CSS培训教程

# CSS概述

## 1.1. 什么是CSS?

层叠样式表(Cascading Style Sheet ),定义了如何显示HTML元素，用来控制网页的样式和布局。

|  |
| --- |
| <html>  **未设计CSS的页面**  <head>  <title>Hello!</title>  </head>  <body>  <h2>通过样式表来改变网页外表</h2>  <p id=*"p1"*>欢迎来到CSS世界</p>  <div>css给我们带来丰富多彩的世界</div>  <br>  <div>尚硅谷欢迎你!</div>  </body>  </html> |

|  |
| --- |
| <html>  **设计了CSS的页面**  <head>  <title>测试: 内部样式表</title>  <style type=*"text/css"*>  **h2** {  color: *red*;  font-size: *46px*;  font-family: *宋体*;  }  *#p1* {  font-size: *30px*;  background: *blue*;  font-family: *黑体*;  }  *.cs* {  font-size: *20px*;  background: *green*;  font-family: *黑体*;  }  </style>  </head>  <body>  <h2>通过样式表来改变网页外表</h2>  <p id=*"p1"*>欢迎来到CSS世界</p>  <div class=*"cs"*>css给我们带来丰富多彩的世界</div>  <br>  <div class=*"cs"*>尚硅谷欢迎你!</div>  </body>  </html> |

## 1.2. 为什么要用CSS?

CSS的作用

**第一，简化网页的代码，提高访问速度**

外部的CSS文件会被浏览器保存在缓存里，加快下载显示的速度，也减少了需要上传的代码数量。

**第二，便于维护**

(1)便于修改网站样式:只要修改css文件，就可改变整个网站的风格特色。避免一个个网页的修改，大大减少重复劳动的工作量；

(2)可以重用样式；

**第三，页面内容和显示样式分离**

# 如何创建CSS?

**创建CSS有三种方式: 外部样式表, 内部样式表和内联样式**

## 外部样式表

先建立外部样式表文件(.css)，然后在网页文件的<head>内使用<link>.链接

|  |
| --- |
| **main.css h1 {font-size: 15px; font-weight: bold; color: red;}** |

|  |
| --- |
| **<head>  <title>title of article</title>  <link rel="stylesheet" href="css/main.css" type="text/css"> </head>** |

## 内部样式表

直接在<head>标签内插入<style>...</style>块

|  |
| --- |
| **<html>**  **<head> <style type="text/css">  h2 {font-size: 15px; font-weight: bold; color: blue} </style>**  **</head>**  **<body>**  **….**  **</body>** |

## 内联样式

在标签的style属性中定义样式

|  |
| --- |
| **<h3 style="font-size:14px;color:green;">**  **h3标签内的内容!**  **</h3>** |

**注意：不推荐使用该方式！将结构与表现混杂在一起，丧失样式表原有的优势！**

# CSS基础语法

## 3.1. 基本语法组成

**CSS语法由三部分组成：选择器、属性(名)和值**

**selector { property: value; }**

**selector:选择器,希望定义样式的HTML标签;**

**property: 样式名称**

**value:样式的值**

**例如：**

**p { color: #ff0000;font-size:12px }**

## 3.2. 注意事项

**1、多个属性值用分号隔开,最后一个可加也可不加;**

**2、如下写法可读性更好些；**

**p {**

**text-align: center;**

**color: black;**

**font-size: 20px;**

**}**

**3、CSS对大小写不敏感，但在选择器中对class和id名称也不敏感；**

**4、注释符号 /\* 注释内容 \*/**

# CSS选择器

CSS的选择器可以分为两大类: (标签名, 属性)

1. **基本选择器**: 通配符, 标签, ID, class和属性选择器
2. **组合选择器**: 分组选择器,后代选择器,子选择器和相邻兄弟选择器

|  |
| --- |
| <h2>通过样式表来改变网页外表</h2>  <p id=*"p1"*>欢迎来到CSS世界</p>  <div class=*"cs"*>css给我们带来丰富多彩的世界</div>  <br>  <div class=*"cs"*>尚硅谷欢迎你!</div>  <br>  <div title=*"dt"*>尚硅谷欢迎你2222!</div> |

## 基本选择器

### 标签选择器

**以标签名作为选择器**

|  |
| --- |
| **h2** {  color: *red*;  }  **div** {  color: *red*;  } |

### ID选择器

**为指定id属性值的某个标签应用样式,以#开头来定义**

|  |
| --- |
| *#p1* {  color:*#00ff00*;  font-size: *40px*;  } |

### 类(class)选择器

**为指定class属性值的某些标签应用样式,以.开头来定义**

|  |
| --- |
| *.cs* {  color: *blue*;  } |

### 属性选择器

**属性选择器可以根据元素的属性[及属性值]来选择元素。**

1. 根据属性名来匹配, 将属性名用[]包含起来

|  |
| --- |
| [title] {  color: *gray*;  font-size: *20px*;  } |

1. 根据属性名和属性值来匹配, 将属性名和值用[]包含起来

|  |
| --- |
| [title='dt'] {  color: *gray*;  font-size: *20px*;  } |

### 通配符选择器

**配置任意标签**

|  |
| --- |
| \* {  font-size: *30px*;  } |

## 组合选择器

### 分组选择器

同时为多个标签或多类标签同时应用指定样式, 中间用,隔开

|  |
| --- |
| **h2,***#p1***,***.cs* {  color:*green*;  } |

|  |
| --- |
| **<h2>通过样式表来改变网页外表</h2>**  **<p id=*"p1"*>欢迎来到CSS世界</p>**  **<div class=*"cs"*>css给我们带来丰富多彩的世界</div>**  **<br>**  **<div class=*"cs"*>尚硅谷欢迎你!</div>**  **<br>** |

### 后代选择器

|  |
| --- |
| **div** **span** {  color: *blue*;  } |

|  |
| --- |
| <div title=*"dt"*>  <h3>  **<span>尚硅谷欢迎你22222!</span>**  </h3>  **<span>尚硅谷欢迎你3333!</span>**  </div> |

### 子选择器

|  |
| --- |
| **div**>**span** {  color: *green*;  } |

|  |
| --- |
| **<div title=*"dt"*>**  **<h3>**  **<span>尚硅谷欢迎你22222!</span>**  **</h3>**  **<span>尚硅谷欢迎你3333!</span>**  **</div>** |

### 相邻兄弟选择器

|  |
| --- |
| **h3**+**span** {  color: *orange*;  } |

|  |
| --- |
| <div title=*"dt"*>  <h3>  <span>尚硅谷欢迎你22222!</span>  </h3>  **<span>尚硅谷欢迎你3333!</span>**  </div> |

# CSS主要属性

## 属性简介

学习和使用CSS属性, 可以参见CSS2.0文档



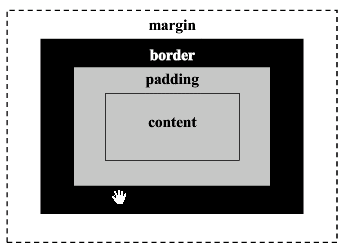
## 常用属性表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **font(字体)** | |  |  |  | | --- | --- | --- | | **color** | **:** | **指定颜色** |   **font-size: 字体尺寸**  **text-decoration: 文本的装饰 none** |
| **background(背景)** | **background-color: 背景颜色**  **background-image: 背景图片** |
| **Layout(布局)** | **visibility: 可见性**  **float: 指定是否及如何浮动** |
| **Margin(外边距)** | **margin: 指定四边的外边距 10px;**  **margin-top: 指定上边的外边距**  **margin-bottom: 指定下边的外边距**  **margin-left: 指定左边的外边距**  **margin-right: 指右四边的外边距** |
| **Border(边框)** | **border : 指定边框的样式(复合值)**  **border-color : 边框的颜色**  **border-width : 边框的宽度** |
| **Paddings(内边距)** | [**padding**](mk:@MSITStore:E:\work\work02\140215\web\html\CSS2.0.chm::/css20/c_padding.html)**: 指定四边的内边距**  [**padding-top**](mk:@MSITStore:E:\work\work02\140215\web\html\CSS2.0.chm::/css20/c_paddingtop.html)**: 指定上边的外边距**  [**padding-bottom**](mk:@MSITStore:E:\work\work02\140215\web\html\CSS2.0.chm::/css20/c_paddingbottom.html)**: 指定下边的内边距**  [**padding-left**](mk:@MSITStore:E:\work\work02\140215\web\html\CSS2.0.chm::/css20/c_paddingleft.html)**: 指定左边的内边距**  [**padding-right**](mk:@MSITStore:E:\work\work02\140215\web\html\CSS2.0.chm::/css20/c_paddingright.html)**: 指定右边的内边距** |
| **Table(表单)** | **border-collapse: 表格的行和单元格的边是合并在一起还是分开**  **border-spacing: 行和单元格的边在横向和纵向上的间距** |
| **Dimension(尺寸)** | **height: 高度**  **width: 宽度** |
| **Position(定位)** | **position: 指定定位方式** |

# CSS核心机制-盒子模型(Box Model)

## 盒子模型概述

* **盒子模型是CSS的基石,指定标签如何显示**

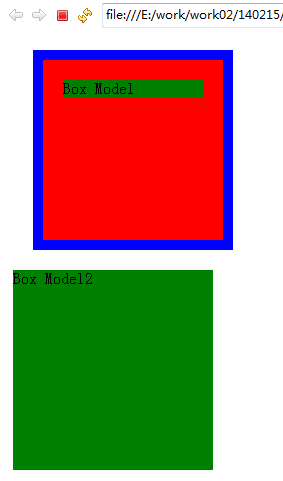


* **页面上的每个元素都被当成一个矩形盒子,占据一定的页面空间,这个盒子由内容(content)、内边距(填充,padding)、边框(border)和外边距(空白边,margin)组成；**
* **任何web页面都是由很多这样的盒子，通过不同排列组合而成，盒子之间相互影响，要掌握盒子模型需要从以下2个方面去理解：**
  + **单个盒子的内部结构；**
  + **多个盒子之间的相互关系**

## 盒子模型的四个要素

* **内容(Content)：盒子里面所装的内容**
* **内边距(填充,padding):内容到边界之间的距离**
* **边框(border)：盒子本身**
* **外边距(空白边,margin)：与其他盒子之间的距离**
  + **注意: 默认情况下,盒子边框是无,背景色是透明的,所以,看不到盒子**

|  |
| --- |
| *#div1* {  width: *200px*;  <div id=*"div1"*>  <div id=*"div2"*>  Box Model  </div>  </div>  <div id=*"div3"*>  Box Model2  </div>  height: *200px*;  background-color: *red*;  margin: *20px*;  padding: *20px*;  border-width:*10px*;  border-style:*solid*;  border-color:*blue*;  }  *#div2* {  background-color:*green*;  }  *#div3* {  width: *200px*;  height: *200px*;  background-color: *green*;  } |



# 补充:

## 块元素与行内元素

* **块元素(Block element)**
  + **默认状态下每次都占据一整行，后面内容必须再新起一行显示**
  + **<p>,<div>,<h1>..<h6>,<ul><li>,<table>…等**
* **行内元素(Inline element)** 
  + **默认状态下其内容在一行内进行显示**
  + **<a>,<img>,<input>,<span>**
* **补充说明:**
  + **每个块元素都独占一行;行元素则在一行内显示;**
  + **可以利用元素的display属性进行转换**
  + **块元素一般用作容器,可以容纳其他块元素和行元素,而行元素只能容纳文本或者其他行元素**
  + **行元素的高度宽度设置无效，其宽度就是自身文字或者图片的宽度**

## 浮动

* **浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。**

*.news* {

background-color: *gray*;

border: *solid 1px black*;

width: *400px*;

}

*.news* **img** {

float: *left*;

}

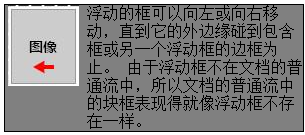
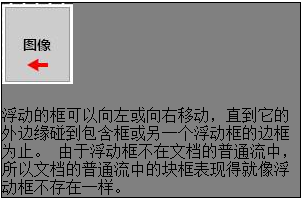
<div class=*"news"*>

<img src=*"../images/test.jpg"* />

<p>浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。由于浮动框不在文档的普通流中，所以文档的普通流中的块框表现得就像浮动框不存在一样。

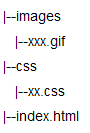
</p>

</div>



## 关于路径

* **目录结构如下：**



* **在Index.html中引用xx.css文件**

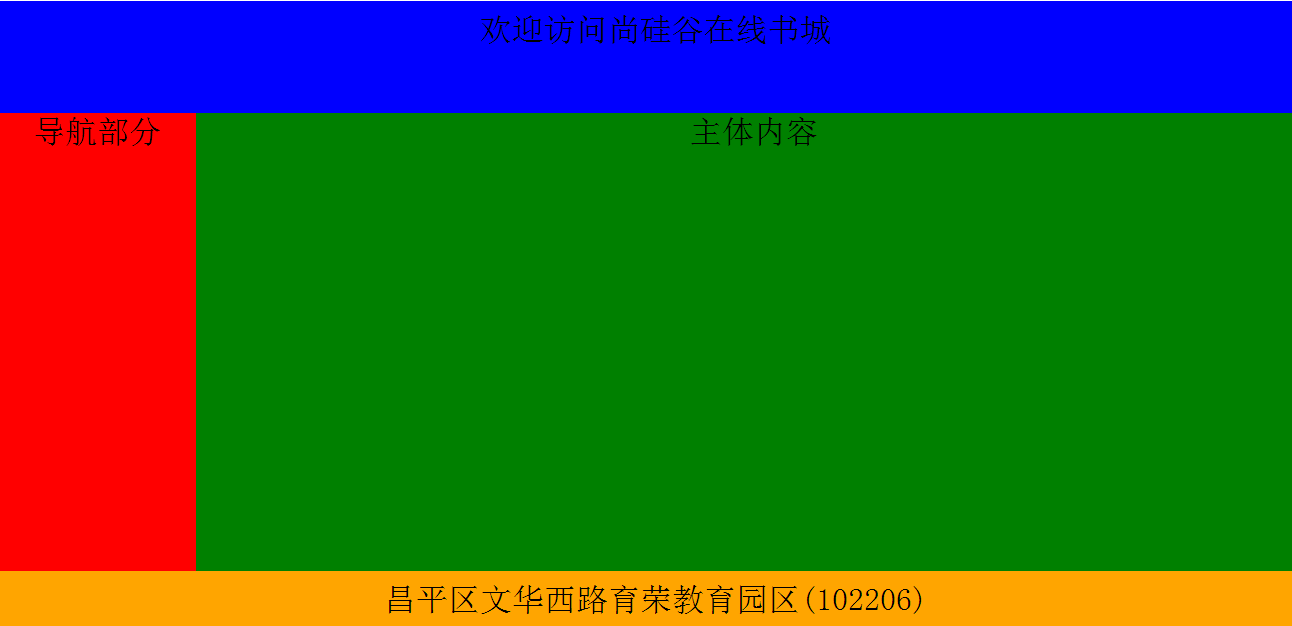
**<link rel=“”stylesheet” type=“text/css” href=“css/xx.css”>**

* **在xx.css中引用xxx.gif**

**Background:url(../images/xxx.gif)**

## DIV+CSS配合使用

* **实现下面的框架页面**



|  |
| --- |
| <html>  <head>  <title>css的综合使用: DIV+CSS</title>  <meta charset=*"UTF-8"*>  <style type=*"text/css"*>    *#head* {  height: *15%*;  background-color: *blue*;  padding: *5px*;  }    *#body* {  height: *75%*;  background-color: *gray*;  }    *#left* {  width: *15%*;  height: *100%*;  background-color: *red*;  float: *left*;  }    *#main* {  width: *85%*;  height: *100%*;  background-color: *green*;  float: *right*;  }    *#foot* {  height: *10%*;  background-color: *orange*;  padding: *5px*;  }  </style>  </head>  <body style="text-align: *center*;">  <div id=*"head"*>欢迎访问尚硅谷在线书城</div>  <div id=*"body"*>  <div id=*"left"*>导航部分</div>  <div id=*"main"*>主体内容</div>  </div>  <div id=*"foot"*>昌平区文华西路育荣教育园区(102206)</div>  </body>  </html> |