

# 01-HTML 和 CSS

讲师：王振国

## 今日任务

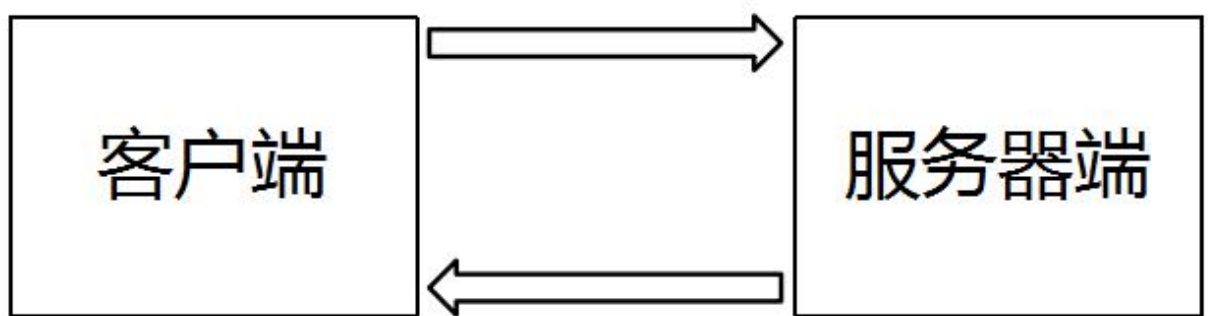
- 1、html 学习
- 2、CSS 学习

## 课堂笔记：

### 1、B/S 软件的结构

JavaSE   C/S   Client Server  
B/S   Browser   Server

#### B/S 软件的结构

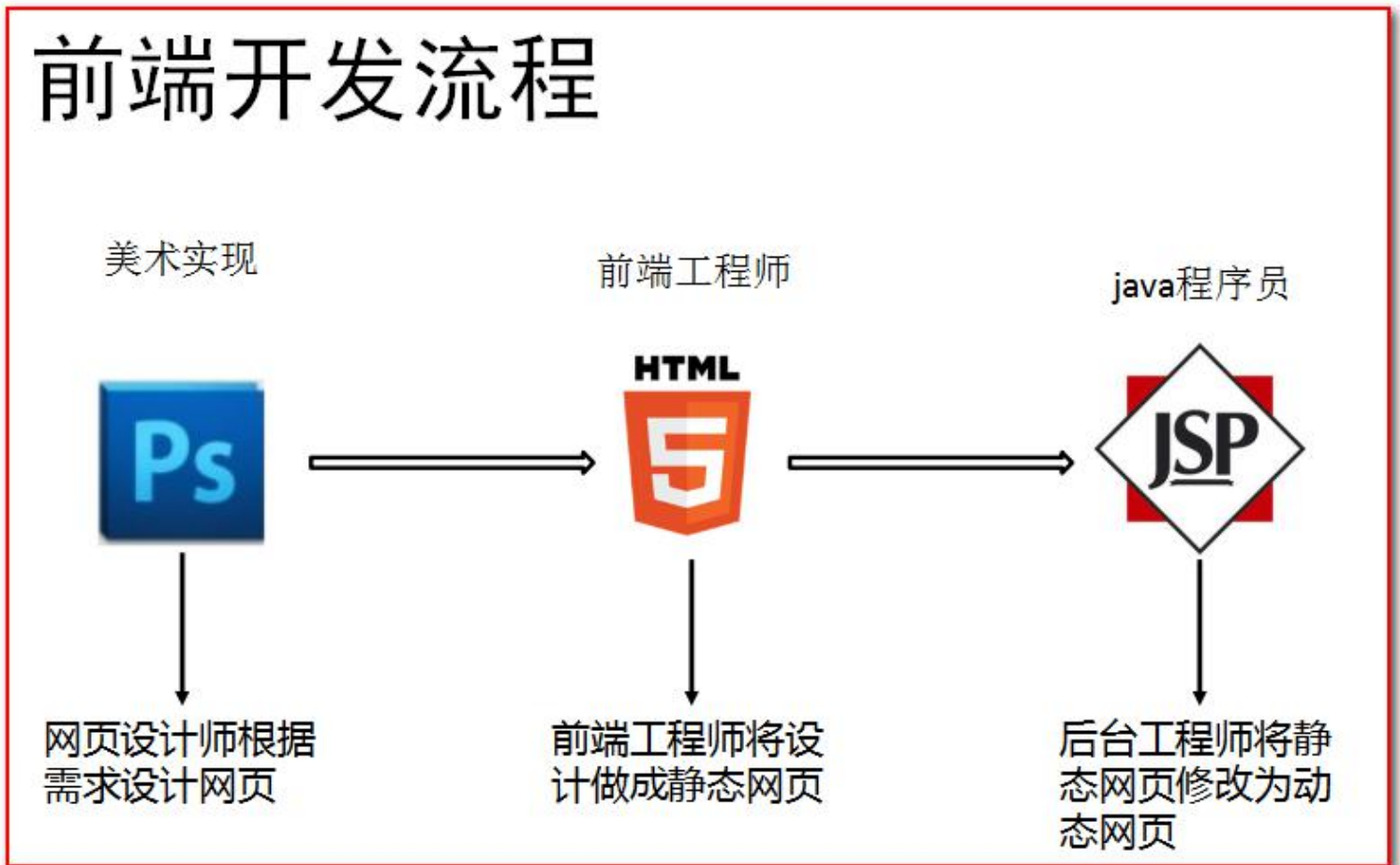


JavaEE项目

客户端：浏览器

服务器端：WEB服务器

## 2、前端的开发流程



## 3、网页的组成部分

页面由三部分内容组成！

分别是内容（结构）、表现、行为。

内容（结构），是我们在页面中可以看到的数据。我们称之为内容。一般内容 我们使用 html 技术来展示。

表现，指的是这些内容在页面上的展示形式。比如说。布局，颜色，大小等等。一般使用 CSS 技术实现

行为，指的是页面中元素与输入设备交互的响应。一般使用 javascript 技术实现。

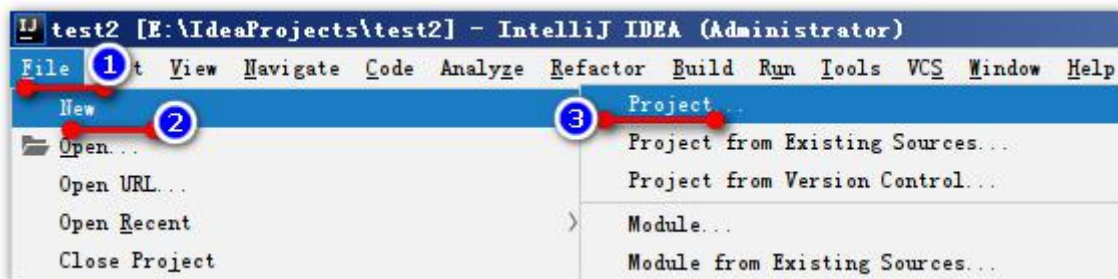
## 4、HTML 简介

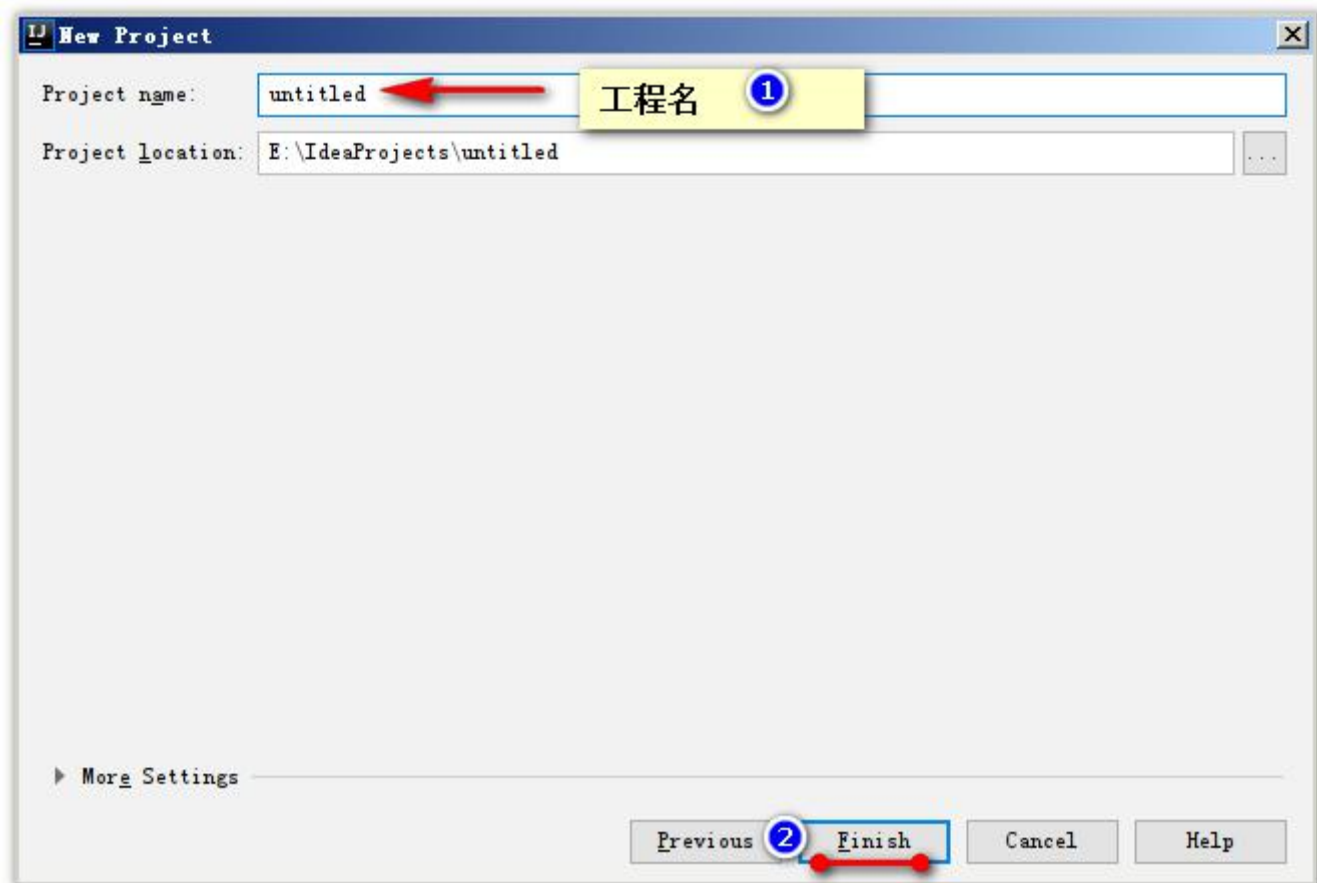
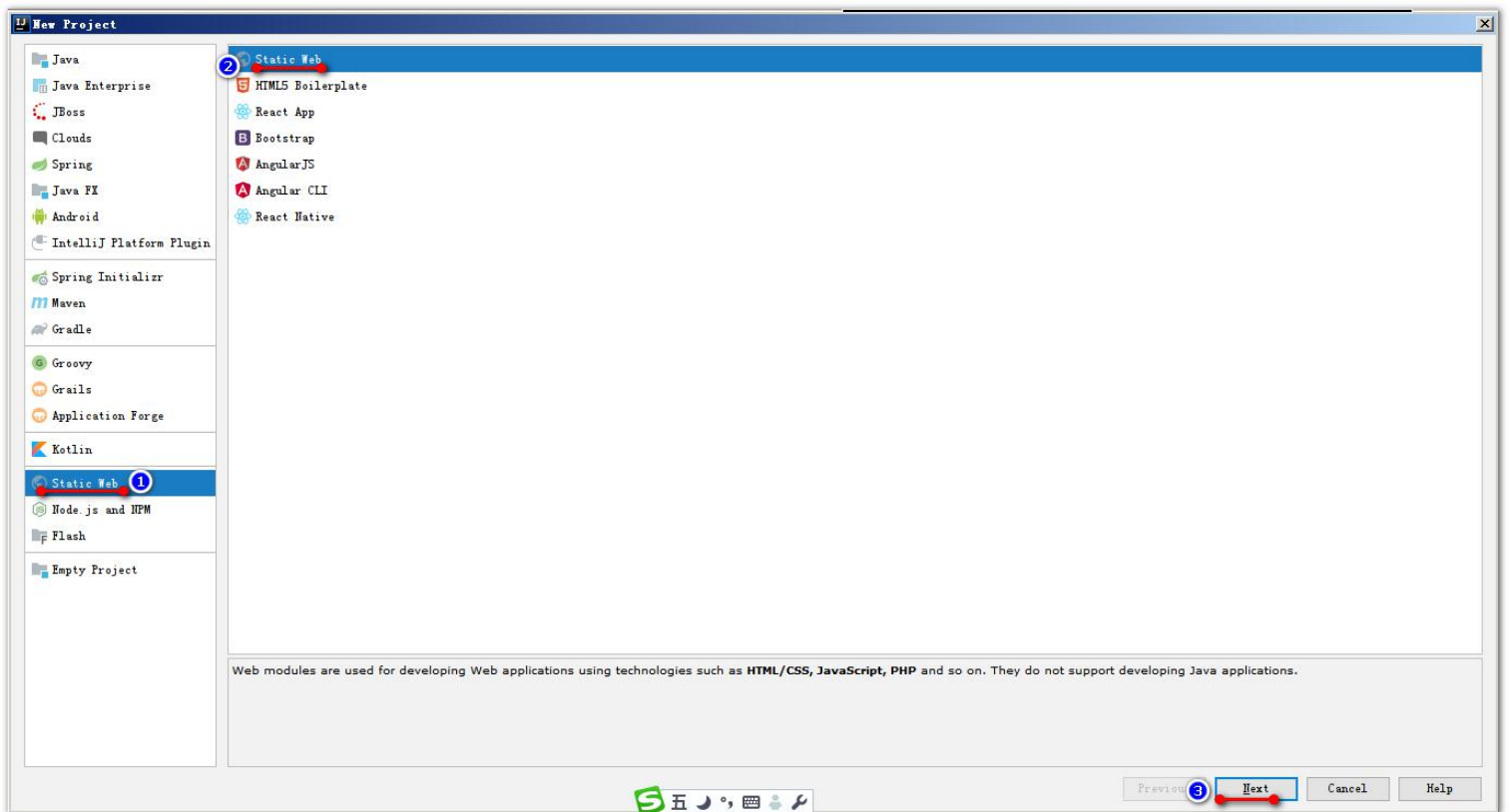
Hyper Text Markup Language （超文本标记语言） 简写：HTML

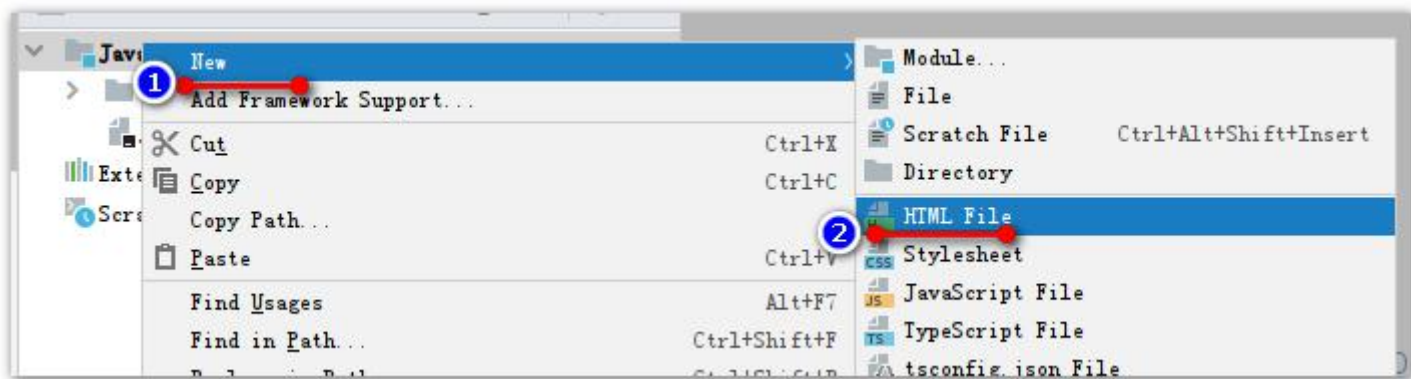
HTML 通过标签来标记要显示的网页中的各个部分。网页文件本身是一种文本文件，通过在文本文件中添加标记符，可以告诉浏览器如何显示其中的内容（如：文字如何处理，画面如何安排，图片如何显示等）

## 5、创建 HTML 文件。

1、创建一个 web 工程（静态的 web 工程）







选择浏览器执行页面



第一个 html 示例:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>标题</title>
</head>
<body>
  hello
</body>
</html>
```

**注:** Java 文件是需要先编译，再由 java 虚拟机跑起来。但 HTML 文件它不需要编译，直接由浏览器进行解析执行。

## 6、HTML 文件的书写规范

<html>	表示整个 html 页面的开始
<head>	头信息
<title>标题</title>	标题
</head>	
<body>	body 是页面的主体内容
页面主体内容	
</body>	
</html>	表示整个 html 页面的结束

**Html 的代码注释**    <!-- 这是 [html](#) 注释，可以在页面右键查看源代码中看到 -->

## 7、HTML 标签介绍

### 1. 标签的格式:

`<标签名>封装的数据</标签名>`

### 2. 标签名大小写不敏感。

### 3. 标签拥有自己的属性。

i. 分为基本属性: `bgcolor="red"`

可以修改简单的样式效果

ii. 事件属性: `onclick="alert('你好!');"`

可以直接设置事件响应后的代码。

### 4. 标签又分为, 单标签和双标签。

i. 单标签格式: `<标签名 />`

`br` 换行      `hr` 水平线

ii. 双标签格式: `<标签名> ...封装的数据...</标签名>`

### 双标签

`<p>`      `</p>`

开始标签

结束标签

### 单标签

`<br />`

自结束标签

``

属性

### 标签的语法:

`<!-- @标签不能交叉嵌套 -->`

正确: `<div><span>早安, 尚硅谷</span></div>`

错误: `<div><span>早安, 尚硅谷</div></span></div>`

`<!-- @标签必须正确关闭 -->`

`<!-- i. 有文本内容的标签: -->`



正确: `<div>早安, 尚硅谷</div>`

错误: `<div>早安, 尚硅谷`

`<hr />`

`<!-- ii.没有文本内容的标签: -->`

正确: `<br />`

错误: `<br>`

`<hr />`

`<!-- ③属性必须有值, 属性值必须加引号 -->`

正确: `<font color="blue">早安, 尚硅谷</font>`

错误: `<font color=blue>早安, 尚硅谷</font>`

错误: `<font color>早安, 尚硅谷</font>`

`<hr />`

`<!-- ④注释不能嵌套 -->`

正确: `<!-- 注释内容 --> <br/>`

错误: `<!-- <!-- 这是错误的 html 注释 --> -->`

`<hr />`

#### 注意事项:

1.html 代码不是很严谨。有时候标签不闭合, 也不会报错。

## 8、常用标签介绍 文档: w3cschool.CHM

### 8.1、font 字体标签

需求 1: 在网页上显示 我是字体标签 , 并修改字体为 宋体, 颜色为红色。

```
<body>
```

```
<!-- 字体标签
```

```
需求 1: 在网页上显示 我是字体标签 , 并修改字体为 宋体, 颜色为红色。
```

```
font 标签是字体标签, 它可以用来修改文本的字体, 颜色, 大小(尺寸)
```

```
color 属性修改颜色
```

```
face 属性修改字体
```

```
size 属性修改文本大小
```

```
-->
```

```
<font color="red" face="宋体" size="7">我是字体标签</font>
```

```
</body>
```

需求 1: 把 `<br>` 换行标签 变成文本 转换成字符显示在页面上

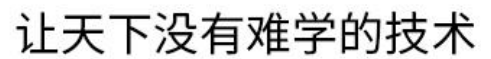
## 最常用的字符实体

显示结果	描述	实体名称	实体编号
	空格	&nbsp;	&#160;
<	小于号	&lt;	&#60;
>	大于号	&gt;	&#62;
&	和号	&amp;	&#38;
"	引号	&quot;	&#34;
'	撇号	&apos; (IE不支持)	&#39;

显示结果	描述	实体名称	实体编号
¢	分	&cent;	&#162;
£	镑	&pound;	&#163;
¥	日圆	&yen;	&#165;
§	节	&sect;	&#167;
©	版权	&copy;	&#169;
®	注册商标	&reg;	&#174;
×	乘号	&times;	&#215;
÷	除号	&divide;	&#247;

[illegible]





&lt;/body&gt;

```
<body>
<!-- a 标签是 超链接
      href 属性设置连接的地址
      target 属性设置哪个目标进行跳转
      _self      表示当前页面(默认值)
      _blank     表示打开新页面来进行跳转
-->
```

```
<a href="http://localhost:8080">百度</a><br/>
<a href="http://localhost:8080" target="_self">百度_self</a><br/>
<a href="http://localhost:8080" target="_blank">百度_blank</a><br/>
</body>
```

## 8.5、列表标签

无序列表 、 有序列表

需求 1：使用无序，列表方式，把东北 F4，赵四，刘能，小沈阳，宋小宝，展示出来

```
<body>
<!-- 需求 1：使用无序，列表方式，把东北 F4，赵四，刘能，小沈阳，宋小宝，展示出来
    ul 是无序列表
        type 属性可以修改列表项前面的符号
    li 是列表项
-->
<ul type="none">
    <li>赵四</li>
    <li>刘能</li>
    <li>小沈阳</li>
    <li>宋小宝</li>
</ul>
</body>
```

## 8.6、img 标签

img 标签可以在 html 页面上显示图片。

需求 1：使用 img 标签显示一张美女的照片。并修改宽高，和边框属性

```
<body>
<!-- 需求 1：使用 img 标签显示一张美女的照片。并修改宽高，和边框属性

    img 标签是图片标签, 用来显示图片
        src 属性可以设置图片的路径
        width 属性设置图片的宽度
        height 属性设置图片的高度
        border 属性设置图片边框大小
        alt 属性设置当指定路径找不到图片时, 用来代替显示的文本内容
```

在 JavaSE 中路径也分为相对路径和绝对路径。

相对路径: 从工程名开始算

绝对路径: 盘符:/ 目录/ 文件名

在 web 中路径分为相对路径和绝对路径两种

相对路径:

- 表示当前文件所在的目录
- .. 表示当前文件所在的上一级目录
- 文件名 表示当前文件所在目录的文件, 相当于 ./ 文件名      ./ 可以省略

绝对路径:

正确格式是: http://ip:port/ 工程名/ 资源路径

错误格式是: 盘符:/ 目录/ 文件名

-->

```






```

</body>

## 8.7、表格标签 ( \*\*\*\* 重点，必须掌握 \* )



需求 1: 做一个 带表头的 ， 三行， 三列的表格， 并显示边框

需求 2: 修改表格的宽度， 高度， 表格的对齐方式， 单元格间距。

```
<body>
```

```
<!--
```

需求 1: 做一个 带表头的 ， 三行， 三列的表格， 并显示边框

需求 2: 修改表格的宽度， 高度， 表格的对齐方式， 单元格间距。

table 标签是表格标签

border 设置表格边框

width 设置表格宽度

height 设置表格高度

align 设置表格相对于页面的对齐方式

cellspacing 设置单元格间距

tr 是行标签

th 是表头标签

td 是单元格标签

align 设置单元格文本对齐方式

b 是加粗标签

-->

```
<table align="center" border="1" width="300" height="300" cellspacing="0">
```

```
<tr>
```

```
<th>1.1</th>
```

```
<th>1.2</th>
```

```
<th>1.3</th>
```

```
</tr>
```

```
<tr>
```

```
<td>2.1</td>
```

```
<td>2.2</td>
```

```
<td>2.3</td>
```

```
</tr>
```

```
<tr>
```

```
<td>3.1</td>
```

```
<td>3.2</td>
```

```
<td>3.3</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

8.8、跨行跨列表格 （\* 次重点，必须掌握 \*）



需求 1：新建一个五行，五列的表格，第一行，第一列的单元格要跨两列，第二行第一列的单元格跨两行，第四行第四列的单元格跨两行两列。

```
<body>
<!-- 需求 1:
    新建一个五行，五列的表格，
    第一行，第一列的单元格要跨两列，
    第二行第一列的单元格跨两行，
    第四行第四列的单元格跨两行两列。

    colspan 属性设置跨列
    rowspan 属性设置跨行
-->

<table width="500" height="500" cellspacing="0" border="1">
    <tr>
        <td colspan="2">1.1</td>
        <td>1.3</td>
        <td>1.4</td>
        <td>1.5</td>
    </tr>
    <tr>
        <td>2.1</td>
        <td>2.2</td>
        <td>2.3</td>
        <td>2.4</td>
    </tr>
    <tr>
        <td>3.1</td>
        <td>3.2</td>
        <td>3.3</td>
        <td>3.4</td>
    </tr>
    <tr>
        <td>4.1</td>
        <td>4.2</td>
        <td colspan="2">4.3</td>
    </tr>
    <tr>
        <td>5.1</td>
        <td>5.2</td>
        <td>5.3</td>
        <td>5.4</td>
    </tr>
</table>
```



```
<td rowspan="2">2.1</td>
<td>2.2</td>
<td>2.3</td>
<td>2.4</td>
<td>2.5</td>
</tr>
<tr>
<td>3.2</td>
<td>3.3</td>
<td>3.4</td>
<td>3.5</td>
</tr>
<tr>
<td>4.1</td>
<td>4.2</td>
<td>4.3</td>
<td colspan="2" rowspan="2">4.4</td>
</tr>
<tr>
<td>5.1</td>
<td>5.2</td>
<td>5.3</td>
</tr>
</table>
</body>
```

## 8.9、了解 iframe 框架标签 (内嵌窗口)

iframe 标签它可以在一个 html 页面上,打开一个小窗口,去加载一个单独的页面.

```
<body>
我是一个单独的完整的页面<br/><br/>
<!--iframe 标签可以在页面上开辟一个小区域显示一个单独的页面
      iframe 和 a 标签组合使用的步骤:
          1 在 iframe 标签中使用 name 属性定义一个名称
          2 在 a 标签的 target 属性上设置 iframe 的 name 的属性值
-->
<iframe src="3.标题标签.html" width="500" height="400" name="abc"></iframe>
<br/>

<ul>
<li><a href="0-标签语法.html" target="abc">0-标签语法.html</a></li>
<li><a href="1.font 标签.html" target="abc">1.font 标签.html</a></li>
<li><a href="2.特殊字符.html" target="abc">2.特殊字符.html</a></li>
```

```
</ul>
</body>
```

## 8.10、表单标签 （ \*\*\*\*\* 重点 ， 必须掌握 \* ）

什么是表单？

表单就是 html 页面中,用来收集用户信息的所有元素集合.然后把这些信息发送给服务器.



用户名称: \* 1 用户名必须由字母、数字、下划线组成, 并且长度为8-15长度

用户密码:

确认密码:

选择性别: ☐ 男 ☐ 女

选择爱好: ☐ C ☐ C++ ☐ Java ☐ PHP ☐ .net

选择国家: --请选择国家--

自我评价:

注册 重置

需求 1: 创建一个个人信息注册的表单界面。包含用户名，密码，确认密码。性别（单选），兴趣爱好（多选），国籍（下拉列表）。

隐藏域，自我评价（多行文本域）。重置，提交。

### 表单的显示:

```
<body>
```

```
<!--需求 1: 创建一个个人信息注册的表单界面。包含用户名，密码，确认密码。性别（单选），兴趣爱好（多选），国籍（下拉列表）。隐藏域，自我评价（多行文本域）。重置，提交。-->
```

```
<!--
```

```
form 标签就是表单
```

```
input type=text 是文件输入框 value 设置默认显示内容
```

```
input type=password 是密码输入框 value 设置默认显示内容
```

```
input type=radio 是单选框 name 属性可以对其进行分组 checked="checked"表示默认选中
```

```
input type=checkbox 是复选框 checked="checked"表示默认选中
input type=reset 是重置按钮 value 属性修改按钮上的文本
input type=submit 是提交按钮 value 属性修改按钮上的文本
input type=button 是按钮 value 属性修改按钮上的文本
input type=file 是文件上传域
input type=hidden 是隐藏域 当我们要发送某些信息，而这些信息，不需要用户参与，就可以使用隐藏域（提交的时候同时发送给服务器）
```

```
select 标签是下拉列表框
option 标签是下拉列表框中的选项 selected="selected"设置默认选中
```

```
textarea 表示多行文本输入框 （起始标签和结束标签中的内容是默认值）
rows 属性设置可以显示几行的高度
cols 属性设置每行可以显示几个字符宽度
```

```
-->
```

```
<form>
```

```
用户名称: <input type="text" value="默认值"/><br/>
用户密码: <input type="password" value="abc"/><br/>
确认密码: <input type="password" value="abc"/><br/>
性别: <input type="radio" name="sex"/>男<input type="radio" name="sex" checked="checked" />女<br/>
兴趣爱好: <input type="checkbox" checked="checked" />Java<input type="checkbox" />JavaScript<input type="checkbox" />C++<br/>
```

```
国籍: <select>
```

```
<option>--请选择国籍--</option>
<option selected="selected">中国</option>
<option>美国</option>
<option>小日本</option>
</select><br/>
```

```
自我评价: <textarea rows="10" cols="20">我才是默认值</textarea><br/>
<input type="reset" value="abc" />
<input type="submit"/>
```

```
</form>
```

```
</body>
```

## 表单格式化:

```
<form>
```

```
<h1 align="center">用户注册</h1>
```

```
<table align="center">
```

```
<tr>
```

```
<td> 用户名称: </td>
```

```
<td>
```

```
<input type="text" value="默认值"/>
```

```
</td>
```

```
</tr>
```

```
<tr>
```

```
<td> 用户密码: </td>
<td><input type="password" value="abc"/></td>
</tr>
<tr>
<td>确认密码: </td>
<td><input type="password" value="abc"/></td>
</tr>
<tr>
<td>性别: </td>
<td>
<input type="radio" name="sex"/>男
<input type="radio" name="sex" checked="checked" />女
</td>
</tr>
<tr>
<td> 兴趣爱好: </td>
<td>
<input type="checkbox" checked="checked" />Java
<input type="checkbox" />JavaScript
<input type="checkbox" />C++
</td>
</tr>
<tr>
<td>国籍: </td>
<td>
<select>
<option>--请选择国籍--</option>
<option selected="selected">中国</option>
<option>美国</option>
<option>小日本</option>
</select>
</td>
</tr>
<tr>
<td>自我评价: </td>
<td><textarea rows="10" cols="20">我才是默认值</textarea></td>
</tr>
<tr>
<td><input type="reset" /></td>
<td align="center"><input type="submit"/></td>
</tr>
</table>
</form>
</body>
```

## 表单提交细节:

```
<body>
  <!--
    form 标签是表单标签
      action 属性设置提交的服务器地址
      method 属性设置提交的方式 GET(默认值)或 POST

    表单提交的时候，数据没有发送给服务器的三种情况：
      1、表单项没有 name 属性值
      2、单选、复选（下拉列表中的 option 标签）都需要添加 value 属性，以便发送给服务器
      3、表单项不在提交的 form 标签中

    GET 请求的特点是：
      1、浏览器地址栏中的地址是：action 属性[+?+请求参数]
        请求参数的格式是：name=value&name=value
      2、不安全
      3、它有数据长度的限制

    POST 请求的特点是：
      1、浏览器地址栏中只有 action 属性值
      2、相对于 GET 请求要安全
      3、理论上没有数据长度的限制

  -->
  <form action="http://localhost:8080" method="post">
    <input type="hidden" name="action" value="login" />
    <h1 align="center">用户注册</h1>
    <table align="center">
      <tr>
        <td> 用户名称： </td>
        <td>
          <input type="text" name="username" value="默认值"/>
        </td>
      </tr>
      <tr>
        <td> 用户密码： </td>
        <td><input type="password" name="password" value="abc"/></td>
      </tr>
      <tr>
        <td>性别： </td>
        <td>
          <input type="radio" name="sex" value="boy"/>男
          <input type="radio" name="sex" checked="checked" value="girl" />女
        </td>
      </tr>
      <tr>
        <td> 兴趣爱好： </td>
        <td>
```

```

        <input name="hobby" type="checkbox" checked="checked" value="java"/>Java
        <input name="hobby" type="checkbox" value="js"/>JavaScript
        <input name="hobby" type="checkbox" value="cpp"/>C++
    </td>
</tr>
<tr>
    <td>国籍: </td>
    <td>
        <select name="country">
            <option value="none">--请选择国籍--</option>
            <option value="cn" selected="selected">中国</option>
            <option value="usa">美国</option>
            <option value="jp">小日本</option>
        </select>
    </td>
</tr>
<tr>
    <td>自我评价: </td>
    <td><textarea name="desc" rows="10" cols="20">我才是默认值</textarea></td>
</tr>
<tr>
    <td><input type="reset" /></td>
    <td align="center"><input type="submit"/></td>
</tr>
</table>
</form>
</body>

```

## 8.11 其他标签

需求 1: div、span、p 标签的演示

```

<body>
    <!--需求 1: div、span、p 标签的演示
    div 标签        默认独占一行
    span 标签        它的长度是封装数据的长度
    p 段落标签        默认会在段落的上方或下方各空出一行来（如果已有就不再空）
    -->
    <div>div 标签 1</div>
    <div>div 标签 2</div>
    <span>span 标签 1</span>
    <span>span 标签 2</span>
    <p>p 段落标签 1</p>
    <p>p 段落标签 2</p>
</body>

```



HTML 最后的练习 2: 创建登录的表单, 包含用户名, 密码框输入。并结合 table 标签排版。让它看上去整齐点

## 9、CSS 技术 文档:CSS2.0.chm

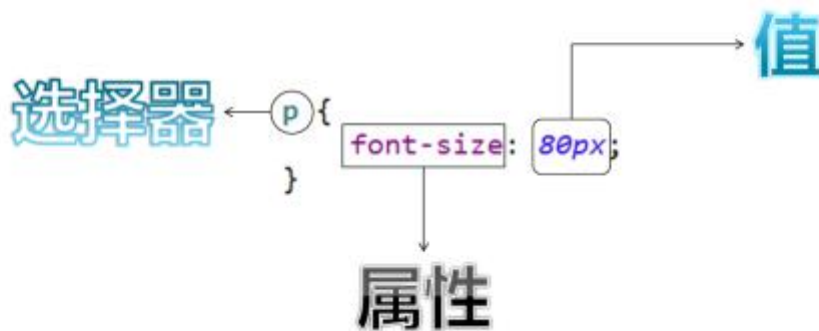
### 9.1、CSS 技术介绍

CSS 是「层叠样式表单」。是用于(增强)控制网页样式并允许将样式信息与网页内容分离的一种标记性语言。

### 9.2、CSS 语法规则:



#### 2、语法规则



**选择器:** 浏览器根据“选择器”决定受 CSS 样式影响的 HTML 元素（标签）。

**属性 (property)** 是你想要改变的样式名，并且每个属性都有一个值。属性和值被冒号分开，并由花括号包围，这样就组成了一个完整的样式声明（**declaration**），例如：p {color: blue}

**多个声明:** 如果要定义不止一个声明，则需要用分号将每个声明分开。虽然最后一条声明的最后可以不加分号(但尽量在每条声明的末尾都加上分号)

例如:

```
p{
    color:red;
```

```
font-size:30px;
```

```
}
```

注：一般每行只描述一个属性

CSS 注释：/\*注释内容\*/

## 9.3、CSS 和 HTML 的结合方式

### 9.3.1、第一种：

在标签的 style 属性上设置"key:value value;"，修改标签样式。

需求 1：分别定义两个 div、span 标签，分别修改每个 div 标签的样式为：边框 1 个像素，实线，红色。

```
<div>div 标签 1</div>
<div>div 标签 2</div>
<span>span 标签 1</span>
<span>span 标签 2</span>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <!--需求 1：分别定义两个 div、span 标签，分别修改每个 div 标签的样式为：边框 1 个像素，实线，红色。-->
  <div style="border: 1px solid red;">div 标签 1</div>
  <div style="border: 1px solid red;">div 标签 2</div>
  <span style="border: 1px solid red;">span 标签 1</span>
  <span style="border: 1px solid red;">span 标签 2</span>
</body>
</html>
```

问题：这种方式的缺点？

- 1.如果标签多了。样式多了。代码量非常庞大。
- 2.可读性非常差。
- 3.Css 代码没什么复用性可方言。

### 9.3.2、第二种：

在 head 标签中，使用 style 标签来定义各种自己需要的 css 样式。

格式如下：

```
xxx {  
    Key : value value;  
}
```

需求 1：分别定义两个 div、span 标签，分别修改每个 div 标签的样式为：边框 1 个像素，实线，红色。

```
<div>div 标签 1</div>  
<div>div 标签 2</div>  
<span>span 标签 1</span>  
<span>span 标签 2</span>
```

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <title>Title</title>  
    <!--style 标签专门用来定义 css 样式代码-->  
    <style type="text/css">  
        /* 需求 1：分别定义两个 div、span 标签，分别修改每个 div 标签的样式为：边框 1 个像素，实线，红色。*/  
        div{  
            border: 1px solid red;  
        }  
        span{  
            border: 1px solid red;  
        }  
    </style>  
</head>  
  
<body>  
    <div>div 标签 1</div>  
    <div>div 标签 2</div>  
  
    <span>span 标签 1</span>  
    <span>span 标签 2</span>  
  
</body>  
</html>
```

Css 注释 /\* 这是 css 的代码注释 \*/

问题：这种方式的缺点。

1. 只能在同一页面内复用代码，不能在多个页面中复用 css 代码。
2. 维护起来不方便，实际的项目中会有成千上万的页面，要到每个页面中去修改。工作量太大了。

### 9.3.3、第三种：

把 css 样式写成一个单独的 css 文件，再通过 link 标签引入即可复用。

使用 html 的 `<link rel="stylesheet" type="text/css" href="./styles.css" />` 标签 导入 css 样式文件。

1、css 文件内容：

```
div{
    border: 1px solid yellow;
}
span{
    border: 1px solid red;
}
```

html 文件代码：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
    <!--link 标签专门用来引入 css 样式代码-->
    <link rel="stylesheet" type="text/css" href="1.css"/>
</head>
<body>
    <div>div 标签 1</div>
    <div>div 标签 2</div>

    <span>span 标签 1</span>
    <span>span 标签 2</span>
</body>
</html>
```

## 9.4、CSS 选择器

### 9.4.1、标签名选择器

标签名选择器的格式是：

```
标签名{  
    属性: 值;  
}
```

标签名选择器，可以决定哪些标签被动的使用这个样式。

```
<div>div 标签 1</div>  
<div>div 标签 2</div>  
<span>span 标签 1</span>  
<span>span 标签 2</span>
```

需求 1：在所有 div 标签上修改字体颜色为蓝色，字体大小 30 个像素。边框为 1 像素黄色实线。  
并且修改所有 span 标签的字体颜色为黄色，字体大小 20 个像素。边框为 5 像素蓝色虚线。

示例代码：

```
<!DOCTYPE html>  
<html>  
<head>  
    <meta charset="UTF-8">  
    <title>CSS 选择器</title>  
    <style type="text/css">  
        div{  
            border: 1px solid yellow;  
            color: blue;  
            font-size: 30px;  
        }  
        span{  
            border: 5px dashed blue;  
            color: yellow;  
            font-size: 20px;  
        }  
    </style>  
</head>  
<body>  
    <!--
```

需求 1：在所有 div 标签上修改字体颜色为蓝色，字体大小 30 个像素。边框为 1 像素黄色实线。

并且修改所有 span 标签的字体颜色为黄色，字体大小 20 个像素。边框为 5 像素蓝色虚线。

```
-->
<div>div 标签 1</div>
<div>div 标签 2</div>
<span>span 标签 1</span>
<span>span 标签 2</span>
</body>
</html>
```

## 9.4.2、id 选择器

id 选择器的格式是：

```
#id 属性值{
    属性: 值;
}
```

id 选择器，可以让我们通过 id 属性选择性的去使用这个样式。

需求 1：分别定义两个 div 标签，

第一个 div 标签定义 id 为 id001，然后根据 id 属性定义 css 样式修改字体颜色为蓝色，字体大小 30 个像素。边框为 1 像素黄色实线。

第二个 div 标签定义 id 为 id002，然后根据 id 属性定义 css 样式 修改的字体颜色为红色，字体大小 20 个像素。边框为 5 像素蓝色点线。

```
<div id="id001">div 标签 1</div>
<div id="id002">div 标签 2</div>
```

示例代码：

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>ID 选择器</title>
    <style type="text/css">

        #id001{
            color: blue;
            font-size: 30px;
            border: 1px yellow solid;
```



```
}

#id002{
    color: red;
    font-size: 20px;
    border: 5px blue dotted ;
}

</style>
</head>
<body>
    <!--
    需求 1: 分别定义两个 div 标签,
    第一个 div 标签定义 id 为 id001 , 然后根据 id 属性定义 css 样式修改字体颜色为蓝色,
    字体大小 30 个像素。边框为 1 像素黄色实线。

    第二个 div 标签定义 id 为 id002 , 然后根据 id 属性定义 css 样式 修改的字体颜色为红色, 字体大小 20 个像素。
    边框为 5 像素蓝色点线。
    -->

    <div id="id002">div 标签 1</div>
    <div id="id001">div 标签 2</div>
</body>
</html>
```

### 9.4.3、class 选择器（类选择器）

class 类型选择器的格式是：

```
.class 属性值{
    属性: 值;
}
```

class 类型选择器，可以通过 class 属性有效的选择性地使用这个样式。

需求 1: 修改 class 属性值为 class01 的 span 或 div 标签，字体颜色为蓝色，字体大小 30 个像素。边框为 1 像素黄色实线。  
需求 2: 修改 class 属性值为 class02 的 div 标签，字体颜色为灰色，字体大小 26 个像素。边框为 1 像素红色实线。

```
<div class="class01">div 标签 class01</div>
<div class="class02">div 标签</div>
<span class="class01">span 标签 class01</span>
<span>span 标签 2</span>
```

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>class 类型选择器</title>
  <style type="text/css">
    .class01{
      color: blue;
      font-size: 30px;
      border: 1px solid yellow;
    }

    .class02{
      color: grey;
      font-size: 26px;
      border: 1px solid red;
    }
  </style>
</head>
<body>
  <!--
    需求 1: 修改 class 属性值为 class01 的 span 或 div 标签, 字体颜色为蓝色, 字体大小 30 个像素。边框为 1 像素黄色实线。
    需求 2: 修改 class 属性值为 class02 的 div 标签, 字体颜色为灰色, 字体大小 26 个像素。边框为 1 像素红色实线。
  -->

  <div class="class02">div 标签 class01</div>
  <div class="class02">div 标签</div>
  <span class="class02">span 标签 class01</span>
  <span>span 标签 2</span>
</body>
</html>
```

## 9.4.4、组合选择器

组合选择器的格式是：

```
选择器 1, 选择器 2, 选择器 n{
  属性: 值;
}
```

组合选择器可以让多个选择器共用同一个 css 样式代码。

```
<div class="class01">div 标签 class01</div> <br />
<span id="id01">span 标签</span> <br />
```

```
<div>div 标签</div> <br />
<div>div 标签 id01</div> <br />
```

需求 1: 修改 `class="class01"` 的 `div` 标签 和 `id="id01"` 所有的 `span` 标签, 字体颜色为蓝色, 字体大小 20 个像素。边框为 1 像素黄色实线。

示例代码:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>class 类型选择器</title>
  <style type="text/css">
    .class01 , #id01{
      color: blue;
      font-size: 20px;
      border: 1px yellow solid;
    }
  </style>
</head>
<body>
  <!--
  需求 1: 修改 class="class01" 的 div 标签 和 id="id01" 所有的 span 标签,
  字体颜色为蓝色, 字体大小 20 个像素。边框为 1 像素黄色实线。
  -->

  <div id="id01">div 标签 class01</div> <br />
  <span>span 标签</span> <br />
  <div>div 标签</div> <br />
  <div>div 标签 id01</div> <br />
</body>
</html>
```

## 9.5、常用样式:

### 1、字体颜色

`color: red;`

颜色可以写颜色名如: black, blue, red, green 等

颜色也可以写 rgb 值和十六进制表示值: 如 `rgb(255,0,0)`, `#00F6DE`, 如果写十六进制值必须加#

## 2、宽度

`width:19px;`

宽度可以写像素值：19px;

也可以写百分比值：20%;

## 3、高度

`height:20px;`

高度可以写像素值：19px;

也可以写百分比值：20%;

## 4、背景颜色

`background-color:#0F2D4C`

## 4、字体样式：

`color: #FF0000;` 字体颜色红色  
`font-size: 20px;` 字体大小

## 5、红色 1 像素实线边框

`border: 1px solid red;`

## 7、DIV 居中

`margin-left: auto;`  
`margin-right: auto;`

## 8、文本居中：

`text-align: center;`

## 9、超连接去下划线

```
text-decoration: none;
```

## 10、表格细线

```
table {  
    border: 1px solid black; /*设置边框*/  
    border-collapse: collapse; /*将边框合并*/  
}  
td,th {  
    border: 1px solid black; /*设置边框*/  
}
```

## 11、列表去除修饰

```
ul {  
    list-style: none;  
}
```

示例代码:

```
<!DOCTYPE html>  
<html>  
<head>  
    <meta charset="UTF-8">  
    <title>06-css 常用样式.html</title>  
    <style type="text/css">  
  
        div{  
            color: red;  
            border: 1px yellow solid;  
            width: 300px;  
            height: 300px;  
            background-color: green;  
            font-size: 30px;  
            margin-left: auto;  
            margin-right: auto;  
            text-align: center;  
        }  
        table{  
            border: 1px red solid;  
            border-collapse: collapse;  
        }  
  
        td{
```

```
        border: 1px red solid;
    }

    a{
        text-decoration: none;
    }

    ul{
        list-style: none;
    }

</style>
</head>
<body>
    <ul>
        <li>1111111111</li>
        <li>1111111111</li>
        <li>1111111111</li>
        <li>1111111111</li>
        <li>1111111111</li>
    </ul>
    <table>
        <tr>
            <td>1.1</td>
            <td>1.2</td>
        </tr>
    </table>
    <a href="http://www.baidu.com">百度</a>
    <div>我是 div 标签</div>
</body>
</html>
```





