }  h1{  text-align: center;  /\*设置外边距\*/  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 #cccccc;  }  input[name="valistr"]{  width: 80px;  }  td{  padding-top: 10px;  }  input[type="submit"]{  border: none;  width: 127px;  height:44px;  font-size: 20px;  color: #ffffff;  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");  }  input[name="valistr"],input[name="valistr"]+img{  vertical-align: middle;  } |

## 5.盒子模型



/\*外边距\*/

margin: 20px;/\*四边\*/

margin: 20px 40px;/\*上下、左右\*/

margin: 20px 30px 40px 50px;/\*上右下左\*/

margin-top: 40px;

margin-bottom: 20px;

margin-left: 10px;

margin-right: 30px;

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。

/\*边框\*/

border: 10px solid pink;

border-bottom: 20px solid grey;

border-top: 20px solid green;

border-left: 20px solid blue;

border-right: 20px solid yellow;

/\*内边距\*/

padding: 20px;/\*四边\*/

padding: 30px 10px;/\*上下、左右\*/

padding: 10px 20px 30px 40px;/\*上右下左\*/

padding-bottom: 20px;

padding-left: 20px;

padding-right: 20px;

padding-top: 20px;

## 6.属性

### 6.1 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

块级元素中可以包含任何元素

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!

行内元素内不能包含块级元素

### 6.2.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）

### 6.3. 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

### 6.4.list列表属性

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

### 6.5. 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

### 6.6.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

### 6.7.其他

width 设置宽度

height 设置高度

**7.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

4.8. div、span、p标签

这三个标签都可以看作是一个容器, 可以用来包裹其他的 html内容, 被包裹起来的内容形成了一个组, 可以通过css样式对这一组的内容统一设置样式.

div: 独占一行，标签可以把文档分割为独立的、不同的部分。

span：并非独占一行，多个span会并排排列，直到一行排满才会换行，标签被用来组合文档中的行内元素。

p：独占一行，但是元素会自动在其前后创建一些空白，是段落标签。

元素的类型:

块级元素: 默认情况下独占一行的元素, 如: div p h1~h6 br hr

行内元素: 默认情况下, 多个行内元素可以处在同一行。如：span input font…