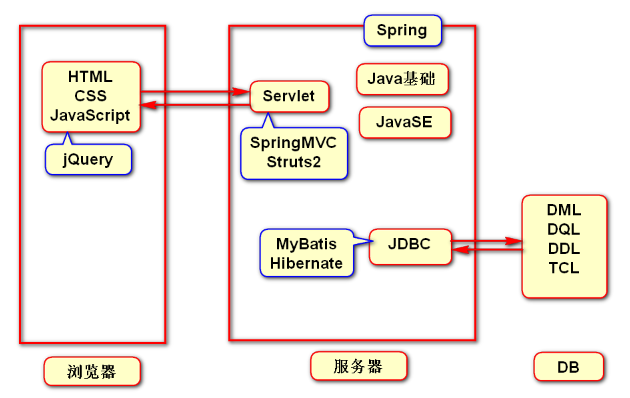
## 主题：HTML+CSS

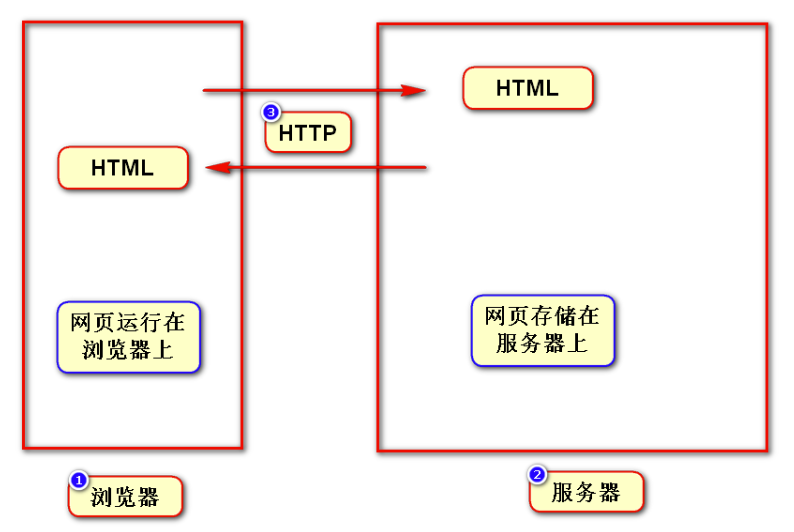
**修订履历**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **修订内容** | **修订者** | **修订时间** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 1.Java体系结构



# 2.WEB概述



# 3.HTML快速入门

## 3.1什么是HTML

Hyper Text Markup Language

超级 文本 标记 语言

该语言所组成的文件 以 .html 或 .htm 作为后缀。

## 3.2 HTML版本

1. 1999.12.24 HTML4.01
2. 2000.1.26 XHTML1.0

规范 与 html4.01 几乎相同

是更严格更纯净的 HTML 版本

1. HTML5

目标：更为简洁的 HTML 代码

## 3.3HTML基础语法

### 3.3.1标记语法 （标签）

1. 什么是标记

html 中用于描述功能的符号称之为 "标记" ，标签

<a> : 超链接 <b> : 加粗显示文本

1. 语法

使用时，必须使用 尖括号(<>)

分类：

1. 封闭类型

也叫 双标记，必须成对出现。

<标记> 内容 </标记>

<b>加粗显示</b>

注意：双标记必须成对出现

<b> fdsafdsa </b>

1. 非封闭类型

又称为 ：空标记或单标记

语法：<标记>或<标记/>

<img>或<img/>

<img></img> 错误

### 3.3.2元素

1. 元素，即标记，<a></a> ，<a>百度</a>
2. 元素的嵌套

<标记>

<标记></标记>

</标记>

注意：

1. 注意嵌套顺序

<div><a></a></div> 正确

<div><a>...</div></a>

1. 用 缩进 表示层级关系

|  |
| --- |
| <div>  <a></a>  </div> |
| <div>  <a>  <b>  <u>.....</u>  </b>  </a>  </div> |

### 3.3.3属性 和 值

1. 什么是属性

属性是用来修饰元素（标记，标签）的

1. 语法
2. 属性声明必须位于开始标记中
3. 允许出现多属性，多个属性之间用 空格 隔开，排名不分先后
4. 属性值

属性 和 属性值之间用 “=”连接。

属性 = “值”

属性 = ‘值’

<标记 属性=值 属性=值></标记>

<标记 属性=值 属性=值/>

<p align="center">...</p>

1. 标准属性
2. id

作用：定义元素在页面中的独一无二的标识

1. title

作用：鼠标移入时所提示的文字

1. class

作用：引用 类 选择器

1. style

作用：定义内联样式

### 3.3.4注释

1. 作用

被注释的东西是不会被浏览器所解释的。

<!-- -->

1. 注意
2. 注释不能被嵌套

<!--

<!--

-->

--> 错误

1. 注释不能出现在标记中

<p <!-- -->></p> 错误

## 3.4文档结构

|  |
| --- |
| <!-- 声明网页的版本 -->  <!doctype html>  <!-- 唯一的根元素 -->  <html>  <!-- 只有这2个子元素 -->  <!-- 用来对网页做出基本的声明 -->  <head>  <meta charset="utf-8"/>  <title>第一个网页</title>  </head>  <!-- 用来写出网页的结构和内容 -->  <body>  你好，HTML。  </body>  </html> |

## 练习1

1. 创建一个 html 文件
2. 步骤：
3. 创建 文本文件，将后缀名 改为 html
4. 搭建网页结构

步骤：

1. 添加文档类型声明
2. 创建 html 根标记

## 3.5常用标签

### 3.5.1 标题

1. 是通过 <h1> - <h6> 等标签进行定义的。
2. <h1> 定义最大的标题。<h6> 定义最小的标题。

|  |
| --- |
| <h1>一级标题</h1>  <h2>二级标题</h2>  <h3>三级标题</h3> |

### 3.5.2 段落

1. 段落是通过 <p> 标签定义的。
2. 浏览器会自动地在段落的前后添加空格。（<p> 是块级元素）。
3. 注意：不要忘记结束标签。

即使忘了使用结束标签，大多数浏览器也会正确地将 HTML 显示出来。

|  |
| --- |
| <p>  这是一个段落  </p> |

### 3.5.3 列表

1. 有序列表
2. 有序列表是一列项目，列表项目使用数字进行标记。
3. 有序列表始于 <ol> 标签。每个列表项始于 <li> 标签。

|  |
| --- |
| <h4>一个有序列表：</h4>  <ol>  <li>咖啡</li>  <li>牛奶</li>  <li>茶</li>  </ol> |

1. 无序列表
2. 无序列表是一个项目的列表，此列项目使用粗体圆点（典型的小黑圆圈）进行标记。
3. 无序列表始于 <ul> 标签。每个列表项始于 <li>。

|  |
| --- |
| <h4>一个无序列表：</h4>  <ul>  <li>咖啡</li>  <li>茶</li>  <li>牛奶</li>  </ul> |

1. 列表嵌套

|  |
| --- |
| <ol>  <li>  河北省  <ul>  <li>石家庄</li>  <li>保定</li>  <li>廊坊</li>  </ul>  </li>  <li>  黑龙江  <ul>  <li>哈尔滨</li>  <li>齐齐哈尔</li>  <li>牡丹江</li>  </ul>  </li>  </ol> |

### 3.5.4行内元素

1. 行内元素只占它对应标签的边框所包含的空间。
2. 格式：默认情况下，行内元素不会以新行开始，而块级元素会新起一行。
3. 内容：一般情况下，行内元素只能包含数据和其他行内元素。

|  |
| --- |
| <p>  郑州市<i>高新区</i>西<b>三环</b>翠竹街<u>6</u>号<span style="color:red;">863软件园</span>9#2楼  </p> |

### 3.5.5 空格折叠

|  |
| --- |
| <p>  窗前明&nbsp;&nbsp;&nbsp;&nbsp;月光，<br />  疑是地&nbsp;&nbsp;&nbsp;&nbsp;上霜。<br />  举头望&nbsp;&nbsp;&nbsp;&nbsp;明月，<br />  低头思&nbsp;&nbsp;&nbsp;&nbsp;故乡。<br />  </p> |

### 3.5.6 图片

1. 绝对路径：

写出图片存放的完整路径和名称，如

d:/workspace/webbasic/day01/1.jpg

/home/soft01/workspace/webbasic/day01/1.jpg

缺点：

1. 路径太长，不好写
2. 该路径是写死的，一旦软件上线时，客户要求项目存放的路径和我们开发时不一致，需要修改源码。
3. 相对路径：

只需写出图片和网页的相对关系即可，软件上线时，无论将项目存放到哪，项目内部的网页和图片的关系不变

1. 图片和网页平级：1.jpg
2. 图片在网页下级：i/1.jpg
3. 图片在网页上级：../1.jpg

注：../是返回上一级目录

|  |
| --- |
| <p>  <img src="01.jpg"/>  <img src="i/02.jpg"/>  <img src="../03.jpg"/>  <!-- 通常这样做 -->  <img src="../images/04.jpg"/>  </p> |

### 3.5.7 超链接

<a> 标签定义超链接，用于从一张页面链接到另一张页面（跳转页面）。

<a> 元素最重要的属性是 href 属性，它指示链接的目标。

1. 跳转到指定网址

|  |
| --- |
| <p>  <a href="http://www.baidu.com">百度</a>  <a href="http://www.baidu.com" target="\_blank">百度</a>  </p> |

1. 跳转到指定页面

|  |
| --- |
| <p>  <a href="Demo01.html">Demo01</a>  </p> |

1. 链接到锚点

|  |
| --- |
| <!--7.1将此处设置为锚点-->  <p>这是<a name="bottom">底部</a></p>  <!--  7.2将超链接链接到锚点上    注意:href里面记得在锚点饿name属性值前加上"#"    如果不加,则点击后会跳转到该name的文件上  (该文件不存在,会报错"找不到该文件")  -->  <p>回<a href="#bottom">底部</a></p>  <!--  从任意位置回顶部,无需设置锚点,  因为顶部有默认锚点,锚点为空,  故:在链接顶部锚点时,href的值直接为"#"即可.  -->  <p>回<a href="#">顶部</a></p> |

## 3.8 多媒体元素

### 3.8.1 图片

img标记：

1. src：图片的地址
2. alt：图片说明，在图片不能正常显示的情况下显示的文字
3. width：图片展示宽度; 可以使用像素和百分比
4. height：图片展示高度; 可以使用像素和百分比

### 3.8.2 音频

audio标记：在HTML中定义音频

1. controls：显示播放控制
2. loop：循环播放
3. autoplay：网页加载后音频立即播放
4. muted：静音

|  |
| --- |
| <audio controls="controls"  autoplay="autoplay" loop="loop" muted="muted">  <source src="Love Story-Edward Maya.mp3"/>  </audio>  注意：不同浏览器对于音频格式的支持情况不同，一般需要至少ogg与 mp3两种格式，可以支持所有主流浏览器。 |

### 3.8.3 视频

video 标记：在HTML中定义视频

1. controls：显示播放控制
2. loop：循环播放
3. autoplay：网页加载后音频立即播放
4. muted：静音
5. poster：视频没有播出时显示的图片

|  |
| --- |
| <video controls="controls" poster="/i/w3school\_logo\_black.gif" >  <source src="movie.mp4"/>  </video> |

### 3.5.8 表格

1. 表格的基本用法

table：表格

tr：行

td：单元格

border属性：设置边框线的像素，单位为像素（px，可省略不写）

cellspacing：单元格间距

caption：表格标题

|  |
| --- |
| <table border="1" cellspacing="0">  <caption>这是表的标题</caption>  <tr>  <td>编号</td>  <td>姓名</td>  <td>电话</td>  <td>住址</td>  </tr>  <tr>  <td>001</td>  <td>张三</td>  <td>110</td>  <td>高新区</td>  </tr>  <tr>  <td>002</td>  <td>李四</td>  <td>120</td>  <td>中原区</td>  </tr>  </table> |

1. 跨行和跨列

rowspan：跨行

colspan：跨列

|  |
| --- |
| <table border="1" cellspacing="0">  <tr>  <td>编号</td>  <td>姓名</td>  <td>电话</td>  <td>住址</td>  </tr>  <tr>  <td rowspan="2">001</td>  <td colspan="3">张三</td>  <!--<td>110</td>  <td>高新区</td>-->  </tr>  <tr>  <!--<td>002</td>-->  <td>李四</td>  <td>120</td>  <td>中原区</td>  </tr>  </table> |

1. 行分组

thead:表的头部

tbody：表的身体

tfoot：表的脚

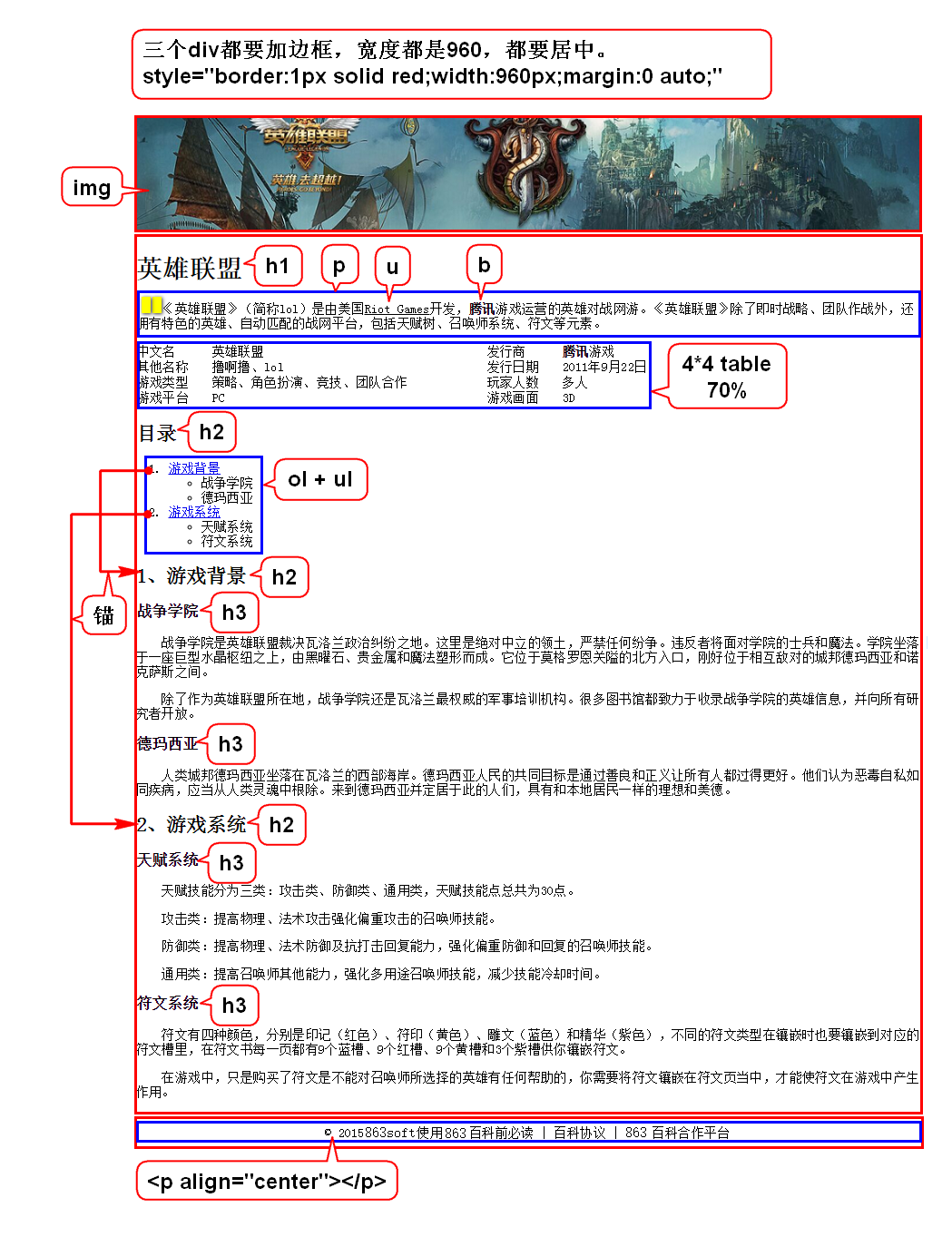
th：特殊的单元格（相比较td，自带加粗和居中效果）

|  |
| --- |
| <table border="1" cellspacing="0">  <thead>  <tr>  <th>编号</th>  <th>商品</th>  <th>金额</th>  </tr>  </thead>  <tbody style="color: blue;">  <tr>  <td>001</td>  <td>键盘</td>  <td>399</td>  </tr>  <tr>  <td>001</td>  <td>鼠标</td>  <td>299</td>  </tr>  </tbody>  <tfoot>  <tr>  <th colspan="2">合计</th>  <th>698</th>  </tr>  </tfoot>  </table> |

### 3.5.9 div版块分区

|  |
| --- |
| <!-- logo区 -->  <div>  <p>这里要放一张图片</p>  </div>  <!—导航栏 -->  <div>  <p>这里是导航栏</p>  </div>  <!-- 内容区 -->  <div style="color:red;">  <h1>宫保鸡丁</h1>  <p>先放油加热至80度</p>  <p>再放鸡丁爆炒</p>  <p>再放葱花接着炒</p>  </div>  <!-- 版权区 -->  <div>  <p>版权所有、盗版必究！</p>  </div> |

## 3.6 LOL百度百科案例



|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>英雄联盟\_863百科</title>  <meta charset="utf-8" />  </head>  <body>  <!-- header -->  <div style="width:960px;margin:0 auto;">  <img src="../images/lol.png" />  </div>    <!-- body -->  <div style="border:1px solid gray;width:960px;margin:0 auto;">  <h1>英雄联盟</h1>    <p>  &nbsp;&nbsp;《英雄联盟》（简称lol）是由美国<u>Riot Games</u>开发，<b>腾讯</b>游戏运营的英雄对战网游。《英雄联盟》除了即时战略、团队作战外，还拥有特色的英雄、自动匹配的战网平台，包括天赋树、召唤师系统、符文等元素。  </p>    <table width="70%">  <tr>  <td>中文名</td>  <td>英雄联盟</td>  <td>发行商</td>  <td><b>腾讯</b>游戏</td>  </tr>  <tr>  <td>其他名称</td>  <td>撸啊撸、lol</td>  <td>发行日期</td>  <td>2011年9月22日</td>  </tr>  <tr>  <td>游戏类型</td>  <td>策略、角色扮演、竞技、团队合作</td>  <td>玩家人数</td>  <td>多人</td>  </tr>  <tr>  <td>游戏平台</td>  <td>PC</td>  <td>游戏画面</td>  <td>3D</td>  </tr>  </table>    <h2>目录</h2>  <ol>  <li>  <a href="#g1">游戏背景</a>  <ul>  <li>战争学院</li>  <li>德玛西亚</li>  </ul>  </li>  <li>  <a href="#g2">游戏系统</a>  <ul>  <li>天赋系统</li>  <li>符文系统</li>  </ul>  </li>  </ol>    <h2><a name="g1">1、游戏背景</a></h2>  <h3>战争学院</h3>  <p>  &nbsp;&nbsp;战争学院是英雄联盟裁决瓦洛兰政治纠纷之地。这里是绝对中立的领土，严禁任何纷争。违反者将面对学院的士兵和魔法。学院坐落于一座巨型水晶枢纽之上，由黑曜石、贵金属和魔法塑形而成。它位于莫格罗恩关隘的北方入口，刚好位于相互敌对的城邦德玛西亚和诺克萨斯之间。  </p>  <p>  &nbsp;&nbsp;除了作为英雄联盟所在地，战争学院还是瓦洛兰最权威的军事培训机构。很多图书馆都致力于收录战争学院的英雄信息，并向所有研究者开放。  </p>  <h3>德玛西亚</h3>  <p>  &nbsp;&nbsp;人类城邦德玛西亚坐落在瓦洛兰的西部海岸。德玛西亚人民的共同目标是通过善良和正义让所有人都过得更好。他们认为恶毒自私如同疾病，应当从人类灵魂中根除。来到德玛西亚并定居于此的人们，具有和本地居民一样的理想和美德。  </p>    <h2><a name="g2">2、游戏系统</a></h2>  <h3>天赋系统</h3>  <p>&nbsp;&nbsp;天赋技能分为三类：攻击类、防御类、通用类，天赋技能点总共为30点。</p>  <p>&nbsp;&nbsp;攻击类：提高物理、法术攻击强化偏重攻击的召唤师技能。</p>  <p>&nbsp;&nbsp;防御类：提高物理、法术防御及抗打击回复能力，强化偏重防御和回复的召唤师技能。</p>  <p>&nbsp;&nbsp;通用类：提高召唤师其他能力，强化多用途召唤师技能，减少技能冷却时间。</p>  <h3>符文系统</h3>  <p>  &nbsp;&nbsp;符文有四种颜色，分别是印记（红色）、符印（黄色）、雕文（蓝色）和精华（紫色），不同的符文类型在镶嵌时也要镶嵌到对应的符文槽里，在符文书每一页都有9个蓝槽、9个红槽、9个黄槽和3个紫槽供你镶嵌符文。  </p>  <p>  &nbsp;&nbsp;在游戏中，只是购买了符文是不能对召唤师所选择的英雄有任何帮助的，你需要将符文镶嵌在符文页当中，才能使符文在游戏中产生作用。  </p>  </div>  <!-- footer -->  <div style="width:960px;margin:0 auto;">  <p align="center">&copy; 2015 863soft 使用863百科前必读 | 百科协议 | 863百科合作平台</p>  </div>  </body>  </html> |

## 3.7 form表单

### 3.7.1 表单元素

表单元素就是表单的边界，用来圈定一个范围，此范围内的数据才能提交给服务器。

action：用来声明表单数据提交的目标

method：后面JavaEE部分会讲

|  |
| --- |
| <form action="" method="">  </form> |

### 3.7.2 表单控件

表单控件：用来输入数据。

input(单标记,9种)，这9个元素的标签名是一样的,但是他们的type属性值不同。

文本框、密码框 通用属性：

1. value:设置默认值
2. maxlength:设置最大长度
3. readonly:设置只读
4. 文本框

type属性值为text

|  |
| --- |
| <p>  账号:<input type="text" value="123"  maxlength="6" readonly/>  </p> |

1. 密码框

type属性值为password

|  |
| --- |
| <p>  密码:<input type="password"/>  </p> |

1. 单选框
2. type属性为radio
3. name:组名,同一组radio互斥(同名互斥)
4. checked:设置默认选中
5. <label for="需要关联的标记的id值"></label>:可是实现点击label标记内的文字,等同于点击单选框/复选框的效果

|  |
| --- |
| <p>  性别:  <input id="man" type="radio" name="sex" checked/>  <label for="man">男</label>  <input type="radio" name="sex"/>女  </p> |

1. 多选框
2. type属性为checkbox
3. name:组名
4. value:复选框对应的值(JavaEE部分会详细讲)

|  |
| --- |
| <p>  兴趣:  <input type="checkbox" name="interest"/>旅游  <input type="checkbox" name="interest" checked/>电子竞技  <input type="checkbox" name="interest"/>不学习  </p> |

1. 文件框

|  |
| --- |
| <p>  头像:<input type="file"/>  </p> |

1. 隐藏框

|  |
| --- |
| <p>  <input type="hidden" value="love"/>  </p> |

1. 按钮

submit按钮、reset按钮、button按钮，他们的名字都是通过value进行定义的。

button按钮默认无功能，需通过js进行定义。

|  |
| --- |
| <p>  <input type="submit" value="注册"/>  <input type="reset" value="重填"/>  <input type="button" value="测试"/>  </p> |

### 3.7.3其他表单元素

1. label(了解)

该标签不是用来输出数据的，仅仅是用来管理表单中的文本的，它可以将文本与框绑定到一起，从而增加框的点击范围，提高易用性。

|  |
| --- |
| <p>  <!-- id是元素的唯一标识，任何元素都可以有id。 -->  <input type="checkbox" id="c1"/>  <label for="c1">我已阅读并自愿遵守此协议！</label>  </p> |

1. 下拉选

selected：设置默认选中

value：将来再讲

|  |
| --- |
| <p>  城市:  <select>  <option>请选择</option>  <option>佳木斯</option>  <option>廊坊</option>  <option>北京</option>  <option selected>沈阳</option>  </select>  </p> |

1. 文本域

cols：设置宽度

rows：设置高度

|  |
| --- |
| <p>  描述:  <textarea cols="30" rows="8"></textarea>  </p> |

## 3.8 多媒体元素

### 3.8.1 图片

img标记：

1. src：图片的地址
2. alt：图片说明，在图片不能正常显示的情况下显示的文字
3. width：图片展示宽度; 可以使用像素和百分比
4. height：图片展示高度; 可以使用像素和百分比

### 3.8.2 音频

audio标记：在HTML中定义音频

1. controls：显示播放控制
2. loop：循环播放
3. autoplay：网页加载后音频立即播放
4. muted：静音

|  |
| --- |
| <audio controls="controls"  autoplay="autoplay" loop="loop" muted="muted">  <source src="Love Story-Edward Maya.mp3"/>  </audio>  注意：不同浏览器对于音频格式的支持情况不同，一般需要至少ogg与 mp3两种格式，可以支持所有主流浏览器。 |

### 3.8.3 视频

video 标记：在HTML中定义视频

1. controls：显示播放控制
2. loop：循环播放
3. autoplay：网页加载后音频立即播放
4. muted：静音
5. poster：视频没有播出时显示的图片

|  |
| --- |
| <video controls="controls" poster="/i/w3school\_logo\_black.gif" >  <source src="movie.mp4"/>  </video> |

# 4.CSS基础

## 4.1 CSS的三种引入方式

### 4.1.1 内联样式

行内样式，直接在style属性里写的样式。

|  |
| --- |
| <!-- 1.内联样式：直接在style属性里写的样式。 -->  <h1 style="color:red;">CSS</h1>  <h2>CSS可以美化网页</h2>  <h2>CSS有3种用法</h2>  <p>1.内联样式</p>  <p>2.内部样式</p>  <p>3.外部样式</p> |

### 4.1.2 内部样式

在head内部的style标签里写的样式。

|  |
| --- |
| <!-- 2.内部样式：在head内部的style标签里写的样式。 -->  <style>  /\*css注释是这样的\*/  h2 {  color: blue;  }  </style> |

### 4.1.3 外部样式

在单独的css文件内写样式，需要将此文件引入到网页上。

|  |
| --- |
| <link rel="stylesheet" href="my.css"/> |

## 4.2 CSS特性

### 4.2.1 继承性

父元素的某些样式，可以被子元素继承(字体、颜色)。

|  |
| --- |
| body {  font-family: "微软雅黑","文泉驿正黑","黑体";  }  <body>  <h1>河南省</h1>  <h2>郑州市</h2>  </body> |

### 4.2.2 层叠性

给一个元素设置不同的样式，其效果会叠加在一起。

|  |
| --- |
| body {  font-family: "微软雅黑","文泉驿正黑","黑体";  }  h1 {  font-size: 50px;  color: red;  }  <body>  <h1>河南省</h1>  <h2>郑州市</h2>  </body> |

### 4.2.3 优先级

给一个元素设置相同的样式，其效果以后者为准，也叫就近原则。

|  |
| --- |
| h2 {  color: yellow;  }  /\*......\*/  h2 {  color: green;  }  <body>  <h1>河南省</h1>  <h2>郑州市</h2>  </body> |

## 4.3 CSS常见选择器

### 4.3.1 元素选择器

|  |
| --- |
| h2 {  color: yellow;  } |

### 4.3.2 类选择器

根据class属性选择一组元素。

|  |
| --- |
| .female {  color: pink;  }  <p class="female">床前明月光</p>  <p>疑是地上霜</p>  <p class="female">举头望明月</p>  <p id="p4">低头思故乡</p> |

### 4.3.3 ID选择器

根据id属性选择唯一的元素。

|  |
| --- |
| #p4 {  color: red;  }  <p class="female">床前明月光</p>  <p>疑是地上霜</p>  <p class="female">举头望明月</p>  <p id="p4">低头思故乡</p> |

### 4.3.4 选择器组

写出一组选择器，选中每个 选择器所对应的目标的并集(和)。

|  |
| --- |
| .female,#p4 {  font-weight: bold;  }  <p class="female">床前明月光</p>  <p>疑是地上霜</p>  <p class="female">举头望明月</p>  <p id="p4">低头思故乡</p> |

### 4.3.5 派生选择器

选择某元素的后代：

1. 选择某元素的子孙

|  |
| --- |
| #p5 b {  color: red;  }  <p id="p5">  郑州市<i><b>高新区</b></i>西<b>三环</b>翠竹街  <u>6</u>号<span style="color:red;">  863软件园</span>9#2楼  </p> |

1. 选择某元素的儿子

|  |
| --- |
| #p5>b {  font-size: 30px;  }  <p id="p5">  郑州市<i><b>高新区</b></i>西<b>三环</b>翠竹街  <u>6</u>号<span style="color:red;">  863软件园</span>9#2楼  </p> |

### 4.3.6 伪类选择器

根据状态选择元素：

1. 未访问过的超链接

|  |
| --- |
| a:link {  color: green;  }  <p>  <a href="http://www.baidu.com">百度</a>  <a href="http://www.xb.com">瞎编</a>  </p> |

1. 访问过的超链接

|  |
| --- |
| a:visited {  color: red;  }  <p>  <a href="http://www.baidu.com">百度</a>  <a href="http://www.xb.com">瞎编</a>  </p> |

1. 选择激活态(正在点击)的按钮

|  |
| --- |
| #b1:active {  background-color: yellow;  }  <p>  <input type="button" value="试试" id="b1"/>  </p> |

1. 选择有焦点(光标)的文本框

|  |
| --- |
| #t1:focus {  background-color: green;  }  <p>  <input type="text" id="t1"/>  </p> |

1. 选择悬停态(鼠标正在上方)的图片

|  |
| --- |
| img:hover {  width: 250px;  height: 250px;  }  <p>  <img src="../images/01.jpg"/>  <img src="../images/02.jpg"/>  <img src="../images/03.jpg"/>  </p> |

## 4.4 CSS声明

### 4.4.1 border与box

1. border属性：用来设置元素的边框。
2. 四边设置：

border：width值 style值 color值;

style值：

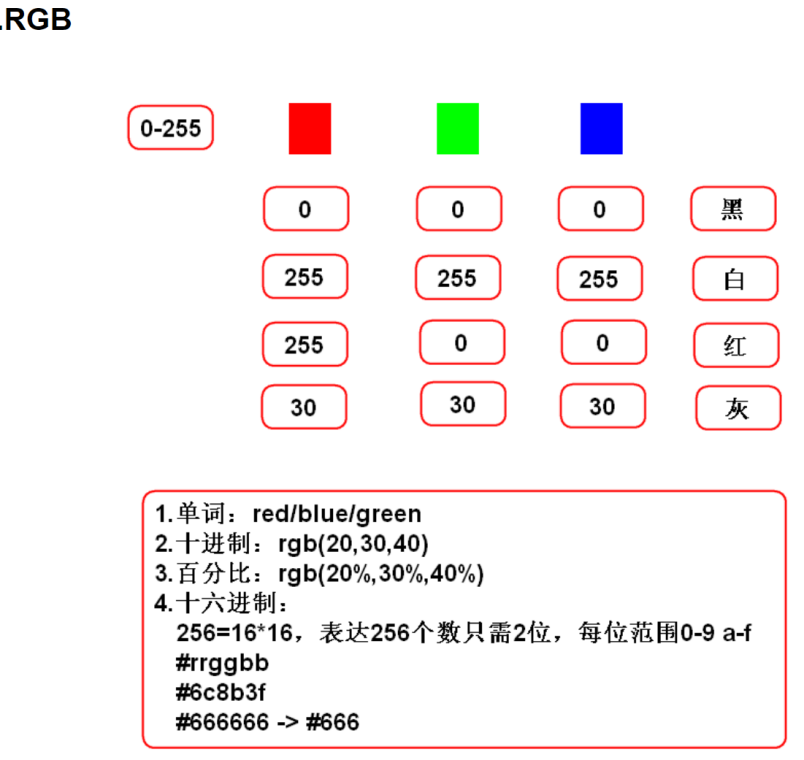
1. dotted：点状线
2. solid：实线
3. double：双线
4. dashed：虚线

width值的单位：

1. px：像素，计算机屏幕上的一个点
2. em：1em等于当前的字体尺寸，2em等于当前字体尺寸的2倍
3. %：百分比
4. in：英寸
5. cm：厘米
6. mm：毫米
7. pt：磅（1pt等于1/72英寸）

color的值：

1. #rrggbb：16进制数，如#ff0000
2. #rgb：简写16进制的数，如：#f00
3. rgb(x,x,x)：RGB值，如rgb( 255,0,0)
4. rgb(x%,x%,x%)：RGB百分比值，如：rgb(100%，0%，0%)
5. 表示颜色的英文单词，如red，blue



1. 单边设置：

border-left：width值 style值 color值;

border-right：width值 style值 color值;

border-top：width值 style值 color值;

border-bottom：width值 style值 color值;

|  |
| --- |
| <h1>关雎</h1>  <p>  关关雎鸠，在河之洲。窈窕淑女，君子好逑。<br />  参差荇菜，左右流之。窈窕淑女，寤寐求之。<br />  求之不得，寤寐思服。悠哉悠哉，辗转反侧。<br />  参差荇菜，左右采之。窈窕淑女，琴瑟友之。<br />  参差荇菜，左右芼之。窈窕淑女，钟鼓乐之。<br />  </p>  /\*1.单个边设置边框\*/  h1 {  border-left: 10px solid blue;  }  /\*2.四个边设置相同的边框\*/  p {  border: 1px dashed red;  } |

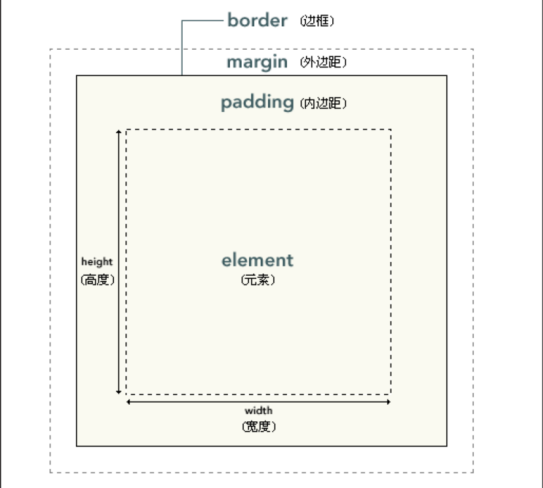
1. overflow属性：规定当内容溢出元素框时发生的事情。

|  |  |
| --- | --- |
| **值** | **描述** |
| visible | 默认值。内容不会被修剪，会呈现在元素框之外。 |
| hidden | 内容会被修剪，并且其余内容是不可见的。 |
| scroll | 内容会被修剪，但是浏览器会显示滚动条以便查看其余的内容。 |
| auto | 如果内容被修剪，则浏览器会显示滚动条以便查看其余的内容。 |
| inherit | 规定应该从父元素继承 overflow 属性的值。 |

|  |
| --- |
| <p>  关关雎鸠，在河之洲。窈窕淑女，君子好逑。<br />  差荇菜，左右流之。窈窕淑女，寤寐求之。<br />  求之不得，寤寐思服。悠哉悠哉，辗转反侧。<br />  参差荇菜，左右采之。窈窕淑女，琴瑟友之。<br />  参差荇菜，左右芼之。窈窕淑女，钟鼓乐之。<br />  </p>  /\*3.内容溢出时的处理：  默认情况下，元素的高度会自适应，不会溢出。  当设置了固定的高度后可能会导致溢出。\*/  p {  width: 200px;  height: 60px;  overflow: auto;  } |

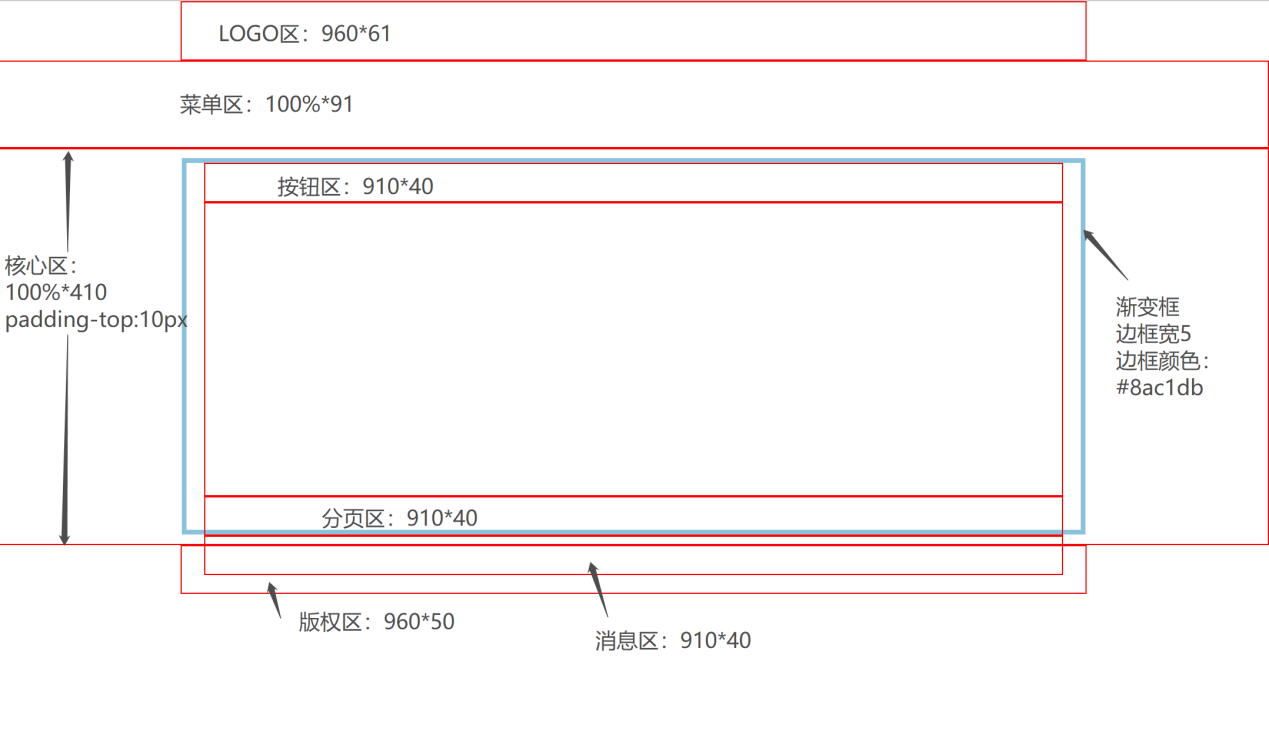
1. box（Box Model）

定义了元素框处理元素内容、内边距、边框和外边距的方式。



|  |
| --- |
| <div id="d0">aa</div>  <div id="d1">bb</div>  <div id="d2">cc</div>  <div id="d3">dd</div>  <div id="d4">ee</div>  <div id="d5">ff</div>  div {  border: 1px solid red;  width: 100px;  height: 100px;  }  /\*1.四个边设置相同的边距\*/  #d1 {  padding: 20px;  margin: 30px;  }  /\*2.四个边设置不同的边距(上右下左)\*/  #d2 {  padding: 10px 20px 30px 40px;  margin: 40px 30px 20px 10px;  }  /\*3.单个边设置边距\*/  #d3 {  padding-left: 20px;  margin-bottom: 30px;  }  /\*4.对边设置边距(上下 左右)\*/  #d4 {  padding: 20px 40px;  margin: 30px 50px;  }  /\*5.外边距的特殊用法：  在使用对边设置外边距时，若左右值为auto，  这元素将水平居中。 \*/  #d5 {  margin: 20px auto;  } |

### 4.4.2 box练习：定义管理员页面的整体布局



HTML代码：

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title>管理员列表</title>  <link rel="stylesheet" href="../css/my\_list.css"/>  </head>  <body>  <!-- logo区 -->  <div class="logo">    </div>  <!-- 菜单区 -->  <div class="menu"></div>  <!-- 内容区 -->  <div class="core">  <div class="content">  <!-- 按钮区 -->  <div class="btn"></div>  <!-- 数据区 -->  <div class="data"></div>  <!-- 分页区 -->  <div class="page"></div>  <!-- 消息区 -->  <div class="message"></div>  </div>  </div>  <!-- 版权区 -->  <div class="cr"></div>  </body>  </html> |

CSS代码：

|  |
| --- |
| /\*0.对网页样式做一些统一的设置\*/  \* {  padding: 0;  margin: 0;  }  /\*1.对网页进行分区布局\*/  div {  border: 1px solid red;  margin: 0 auto;  }  .logo {  width: 960px;  height: 61px;  }  .logo\_img{  height: 61px;  }  .menu {  height: 91px;  }  .core {  height: 410px;  /\*顶部预留一个放渐变条的空间\*/  padding-top: 10px;  }  .cr {  width: 960px;  height: 50px;  }  .content {  width: 950px;  height: 390px;  border: 5px;  }  .content div {  width: 910px;  height: 40px;  }  .content .data {  height: 310px;  } |

### 4.4.3 背景

1. 背景色

background-color属性用于为元素设置背景色，该属性接受任何合法的颜色值。

|  |
| --- |
| body{  background-color:#ccc;  } |

1. 背景图片

background-image属性用于设置背景图片

1. 默认值是none，表示背景上没有放置任何图像
2. 如果需要设置一个背景图像，需要用url附带一个图像的URL值
3. URL值可以是绝对路径，也可以是相对路径

|  |
| --- |
| body {  background-image: url(../images/background.png);  } |

默认情况下，背景图片在水平和垂直方向上重复出现，创建一种称为“墙纸”的效果。

background-repeat可取值为：

|  |  |
| --- | --- |
| **值** | **描述** |
| repeat | 默认。背景图像将在垂直方向和水平方向重复。 |
| repeat-x | 背景图像将在水平方向重复。 |
| repeat-y | 背景图像将在垂直方向重复。 |
| no-repeat | 背景图像将仅显示一次。 |
| inherit | 规定应该从父元素继承 background-repeat 属性的设置。 |

|  |
| --- |
| body {  background-image: url(../images/background.png);  background-repeat: repeat-y;  background-position: center;  } |

background-position属性用于改变背景图片在元素中的位置，该属性的取值为：

|  |  |
| --- | --- |
| 值 | 说明 |
| x% y% | 第一个值是水平位置，第二个值是垂直位置。  左上角是 0% 0%。右下角是 100% 100%。  如果您仅规定了一个值，另一个值将是 50%。 |
| x y | 第一个值是水平位置，第二个值是垂直位置。  左上角是 0 0。单位是像素 (0px 0px) 或任何其他的 CSS 单位。  如果您仅规定了一个值，另一个值将是50%。  您可以混合使用 % 和 position 值。 |
| left | 在页面或者包含元素的左边显示 |
| center | 在页面或者包含元素的中间显示 |
| right | 在页面或者包含元素的右边显示 |
| top | 在页面或者包含元素的顶部显示 |
| bottom | 在页面或者包含元素的底部显示 |

|  |
| --- |
| body {  background-image:url(../img/01.jpg);  background-repeat: repeat-y;  background-position: center;  } |

背景图像可以简写：

|  |
| --- |
| /\*2.采用简化的方式设置背景:  background: 颜色 图片 平铺 位置;  上述值可以酌情省略，但至少要保留颜色或图片之一。 \*/  .enemy {  background:  url(../images/airplane.png)  no-repeat center;  }  .hero {  background:  url(../images/hero0.png)  no-repeat center;  } |

默认情况下，背景图像会随着页面的滚动而移动，我们可以通过background-attachment属性来改变此特征。

-默认值是scroll：默认情况下，背景会随文档而滚动

-可取值fixed：背景图像固定，并不会随着页面的其余部分滚动，常用语实现称为水印的图像。

|  |
| --- |
| body {  background-attachment: fixed;  } |

### 4.4.4 文本格式化

* 1. 控制文本

1. 指定字体：font-family:value1,value2;
2. 字体大小：font-size:value;
3. 字体加粗：font-weight: bold/ normal(默认)

|  |
| --- |
| <p id="p1">少壮不努力，老大徒伤悲</p>  #p1{  font-family: 宋体;  font-size: 50px;  font-weight: bold;  } |

* 1. 控制文本格式

1. 文本颜色：color:value
2. 文本排列：text-align:left/right/center
3. 文字修饰：text-decoration：value

|  |  |
| --- | --- |
| **值** | **描述** |
| none | 默认。定义标准的文本。 |
| underline | 定义文本下的一条线。 |
| overline | 定义文本上的一条线。 |
| line-through | 定义穿过文本的一条线。 |

1. 行高：line-height : 值(1.6em)
2. 首行文本缩进：text-indent:value(2em)

|  |
| --- |
| <p id="p1">  少壮不努力，老大徒伤悲<br />  少小离家老大回，乡音无改鬓毛衰  </p>  #p1{  color: red;  text-decoration:line-through;  line-height: 1.5em;  text-indent:2em;  } |

### 4.4.5 表格样式

表格可以使用box模型（边框、内边距、宽、高）以及文本格式化属性。

表格特有的样式属性为：border-collapse:collapse; 如果可能，边框会合并为一个单一的边框。

### 4.4.6 定位

定位：定义元素框相对于其正常位置应该出现的位置，或者相对于父元素、另一个元素甚至浏览器窗口本身的位置。

1. 流定位
2. 浮动定位
3. 相对定位
4. 绝对定位
5. 固定定位

#### 4.4.6.1 流定位

流定位：

1. 页面中的块级元素框从上到下一个接一个的排列。

-每一个块级元素将在一个新行中（块级元素会自动换行，不会自动换行的叫做行级元素）

-元素框之间的垂直距离是由框的垂直外边距计算出来的

1. 行内元素将在一行中从左到右排列水平布置。

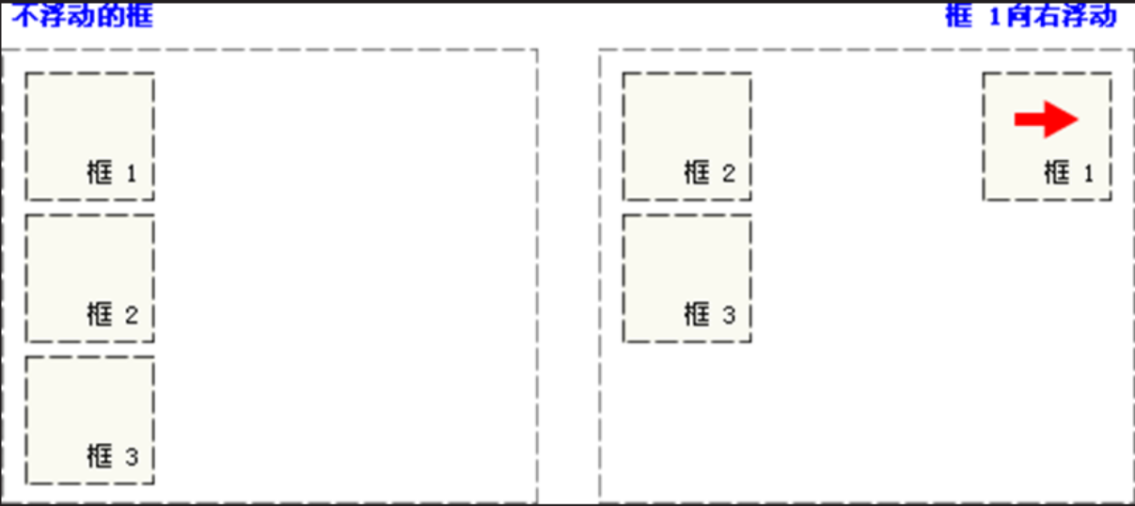
-不需要重新开始行

-可以使用水平内边距、边框和外边距调整它们之间的距离

#### 4.4.6.2 浮动定位

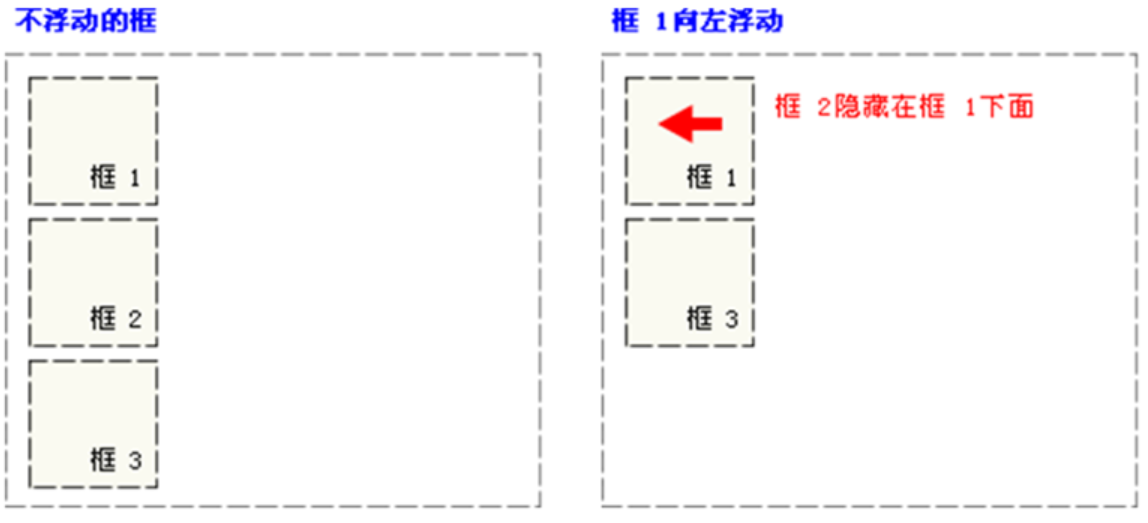
1. 浮动定位是指
2. 让元素脱离普通流定位
3. 将浮动元素放置在父元素的左边或者右边
4. 浮动元素依旧位于父元素之内
5. 浮动的框可以向左或向右移动，直到它的外边缘碰到父元素或另一个浮动框的边框为止。

案例：父元素中有三个元素框，如果把框1向右浮动，则它脱离文档流并且向右移动，直到它的右边缘碰到包含框的右边缘。



当第1个框向左浮动时，它脱离流定位，并且向左移动，直到它的边缘碰到父元素的边缘。

因为框1不再处于流定位中，所以它不占据空间，实际上覆盖住了框2，使框2从视图中消失了。



|  |
| --- |
| <div id="d0">  <div id="d1"></div>  <div id="d2"></div>  <div id="d3"></div>  <div id="d4"></div>  </div>  <p>浮动时观察我的位置</p>  #d0,p {  width: 400px;  border: 1px solid red;  }    #d1,#d2,#d3 {  width: 100px;  height: 100px;  margin: 10px;  }    #d1 {  background-color: red;  }    #d2 {  background-color: green;  }    #d3 {  background-color: blue;  }  /\*浮动\*/  #d1,#d2,#d3 {  float: left;  }  /\*消除浮动影响  \* 可以通过对p设置消除浮动来解决浮动带来的影响，  \* 也可以在下方加1个空白元素，让浮动元素来覆盖他，  \* 达到消除浮动影响的效果  \* \*/    p {  /\*clear: left;\*/  }    #d4 {  clear: left;  } |

clear属性用于清除浮动所带来的影响。

clear：none/left/right/both(左右两侧)

定义了元素的哪边上不允许出现浮动元素。

#### 4.4.6.3 相对定位

1. 元素原本所占空间不释放
2. 元素框会相对于它原来的位置偏移某个距离

设置垂直或水平位置，让元素对于它的起点进行移动

1. 设置元素为相对定位
2. 首先需要设置position属性的值为relative
3. 然后需要使用left属性或者right属性设置水平方向的偏移量
4. 也可以使用top属性或bottom属性设置垂直方向的偏移量

|  |
| --- |
| <body>  <img src="../img/0.jpg" />  </body>  //离左边偏移2像素，离上方偏移-2像素  img:hover {  position: relative;  left: 2px;  top: -2px;  } |

#### 4.4.6.4 照片墙案例

|  |
| --- |
| <ul>  <li>  <img src="../img/01.jpg"/>  <p>照片墙1</p>  </li>  <li>  <img src="../img/02.jpg"/>  <p>照片墙2</p>  </li>  <li>  <img src="../img/03.jpg"/>  <p>照片墙3</p>  </li>  <li>  <img src="../img/04.jpg"/>  <p>照片墙4</p>  </li>  <li>  <img src="../img/05.jpg"/>  <p>照片墙5</p>  </li>  <li>  <img src="../img/06.jpg"/>  <p>照片墙6</p>  </li>  </ul>  body {  background-color: #066;  }  ul {  width: 780px;  /\*border: 1px solid red;\*/  margin: 20px auto;  /\*去掉左侧的符号\*/  list-style-type: none;  /\*去掉左侧自带的内边距\*/  padding-left: 0;  }  li {  padding: 10px;  margin: 10px;  border: 1px solid #ccc;  float: left;  background-color: #FFF;  }  p {  text-align: center;  }  /\*采用相对定位实现抖动效果\*/  li:hover {  position: relative;  left: 2px;  top: -2px;  } |

#### 4.4.6.5 绝对定位

1. 将元素的内容从当前定位中移除，释放空间
2. 并使用偏移属性来固定该元素的位置
3. 相对于最近的已定位祖先元素
4. 如果元素没有已定位的祖先元素，那么它的位置相对于最初的包含块，比如body元素。
5. 设置元素为绝对定位
6. 首先需要设置position属性的值为absolute
7. 然后使用left属性或者right属性设置元素的水平位置
8. 也可以使用top属性或者bottom属性设置元素的垂直位置。

|  |
| --- |
| <div>  <img src="../img/01.jpg"/>  <p>那是一个秋天</p>  </div>  div {    border: 1px solid red;  /\*此处加position仅仅是为了将其作为  相对定位参考的目标而已，该元素不需要  释放位置及偏移。 \*/  position: relative;  }  img,div{  width: 318px;  height: 318px;  }  p {  /\*绝对定位\*/  position: absolute;  bottom: 250px;  width: 318px;  line-height: 20px;  background-color: #FFF;  text-align: center;  } |

#### 4.4.6.6 固定定位

1. 将元素的内容固定在页面的某个位置
2. 元素从普通流中完全移除，不占用页面空间
3. 当用户向下滚动页面时元素框并不随着移动
4. 设置固定定位
5. 首先需要设置position属性的值为fixed
6. 通过left、top、right以及这些偏移量属性来定义元素的位置

|  |
| --- |
| <h1>iPhone7</h1>  <p>这是一个新款手机</p>  <p>它能防火</p>  <p>它能防盗</p>  <p>它能防老王</p>  <p>......</p>  此处省略n多段落  <p>......</p>  <div>  <a href="#">顶部</a>  </div>  <style>  div {  width: 60px;  text-align: center;  background-color: #CCC;  /\*固定定位\*/  position: fixed;  bottom: 30px;  right: 10px;  }  </style> |

#### 4.4.6.7 堆叠顺序

1. 一旦修改了元素的定位方式，则元素可能会发生堆叠
2. 可以使用z-index属性来控制元素框出现的重叠顺序
3. z-index属性：
4. 值为数值：数值越大表示堆叠顺序越高，即离用户更近
5. 拥有更高堆叠顺序的元素总是会处于堆叠顺序较低的元素的前面，即元素的显示会接近页面表面
6. 可以设置为负值，表示离用户越远。

|  |
| --- |
| <div class="d0">  <img src="../img/03.jpg" class="i1"/>  <img src="../img/04.jpg" class="i2"/>  <img src="../img/05.jpg" class="i3"/>  </div>  <style>  .d0{  width: 10px;  height: 10px;  border: 1px solid red;  margin: 200px 500px;  position: relative;  }  .i1{  position: absolute;  z-index: 3;  }  .i2{  position: absolute;  left: 50px;  top: -150px;  }  .i3{  position: absolute;  left: 250px;  top: -50px;  }  </style> |

#### 4.4.6.8照片墙加强版

鼠标悬停在图片上时，将图片放大。

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <title></title>  <style type="text/css">  img {  border: 0;  width: 100px;  height: 75px;  }    .box {  width: 560px;  margin: 100px auto;  /\*margin: 0 auto;\*/  position: relative;  }    .box div {  width: 106px;  height: 81px;  border: 3px solid #efedf0;  float: left;  }    .box div:hover img {  width: 200px;  height: 150px;  position: absolute;  margin-top: -25px;  margin-left: -100px;  }  </style>  </head>  <body>  <div class="box">  <div><img src="../img/01.jpg"></div>  <div><img src="../img/02.jpg"></div>  <div><img src="../img/03.jpg"></div>  <div><img src="../img/04.jpg"></div>  <div><img src="../img/01.jpg"></div>  <div><img src="../img/02.jpg"></div>  <div><img src="../img/03.jpg"></div>  <div><img src="../img/04.jpg"></div>  <div><img src="../img/01.jpg"></div>  <div><img src="../img/02.jpg"></div>  <div><img src="../img/03.jpg"></div>  <div><img src="../img/04.jpg"></div>  <div><img src="../img/01.jpg"></div>  <div><img src="../img/02.jpg"></div>  <div><img src="../img/03.jpg"></div>  <div><img src="../img/04.jpg"></div>  <div><img src="../img/01.jpg"></div>  <div><img src="../img/02.jpg"></div>  <div><img src="../img/03.jpg"></div>  <div><img src="../img/04.jpg"></div>  <div><img src="../img/01.jpg"></div>  <div><img src="../img/02.jpg"></div>  <div><img src="../img/03.jpg"></div>  <div><img src="../img/04.jpg"></div>  <div><img src="../img/01.jpg"></div>  <div><img src="../img/02.jpg"></div>  <div><img src="../img/03.jpg"></div>  <div><img src="../img/04.jpg"></div>  <div><img src="../img/01.jpg"></div>  <div><img src="../img/02.jpg"></div>  <div><img src="../img/03.jpg"></div>  <div><img src="../img/04.jpg"></div>  </div>  </body>  </html> |

### 4.4.7 列表样式

**list-style-type属性用于控制列表中列表项标志的样式。**

-无序列表：出现在各列表项旁边的圆点

-有序列表：可能是字母、数字或另外某种计数体系中的一个符号

1. 无序列表取值：

none：无标记

disc：实心圆，默认值

circle：空心圆

square：实心方块

1. 有序列表取值：

none：无标记

decimal：数字（如1,2,3,4,5），为默认值

lower-roman：小写罗马数字（如I,ii,iii,iv,v）

upper-roman：大写罗马数字（如I，II，III，IV，V）

**list-style-image属性使用图像来替换列表项的标记。**

取值为：url( )，指定图像作为有序或无序列表的标志。

### 4.4.8 显示方式

#### 4.4.8.1 默认显示方式

元素都有自己默认的显示方式，包括：

1. 块元素

从上至下，可以设置宽高，比如<h1>、<p>、<div>等

1. 行内元素

从左至右，不能设置宽高，比如<span>、<a>等

1. 行内块

从左至右，可以设置宽高，比如：<input>、<img>等

#### 4.4.8.2 改变显示方式

除了默认显示的效果之外，可以使用display属性来修改元素框的显示方式，即改变生成框的类型。

display属性用于改变元素的显示方式， 其取值有：

1. none：不显示该元素，释放其占用位置
2. block：将元素显示方式设置为块，如可以将行内元素<a>转换为块元素
3. inline：将元素显示方式设置为行内，如可以将块元素<p>转换为行内元素
4. inline-block：将元素显示方式设置为行内块，如可以将行内元素<span>转换为行内块元素。

### 4.4.9 鼠标形状

默认情况下，光标会根据用户的操作发生改变：

1. 当鼠标悬停在一个链接上时，光标将从指针形状变为手形状
2. 当鼠标悬停在文本区域时，会显示“I”形状
3. 当鼠标悬停在一个按钮上时，光标会显示为箭头

可以使用cursor属性指定显示给用户的鼠标光标类型（形状），这种形状能够给用户提供一种可视化的暗示，提示可以进行的操作。

cursor属性定义了鼠标指针放在一个元素边界范围内时所用的光标形状，可取值：

|  |  |
| --- | --- |
| default | 默认光标（通常是一个箭头） |
| crosshair | 光标呈现为十字线。 |
| pointer | 光标呈现为指示链接的指针（一只手） |
| text | 此光标指示文本。 |
| wait | 此光标指示程序正忙（通常是一只表或沙漏）。 |
| help | 此光标指示可用的帮助（通常是一个问号或一个气球）。 |

# 5.案例练习

## 5.1 太极

补充知识点：

border-radius: 50%;设置圆角

|  |
| --- |
| 在正方形div中，当圆角超过50%时，会成圆  div{  width: 100px;  height: 100px;  border: 1px solid red;  border-radius: 50%;  } |

太极代码实现：

|  |
| --- |
| 1.绘制背景  <div class="content">  <div class="left"></div>  <div class="rught"></div>  </div>  body {  margin: 0px;  background-color: #ccc;  }    .content {  height: 400px;  width: 400px;  margin: auto;  background-color: white;  }    2.接下来画一个圆  .content {  height: 400px;  width: 400px;  margin: auto;  background-color: white;  /\*画圆\*/  position:relative;  border-radius: 50%;  }    3.将圆画成左右两个颜色不同的半圆  .content:before,.content:after {  width:200px;  height:400px;  position:absolute;  top:0;  display:block;  content:"";  }  .content:before {  left:0;  border-radius:200px 0 0 200px;  background-color:#000;  }  .content:after {  right:0;  border-radius:0 200px 200px 0;  background-color:#fff;  }    4.画上面的圆  .left,.right{  width:200px;  height:200px;  position:absolute;  z-index:2;  border-radius:375px;  }  .left{  top:0;  left:100px;  background-color:#000;  }    5.同理，画下面的圆  .right{  bottom:0;  right:100px;  background-color:#fff;  }    6. 在下面小圆里面画小黑色圆  .left:after,.right:after{  width:50px;  height:50px;  z-index:3;  position:absolute;  display:block;  content:"";  border-radius:100px;  }  .right:after{  bottom:75px;  right:75px;  background-color:#000;  }    7. 同理，在上面的圆里画一个小圆  .left:after{  top:75px;  left:75px;  background-color:#fff;  } |