



Web 前端技术简介

动态 Web 项目由动态页面构成。动态页面是在静态页面的基础上加入处理数据的高级语言代码演化而来。静态页面在动态 Web 项目中起到人机交互的作用。具体来说,首先静态页面会提供图形界面让客户在控件中输入待处理的数据。然后,静态页面会提供合适的图形控件触发一个请求,此请求会包含待处理的数据,并且此请求会发给处理这些数据的程序。最后,程序处理完请求后又需要用静态页面来对处理结果进行可视化展示。因此,在动态 Web 项目中静态页面不可或缺,只有通过静态页面才能图形化地触发请求,提交数据,显示请求处理结果。而静态网页需要用 Web 前端技术实现,因此本章将简单介绍 Web 前端技术中 HTML 语言、CSS 语言、JavaScript 语言和 JQuery 技术。

学习目标

- (1) 了解 HTML、CSS、JavaScript 和 JQuery 的含义和作用。
- (2) 理解 CSS 代码规则、CSS 选择器,在 HTML 代码中引入 CSS。
- (3) 理解 JavaScript 代码放置方式、JavaScript 语法、JQuery 编码。
- (4) 掌握 HTML 标签中 URL 的求解方法和表单标签的作用。
- (5) 掌握在 Web 项目中如何提交数据。

3.1 HTML

互联网技术出现后,人们可以通过电子邮件、互传文件等方式进行信息交流,使得信息交流更加及时、方便。但还是急需一个在互联网中对交流的信息进行集中管理、可视化展示的技术,HTML 由此应运而生。

HTML 的英文全称是 Hyper Text Markup Language,即超文本标记语言。HTML 是由 Web 的发明者 Tim Berners-Lee 和同事 Daniel W. Connolly 于 1990 年创立的一种标记语言。自 1990 年以来,HTML 就一直被用作互联网的信息展示语言,使用 HTML 语言编写的文件需要通过互联网浏览器才能显示出效果。HTML 是一种建立网页文件的语言,通过标记式的指令(Tag,即标签),将文本、声音、图片、影视等内容以可视化方式显示出来。

在 Web 应用开发中,在需求阶段可以用 HTML 来制作项目的静态原型网站以确认需求。在编码阶段可以将 HTML 页面直接作为最终图形用户界面,并在 HTML 页面中加入高级语言代码进而将静态页面转化为动态页面。

3.1.1 HTML 标签及其分类

1. HTML 标签的结构

HTML 语言通过标签将文本、声音、图片、影视等信息进行可视化展示。HTML 中每个

标签都由开始标签、结束标签、标签体和属性构成,如图 3-1 所示。



图 3-1 HTML 标签的构成

1) 开始标签

开始标签以“<”开头,以“>”结尾,中间放置标签名。

2) 结束标签

结束标签以“</”开头,以“>”结尾,中间放置标签名。

3) 标签体

标签体是开始标签和结束标签之间的内容,即需要进行可视化展示的信息。

4) 属性

属性用于提供标签的更多信息,例如,标签体的格式、布局等。属性总是以属性名称/属性值对的形式出现,要求放置在开始标签中。在 HTML 中属性分为全局属性和专用属性。全局属性是指所有标签都具有的属性,专用属性是指某个标签特有的属性。

HTML 的全局属性分为标准全局属性和事件全局属性。标准全局属性用于修饰标签体,赋予 HTML 元素(标签)特定的意义和语境,而事件全局属性用于捕获和处理 HTML 元素上触发的事件。HTML 全局属性的详细内容请扫描右侧的二维码进行自学。



自学资料

所有 HTML 标签都必须有开始标签,但是结束标签、标签体和属性有些标签有,有些标签没有。



注意: 在 HTML 5 中不再支持 HTML 4.01 中的布局属性。建议使用 CSS 来为 HTML 标签指定布局。

2. HTML 标签的两种视图

HTML 标签有两种视图(见图 3-2):一是标签源代码视图,编码人员使用此视图在 HTML 网页文件中设计、编码 Web 页面;二是运行结果视图,用户在互联网浏览器中解释运行此标签所在 HTML 页面时,展示给用户的结果图形界面。

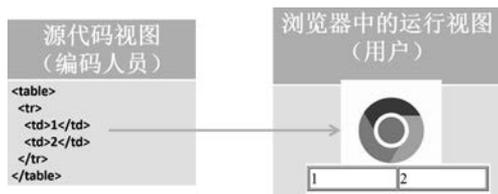


图 3-2 HTML 标签的两种视图

3. HTML 标签分类

HTML 语言主要用于编写网页,而网页是一种用户图形界面,主要用于人机交互,即用户用图形界面将数据提交给后台程序,后台程序处理完数据后将结果显示在图形界面上。因此可以将 HTML 中的常用标签分为两类:一类是显示数据的标签;另一类是提交数据的标签。每

类标签中又包含很多具体标签,如图 3-3 所示。

提交数据类标签将在 3.1.4 节详细讲解。显示数据类标签根据 HTML 元素控件的不同有很多不同的标签,详细内容请扫描右侧的二维码进行自学。



自学资料

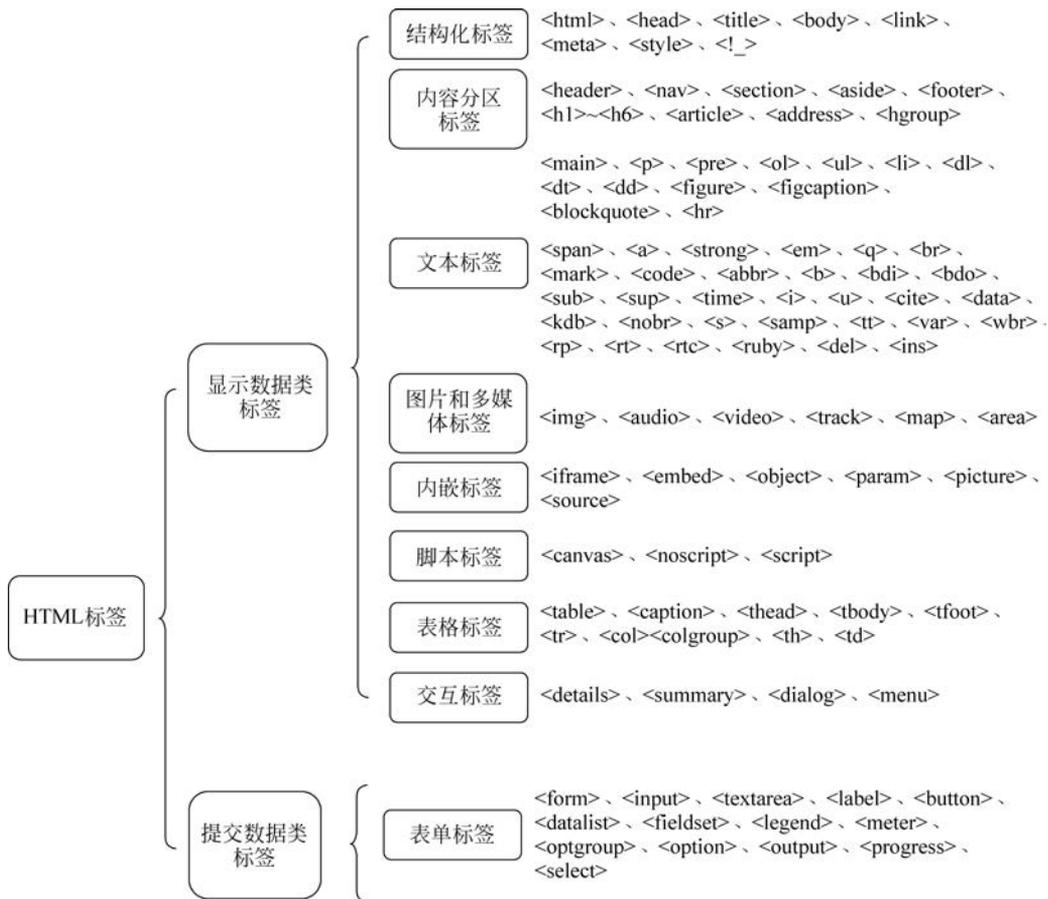


图 3-3 HTML 标签分类

3.1.2 HTML5 源代码结构

HTML 语言最新版本是 HTML5, HTML5 源代码标准结构如图 3-4 所示。在 HTML5 源代码标准结构中, 首先第一行用 <!DOCTYPE html> 标签表示当前文件的文档类型为 HTML; 然后用 <html> 标签作为网页的根标签; 然后在 <html> 标签内部, 用 <head> 标签表示 HTML 文档的头部信息, 用 <body> 标签表示 HTML 文档的主体信息。在 <head> 标签内部可以包含 <title>、<link>、<meta>、<script>、<style> 等全局公共标签来指定整个 HTML 的整体信息。

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>网页标题</title>
  </head>
  <body>
    网页要显示的内容
  </body>
</html>
  
```

头部信息

主体信息

图 3-4 HTML5 源代码标准结构

3.1.3 HTML 标签中的 URL

很多 HTML 标签的属性值是 URL, 例如 、<a>、<link>、<iframe> 等。URL (Uniform Resource Locator) 即统一资源定位符, 用于唯一确定互联网上的某个资源。在 Web 项目中有两类 URL: 一类是绝对 URL; 另一类是相对 URL。



视频讲解

1. 绝对 URL

绝对 URL 由协议名 (http://)、Web 服务器路径 (服务器 IP: 端口号) 和资源在服务器内部的路径构成, 各部分用 URL 分隔符“/”分隔。例如, 访问 127.0.0.1 上 Tomcat(8080 端口) 中 infoSubSys 项目里的 login.html 的绝对 URL 为“http://localhost:8080/infoSubSys/login.html”。绝对 URL 的优点是 HTML 既能请求当前项目内部的资源, 又能请求当前项目外部的资源; 缺点是请求当前项目内部资源时, 每个 URL 中都有相同的服务器路径和项目路径, 而且服务器路径和项目路径都是硬编码, 导致当服务器路径或项目路径改变时, 所有 URL 都要修改。鉴于绝对 URL 在请求项目内部资源时的缺点, 就出现了相对 URL。

2. 相对 URL

相对 URL 就是省略了绝对 URL 中协议名和 Web 服务器路径这两个部分而得到的 URL 路径。相对 URL 有两种写法: 一种是以“/”开头的相对路径; 另一种是不以“/”开头的路径。由于这两种相对 URL 的相对起点不同, 导致相对 URL 的求解方法也不一样。

1) 以“/”开头的相对 URL

以“/”开头的相对 URL, 其相对起点为 Web 服务器路径, 即 http://服务器地址: 端口号。例如, http://localhost:8080。以“/”开头的相对 URL 的求解步骤如下。

(1) 求解出终点 (需要访问的资源的绝对 URL), 例如 http://localhost:8080/infoSubSys/images/portrait.jpg。

(2) 终点一起点, 即所需要访问的资源的绝对 URL 与服务器 URL 的差, 例如/infoSubSys/images/portrait.jpg。

2) 不以“/”开头的相对 URL

不以“/”开头的相对 URL, 其相对起点为当前请求目录。

求解当前请求目录的步骤如下。

(1) 获得当前请求。当前请求就是访问相对 URL 所在的 HTML 页面的那个请求。例如, http://localhost:8080/infoSubSys/login.html。

(2) 获得当前请求后, 从当前请求 URL 中删除末尾的资源名。例如, http://localhost:8080/infoSubSys/。

求解出当前请求目录后, 不以“/”开头的相对 URL 的求解步骤如下。

例如, 在 http://localhost:8080/infoSubSys/pages/URL.html 中访问 http://localhost:8080/infoSubSys/images/portrait.jpg。

(1) 求解出起点 (当前请求目录), 得到 http://localhost:8080/infoSubSys/pages。

(2) 求解出终点 (需要访问的资源的绝对 URL), 得到 http://localhost:8080/infoSubSys/images/portrait.jpg。

(3) 求起点、终点的相同部分 (交叉点), 得到 http://localhost:8080/infoSubSys。

(4) 从起点到交叉点, 得到../。“./”表示进入当前目录, “../”表示后退进入上一级目录。

(5) 在步骤(4)的路径的基础上追加终点和交叉点之差, 得到../images/portrait.jpg。

3. 编写 URL 的建议

通过前面对绝对 URL 和相对 URL 的讲解可知相对 URL 只能访问项目内部资源, 要访问外部资源必须要用绝对路径。通过前面对相对 URL 的求解可知, 以“/”开头的相对 URL 的相对起点是固定的, 因此求解十分简单, 而不以“/”开头的相对 URL 的相对起点是随当前请求变化的, 因此求解十分麻烦。综上所述, 今后在 Web 项目中编写 URL 的建议

如下。

- (1) 访问当前项目外部的资源,必须用绝对路径。
- (2) 访问当前项目中的内部资源,建议用以“/”开头的相对 URL。

3.1.4 数据提交标签

用 HTML 语言编写的静态页面不仅能用显示标签来对请求处理结果进行可视化展示,更重要的是要提供图形控件让客户输入待处理的数据,并将数据放在一个请求中提交给后台程序进行处理。而在 HTML 语言中数据提交标签就是用来提交数据的,包括超链接标签和表单标签。超链接标签既可以显示数据又可以提交数据,在前面已有讲解,此处不再赘述。本节重点讲解表单标签。

1. 数据提交统一格式

在 HTML 中,不管是用表单标签提交数据,还是用超链接提交数据,它们提交数据的常用格式都是“被请求资源的 URL?参数名 1=值 1& 参数名 2=值 2”。例如,在下面的 URL “http://localhost:8080/infoSubSys/postAndGet/registerInfo.jsp?name=ZhangXiaohua&login_name=zxh&password=123456”中,向 registerInfo.jsp 提交了 3 个数据,分别是 ZhangXiaohua、zxh 和 123456,提交这 3 个数据的参数名分别为 name、login_name 和 password。

2. 表单标签

表单标签用于提供让用户输入数据的图形控件,并将用户输入的数据进行提交。表单标签包括 form 标签、select 标签、option 标签、textarea 标签、input 标签。因为所有表单标签都要提交数据,因此每个表单标签基本上都有 name 和 value 属性。value 属性的值就是要提交的数据,而 name 属性的值就是所提交数据的参数名。下面具体介绍每个表单标签。

1) form 标签

form 标签用于声明表单,而且其他表单标签只能在其范围内才能随表单传输数据。其常用的属性如表 3-1 所示。

表 3-1 form 标签常用属性

属性名	含义
action	其属性值用于指定处理表单数据的应用程序的位置,是一个 URL
method	其属性值用于指定传输表单数据的方式,其常用属性值有 get 或 post
enctype	其属性值用于指定在发送表单数据之前如何对其进行编码,可选择的值有 application/x-www-form-urlencoded、multipart/form-data 和 text/plain,其中默认值是 application/x-www-form-urlencoded

2) select 标签

select 标签用于声明下拉列表控件,让用户可以用选择数据的方式输入数据。其常用属性如表 3-2 所示。

表 3-2 select 标签常用属性

属性名	含义
name	其属性值用于指定传输数据的参数名
size	其属性值用于指定可见的下拉列表项个数,默认可见一个下拉列表项
multiple	multiple 属性是个标识属性,标识下拉列表可同时选择多个选项,multiple 属性可以有 "multiple" 属性值也可以无属性值。对于 Windows 系统来说,可按住 Ctrl 键来选择多个选项;对于 Mac 系统来说,可按住 command 键来选择多个选项

在 select 标签中没有 value 属性,因此 select 标签单独使用是没办法提交数据的,它必须和 option 标签联合使用,并将 option 标签作为 select 标签的子标签,表示下拉列表的列表项。

3) option 标签

option 标签用于指定下拉列表控件中的列表项,其常用属性如表 3-3 所示。

表 3-3 option 标签常用属性

属 性 名	含 义
value	其属性值用于指定此选项被选中时要提交的数据值
selected	selected 属性是个标识属性,标识此选项被默认选中,selected 属性可以有"selected"属性值也可以无属性值

4) textarea 标签

textarea 标签用于声明多行文本控件,让用户可以输入多行文本。其常用属性如表 3-4 所示。

表 3-4 textarea 标签常用属性

属 性 名	含 义
name	其属性值用于指定传输数据的参数名
cols	其属性值用于指定一行可看见多少个字符。如果数据行中的字符数超过 cols 属性值,那么多行文本控件会自动附带水平滑条
rows	其属性值用于指定可以显示多少行。如果数据行数超过 rows 属性值,那么多行文本控件会自动附带垂直滑条



注意: textarea 标签没有 value 属性,其提交的数据值就是 textarea 标签体的内容,也就是用户在多行文本控件中输入的数据。

5) input 标签

input 标签以不同的 type 属性值表示不同的输入控件,这些控件都有 name 和 value 属性,其中 value 属性的值就是要提交的数据,而 name 属性的值就是所提交数据的参数名。下面将详细介绍各个输入控件。

(1) 单行文本框。

当 input 标签的 type 属性值为 text 时,input 标签表示单行文本框,可以在单行文本框中录入一行数据,并且数据是明文显示的。单行文本框常用属性如表 3-5 所示。

表 3-5 单行文本框常用属性

属 性 名	含 义
name	其属性值用于指定传输数据的参数名
value	其属性值用于指定传输数据的数据值
size	其属性值用于指定单行文本框中可见的文本字符数
maxlength	其属性值用于指定单行文本框能录入的最多字符数

(2) 密码框。

当 input 标签的 type 属性值为 password 时,input 标签表示密码框,可以在密码框中录入密码,并且密码数据会用掩码显示。密码框的常用属性名和含义与单行文本框一样,此处不再赘述。

(3) 隐藏域。

当 input 标签的 type 属性值为 hidden 时,input 标签表示隐藏域。隐藏域控件不会显示

在 HTML 页面的运行结果界面中,但其源代码会出现在 HTML 页面的源代码中。隐藏域只有 name 和 value 属性。

隐藏域主要用于提交那些必须要提交,但是又不想让用户看到的数据,例如,数据记录的序号等。

(4) 单选按钮。

当 input 标签的 type 属性值为 radio 时,input 标签表示单选按钮。对于 name 属性值相同的多个单选按钮来说,只能选择其中的一个。

单选按钮的常用属性如表 3-6 所示。

表 3-6 单选按钮的常用属性

属 性 名	含 义
name	其属性值用于指定传输数据的参数名
value	其属性值用于指定传输数据的数据值
checked	checked 属性是个标识属性,标识此单选按钮被默认选中,checked 属性可以有"checked"属性值也可以无属性值

(5) 复选按钮。

当 input 标签的 type 属性值为 checkbox 时,input 标签表示复选按钮。name 属性值相同的多个复选按钮可以选择多个。

复选按钮的常用属性与单选按钮一致,此处不再赘述。



注意: 具有相同 name 属性值的多个复选按钮提交数据时,被选中的多个复选按钮的值(即每个复选按钮的 value 属性值)将会用同一个参数名(即每个复选按钮的 name 属性值)提交。

(6) 提交按钮。

当 input 标签的 type 属性值为 submit 时,input 标签表示提交按钮。从功能上理解提交按钮要把握 4 个要点:一是当用户在 HTML 运行结果页面单击提交按钮时,提交按钮会自动提交数据;二是提交按钮会自动提交其所在表单中所有表单控件的数据(即数据参数名为控件的 name 属性值,数据值为控件的 value 属性值);三是提交按钮会将数据提交给其所在表单 action 属性值所指定的资源;四是提交按钮提交数据时的传输方式由其所在表单的 method 属性值决定。

(7) 复位按钮。

当 input 标签的 type 属性值为 reset 时,input 标签表示复位按钮。复位按钮不能自动提交表单的数据,其主要作用是当客户单击复位按钮时,会自动将其所在表单中所有表单控件的 value 属性值设置为初始值。

(8) 普通按钮。

当 input 标签的 type 属性值为 button 时,input 标签表示普通按钮。普通按钮也不能自动提交表单的数据,如果要提交其所在表单的数据就要用事件捕获属性来捕获并处理事件。事件捕获属性名格式为 onXXX,其中 XXX 表示事件名,例如,鼠标的单击事件为 click,那么其捕获属性名为 onclick。事件捕获属性的值通常是一个 JavaScript 函数调用,此 JavaScript 函数包含一段处理此事件的 JavaScript 代码。JavaScript 语言将在 3.3 节介绍。

普通按钮一般用于将表单中的数据提交到多个资源进行处理,如果只提交到一个资源,则用提交按钮最简单。

(9) 图片按钮。

当 input 标签的 type 属性值为 image 时, input 标签表示图片按钮。图片按钮的作用和普通按钮一样,只是在按钮上面可以链接一张图片,因此图片按钮比普通按钮多了个 src 属性,该属性用于指定要链接的图片的 URL。

(10) 文件域。

当 input 标签的 type 属性值为 file 时, input 标签表示文件域。当用户单击文件域控件时,会弹出一个文件资源浏览窗口来选择将要上传的本地文件,但是不会自动提交本地文件数据。如果要提交本地文件数据,那么就需要用提交按钮触发一个请求来提交,而且还要求表单的 method 属性值必须是 post、enctype 属性值必须是 multipart/form-data。

3.1.5 在 Web 项目中提交数据

动态 Web 项目由动态页面构成,而动态页面的核心是处理请求中的数据,因此请求中必须要有数据。在动态 Web 项目中,数据通常是由客户随请求提供的。因此在动态 Web 项目中提交数据非常重要,本节将它作为一个专题进行详细讲解。

在 Web 项目中,前端页面可以用超链接和表单来提交数据,但是不管是用超链接还是表单,都需要触发一个 HTTP 请求,并将数据放到此 HTTP 请求中才能提交数据,而且提交数据的常用格式都是相同的,都是“被请求资源的 URL?参数名 1=值 1&参数名 2=值 2”。因此要掌握 Web 项目中的数据提交就必须先了解 HTTP 请求。

1. HTTP 请求

一个 HTTP 请求的内容由请求行、请求头和数据体 3 部分构成,如图 3-5 所示。



图 3-5 HTTP 请求内容

1) 请求行

请求行对当前 HTTP 请求进行了概要说明,其中指定了请求的方法、请求目标资源的 URL、请求的协议名和协议版本号。

2) 请求头

请求头中指定了发出请求的客户端浏览器的一些属性数据,每个属性数据都用“属性名:属性值”的方式进行指定。

3) 数据体

数据体指定了 POST 请求传递的数据,GET 请求将数据放到了请求行中,例如,“GET/infoSubSys/postAndGet/registerInfo.jsp?login_name=zxh&password=123456&submi1=



视频讲解

%E6%8F%90%E4%BA%A4 HTTP/1.1”。

了解了 HTTP 请求后,下面将分别讲解用表单提交数据和用超链接提交数据。

2. 用表单提交数据

用表单提交数据是指将提交的数据放在一个表单的表单控件中,提交时,表单控件的数据会随着表单一起提交,提交的数据就是每个表单控件的 value 属性值,数据的参数名就是每个表单控件的 name 属性值。因此用表单提交数据应满足一些条件:要有表单,表单中要有表单控件,表单中要有触发请求的控件(如按钮)。表单数据提交方法通常有 GET 和 POST。

1) 用表单提交数据的 GET/POST 方式

表单中的数据传输有两种方式:一种是 GET 方式,另一种是 POST 方式。

用表单提交数据的 GET 方式指表单的 method 属性值为 GET,其特点是将参数数据放入 HTTP 请求的请求行中,导致浏览器的地址栏中会显示完整的请求字符串,包括请求资源和参数数据。之所以在浏览器的地址栏中会显示数据,是因为 HTTP 请求的请求行会显示在地址栏,而 GET 请求恰好就将数据放到了请求行。同时,因为浏览器对请求字符串的长度进行了限制,因此 GET 无法提交大量的数据。

用表单提交数据的 POST 方式指表单的 method 属性值为 POST,其特点是将参数数据放入 HTTP 请求的数据体,而数据体内容不会显示在浏览器地址栏中,而且数据体没有数据大小的限制。

因此常常将 GET 方式提交数据称为明文传输数据,而 POST 方式提交数据称为暗文传输数据。

2) 提交表单数据的请求的触发实现

表单中的数据要放到 HTTP 请求中才能提交,而请求则是需要用户操作界面中的控件来触发。

有两种常用方式来触发请求提交表单中的数据:一种是通过表单 action 属性值和 submit 按钮实现;另一种是通过 JavaScript 代码和普通按钮实现,即在普通按钮上将单击事件捕获属性的值指定为一个 JavaScript 函数,而在 JavaScript 函数中通过代码来提交表单数据。示例代码如下:

```
<input type = "button" value = "按钮" onclick = "register()">

function register()
{
var thisForm = document.forms[0];
thisForm.action = "registerInfo. jsp";
}
```

3. 用超链接提交数据

除了可以用表单提交数据外,还可以用超链接提交数据。

用超链接提交数据指用超链接标签<a>来提交数据。需要特别注意的是,超链接提交数据只能是 GET 方式,而没有 POST 方式。

用超链接提交数据也要触发请求,触发方式也有两种:一种是用超链接标签<a>的 href 属性值实现,例如,当被单击时,提交数据 zxh、zhangxiaohua,对应的参数名分别是 name 和 login_name;另一种也是通过 JavaScript 代码来实现,即在超链接上将单击事件捕获属性的值指定为一个 JavaScript 函数,而在 JavaScript 函数中通过代码来模拟用超链接提交数据。示例代码如下。

```
<input type = "button" value = "按钮 2" onclick = "regist1()">

function regist1()
{
    location.href = "registerInfo.jsp?name = zxh&login_name = zhangxiaohua";
}
}
```

4. 总结及建议

通过前面的讲解,综合考虑数据提交的途径和传输方式,数据提交方式有以下 3 种。

- (1) 表单 GET 方式。
- (2) 表单 POST 方式。
- (3) 超链接 GET 方式。

考虑到数据传输的安全性和提交数据量的大小,建议今后做动态 Web 项目时,无特殊原因都采用表单 POST 方式来提交数据。

3.2 CSS 语言

HTML 标签原本被设计用于定义网页文档内容。网页文档布局由浏览器根据标签及其属性来完成,而不使用任何的格式化标签。但是随着网页越来越复杂,由浏览器根据标签和其属性来对文档进行布局越来越力不从心。CSS 技术由此出现。

CSS(Cascading Style Sheets,层叠样式表)是一种用来表现 HTML 文件样式的计算机语言。CSS 不仅可以以静态方式修饰网页,还可以配合各种脚本语言(如 JavaScript)动态地对网页各个元素进行格式化。

3.2.1 CSS 代码规则

每个 CSS 代码都由选择器、属性和属性值构成,如图 3-6 所示。

(1) CSS 选择器用于“查找”(或选取)要设置样式的 HTML 元素。

(2) CSS 属性与 CSS 属性值经常配对使用,用于设置控件的一个样式。其中 CSS 属性表示此样式的名称,CSS 属性值表示此样式的具体值。CSS 属性可以分为字体属性、文本属性、背景属性、布局属性、边界属性、列表项目属性、表格属性。

CSS 属性的详细内容请扫描右侧的二维码进行自学。

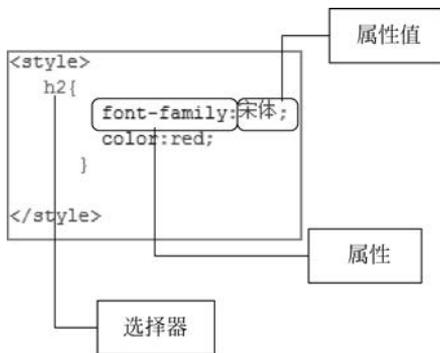


图 3-6 CSS 代码的构成



自学资料

3.2.2 CSS 选择器

CSS 选择器用于“查找”(或选取)要设置样式的 HTML 元素。CSS 选择器根据其功能可以分为以下五大类。

- (1) 简单选择器(根据标签名称、id 属性值、class 属性值来选取元素)。
- (2) 组合器选择器(将多个简单选择器组合在一起选取元素)。
- (3) 伪类选择器(根据特定状态来选取元素)。



自学资料

(4) 伪元素选择器(选取元素的一部分并设置其样式)。

(5) 属性选择器(根据属性或属性值来选取元素)。

本书只讲解简单选择器,其他类别的选择器请扫描左侧的二维码进行自学。

简单选择器这一类别包括标签选择器、类别选择器和 id 选择器,下面逐一介绍。

1. 标签选择器

标签选择器用来声明页面中哪些标签采用哪些 CSS 样式,标签选择器名就是要指定样式的标签的名字,例如,`a{font-size:9px; color:#F93;}`用来指定当前 HTML 页面中所有超链接标签字号都是 9 像素,字体颜色都是 #F93。

2. 类别选择器

类别选择器的书写以英文点号开头,后面紧跟着是类别名,即“`.类别名{}`”。类别选择器用于选择当前 HTML 页面中具有相同 class 属性值的所有控件,选择出这些控件后就能统一指定它们的样式,如图 3-7 所示。



图 3-7 类别选择器的使用示例

在图 3-7 中,左边代码用 CSS 定义了 3 个类别选择器样式,类别名分别为 one、two 和 three,其中每个类别样式都指定了文字字体、文字大小和文字颜色。

在图 3-7 中,中间代码将左边定义的 3 个类别样式用于一段 HTML 代码中,即在 HTML 标签中如果要引用类别样式,则只需把要引用的类别样式名作为此标签的 class 属性值即可。例如,图 3-7 中代码`<h2 class="one">应用了选择器 one </h2 >`。

在图 3-7 中,最右边是中间 HTML 代码块在浏览器中的运行结果。

3. id 选择器

id 选择器的书写形式是以英文 # 号开头,后面紧跟着是 id 名,即“`#id 名{}`”。id 选择器用于选择当前 HTML 页面中 id 属性值等于 id 样式名的某个控件。id 选择器的使用示例如图 3-8 所示。

在图 3-8 中,左边代码用 CSS 定义了 3 个 id 选择器样式,id 名分别为 first、second 和 three,其中每个 id 样式都指定了不同的文字大小。

在图 3-8 中,中间代码将左边定义的 3 个 id 样式用于一段 HTML 代码中,即在 HTML 标签中如果要引用 id 样式,则只需把要引用的 id 样式名作为此标签的 id 属性值即可。例如,图 3-8 中代码`<p id="first"> ID 选择器</p >`。

在图 3-8 中,最右边是中间 HTML 代码块在浏览器中的运行结果。

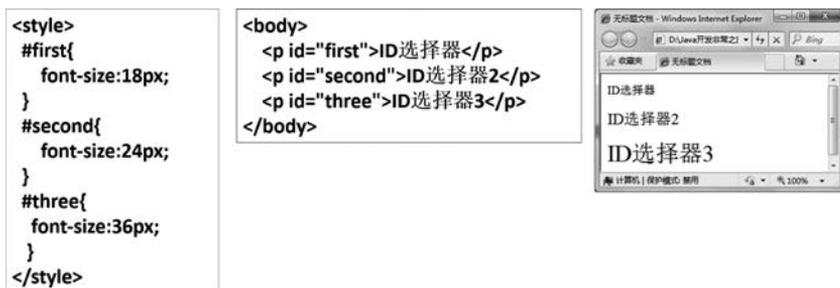


图 3-8 id 选择器的使用示例

3.2.3 在 HTML 代码中引入 CSS

如果 HTML 代码中要使用预先定义好的样式,那么首先要将 CSS 样式引入 HTML 页面。在 HTML 页面中引入 CSS 样式有 3 种方式,分别为链接式引入、内嵌式引入和行内定义式引入。

1. 链接式引入

当一些样式在多个 HTML 都要使用时,可以将这些样式的定义代码放在一个单独的 CSS 文件中(扩展名为 .css 的文件)。如果当前 HTML 页面要使用此 CSS 文件中的样式,那么必须将此 CSS 文件引入 HTML 页面,就要用到链接式引入。用链接方式引入的样式被称为链接样式。

链接式引入用 link 标签实现。例如,代码< link rel = "stylesheet" href = " test. css" type = "text/css">,引入了当前请求目录下的 test. css 样式文件。详细示例代码如图 3-9 所示。

在图 3-9 中,左边的 test. css 样式文件中定义了 3 个 id 样式 first、second、three,右边的 test. html 中用 link 标签将 test. css 样式文件引入后,在 3 个< p >标签中用 id 属性引用了 test. css 中的 3 个 id 样式。

2. 内嵌式引入

当一些样式只在当前 HTML 使用时,就可以使用< style >标签在当前页面中定义样式,样式定义好后只能在当前页面使用。用内嵌方式引入的样式被称为内嵌样式。详细示例代码如图 3-10 所示。

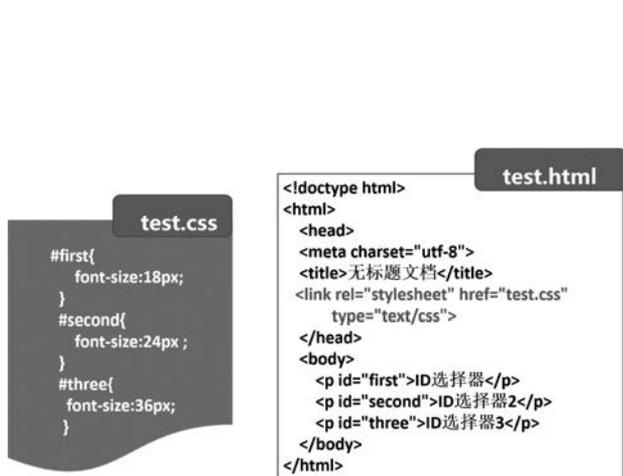


图 3-9 链接样式的使用示例



图 3-10 内嵌样式的使用示例

在如图 3-10 所示的 HTML 页面中, < style > 标签内部定义了 3 个 id 样式 first、second、three, 然后在 HTML 中的 3 个 < p > 标签中用 id 属性引用了这 3 个 id 样式。

3. 行内定义式引入

当一些样式只在当前标签使用时, 就可以使用当前标签的 style 属性来指定样式, 即将样式代码作为 style 属性值。用行内定义方式引入的样式被称为行内样式。详细示例代码如图 3-11 所示。

在如图 3-11 所示的代码中, 3 个 < p > 标签中用 style 属性值指定了本标签要使用的 CSS 样式。

4. 样式的优先级

当 HTML 页面的某个标签中, 链接样式、内嵌样式和行内样式都有定义时, 优先采用行内样式, 次之采用内嵌样式, 最后才是链接样式。

```
<table width="200" border="1" align="center">
  <tr>
    <td><p style="color:#F00; font-size:36px;">行内样式一</p></td>
  </tr>
  <tr>
    <td><p style="color:#F00; font-size:24px;">行内样式二</p></td>
  </tr>
  <tr>
    <td><p style="color:#F00; font-size:18px;">行内样式三</p></td>
  </tr>
  <tr>
    <td><p style="color:#F00; font-size:14px;">行内样式四</p></td>
  </tr>
</table>
```

图 3-11 行内样式的使用示例

3.3 JavaScript

用 HTML 和 CSS 制作的网页, 因为网页上的数据(包括标签、样式、显示的数据)都是硬编码的, 因此网页没有动态效果。要使得网页有动态效果, 就需要动态地更改网页的数据, 即需要一种能对网页中数据进行处理的语言, JavaScript 由此产生。

JavaScript 是一种基于对象和事件驱动并具有安全性的解释型脚本语言。它不需要进行编译, 而是直接嵌入在 HTML 页面中, 把静态页面转变成支持用户交互并响应事件的动态页面。JavaScript 具有解释性、基于对象、事件驱动、跨平台等特点。

用 JavaScript 编写的代码(Javascript 代码)被放置在 HTML 页面中, JavaScript 代码会被客户端浏览器解释运行, 而不会在 Web 服务器上运行。

3.3.1 JavaScript 代码放置方式

放置 JavaScript 代码有两种方式: 一种是内嵌式; 另一种是链接式。

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>内嵌式JS代码示例</title>
  </head>
  <body>
    <script language="javascript">
      var now=new Date();
      var hour=now.getHours();
      var minu=now.getMinutes();
      alert("现在是"+hour+"-"+minu);
    </script>
  </body>
</html>
```

图 3-12 内嵌式 JavaScript 代码示例

1. 内嵌式

如果 JavaScript 代码只在当前 HTML 页面使用, 则采用内嵌方式放置 JavaScript 代码, 即在当前 HTML 页面的 < script > 标签中编写 JavaScript 代码, 示例代码如图 3-12 所示。

2. 链接式

如果 JavaScript 代码要在多个 HTML 页面使用, 则采用链接式放置 JavaScript 代码, 即先将 JavaScript 代码放在一个公共 JS 文件中(扩展名为 .js 的文件), 然后在要运行此 JavaScript 代码的 HTML 页面中用 < script > 标签引入 JS 文件,

示例代码如图 3-13 所示。

```

<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <title>链接式JS代码示例</title>
  </head>
  <body>
    <script language="javascript" src="javascript.js">
    </script>
  </body>
</html>

```

图 3-13 链接式 JavaScript 代码示例

3.3.2 JavaScript 语法

JavaScript 语法和其他高级语言语法一样,分为面向数据处理过程的语法和面向对象语法,再加上事件处理。面向数据处理过程的语法包括数据(常量、变量)、数据存取(变量定义)、数据运算(表达式、程序流程控制)、代码复用(函数);面向对象语法包括对象定义和创建、内置对象。

JavaScript 语法和 Java 语法很相似,标识符都区分字母大小写,但是和 Java 语言又有以下不同。

- (1) JavaScript 每条语句结尾的分号可有可无,而 Java 每条语句必须以分号结尾。
- (2) JavaScript 是弱类型语言,而 Java 是强制类型语言。

JavaScript 在编码阶段无须定义数据的类型,在运行阶段由浏览器根据数据值动态指定数据类型;而在 Java 中首次使用数据时必须指定数据的类型。

- (3) JavaScript 是脚本语言,而 Java 是编译语言。

JavaScript 代码被浏览器解释运行,而不用像其他高级语言一样需要编译,因此 JavaScript 语言被划归到脚本语言类别。而 Java 代码必须编译成 Java 类才能运行。

JavaScript 语法的详细内容请扫描右侧二维码自学。



自学资料

3.4 JQuery 技术

JavaScript 语言出现后,就能用 JavaScript 代码对 HTML 页面进行处理,从而使 HTML 页面具有动态效果。但是用原生 JavaScript 编写代码效率比较低,因此出现了很多对原生 JavaScript 进行封装的 JavaScript 库(简称 JS 库)来提高 JavaScript 代码的编写效率。其中 JQuery 是目前最流行的 JS 库,它提供了大量的扩展,很多大公司都在使用 JQuery。

本书只对 JQuery 技术进行简要介绍,JQuery 技术的详细内容请扫描右侧二维码自学。

1. 导入 JQuery 库

在用 JQuery 库编写代码之前要从 JQuery 官方网站 <http://jquery.com> 下载 JQuery 库,然后将 JQuery 库导入到动态 Web 项目中,如图 3-14 所示。

2. JQuery 编码

使用 JQuery 库编码的通用流程如下。

(1) 首先使用 JQuery 选择器在当前 HTML 中查找出要操作的控件。JQuery 选择器的语法如下。

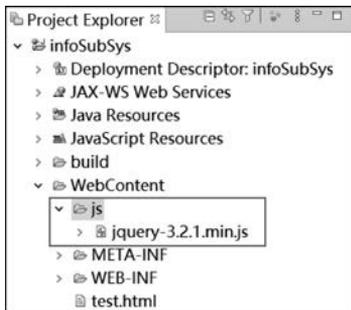


图 3-14 将 JQuery 库导入到动态 Web 项目



自学资料

```
$("#选择规则") 或 $(DOM对象)
```

(2) 然后可以针对查找出来的控件获取或修改控件的内容和属性值,获取或修改控件的样式值,指定控件的事件处理函数等。例如,下面的代码。

```
$("#[name = 'member']").focus(function(){
    $(this).addClass("input_focus");
});
```

在上面的代码中,首先从当前 HTML 页面选择出 name 属性值为 'member' 的控件,然后指定此控件的 focus 事件处理函数,最后在事件处理函数中给控件新增了一个名称为 "input_focus" 的类样式。

本章小结

本章简单讲解了 Web 前端核心技术,包括 HTML、CSS、JavaScript 和 JQuery。需重点掌握的是 HTML 标签中的 URL、表单标签、在 Web 项目如何提交数据,而其他内容只做了简单介绍,详细内容请读者扫描书中对应二维码自学。

读者学完本章内容后,就能看懂 UI 工程师设计的 UI 界面源代码,并在看懂源代码的情况下进行修改。

习题

一、单项选择题

- 在 HTML 标签中用于捕获和处理鼠标单击事件的属性是()。
 - onclick
 - click
 - ondblclick
 - onfocus
- img 标签中指定图片 URL 地址的属性是()。
 - src
 - href
 - action
 - file
- HTML 的超链接标签中指定连接资源 URL 地址的属性是()。
 - src
 - href
 - action
 - file
- input 标签中 type 属性值是()时表示隐藏域控件。
 - text
 - hidden
 - password
 - file
- 如果要用文件域上传文件,那么文件域所在 form 标签的 enctype 属性值必须是()。
 - text/plain
 - multipart/form-data
 - text/html
 - application/x-www-form-urlencoded
- CSS 简单选择器不包括()。
 - 标签选择器
 - 类别选择器
 - id 选择器
 - 属性选择器
- 当某个标签中链接样式、内嵌样式和行内样式都有定义时,优先级最高的是()。
 - 链接样式
 - 内嵌样式
 - 行内样式
 - 没有优先级
- 分析如下的 JavaScript 代码段,运行后在页面上输出()。


```
var c = "10", d = 10; document.write(c + d);
```

 - 10
 - 20
 - 1010
 - 页面报错

9. 下面在 JavaScript 中调用名称为 myFunction 的函数,正确的是()。

- A. call function myFunction B. call myFunction()
C. myFunction() D. 以上都不对

10. 以下选项中不能正确地得到标签的是()。

`<input id="btnGo" type="button" value="单击" class="btn">`

- A. `$("#btnGo")` B. `$(".btnGo")`
C. `$(".btn")` D. `$("#input[type='button']")`

二、判断题

1. `<a>` 标签定义超链接,用于从一个网页链接到另一个网页。 ()
2. name 属性规定了元素的唯一 id。 ()
3. `` 标签中的文字显示为斜体。 ()
4. `<table>` 标签定义 HTML 表格。 ()
5. ID 选择器与类选择器完全相同。 ()
6. JavaScript 对字母不区分大小写。 ()
7. JavaScript 声明函数使用关键词 var。 ()
8. HTML DOM 定义了访问和操作 HTML 文档的标准方法。 ()
9. JQuery 是一个 JavaScript 库。 ()
10. 通过 `$("#div.intro")` 能够选取的元素是 `class="intro"` 的首个 div 元素。 ()

三、填空题

1. 所有 HTML 标签用 _____ 属性指定元素的唯一标识,用 _____ 属性指定类 CSS 值。
2. 在 HTML 中用于元素换行的标签是 _____,分块标签是 _____。
3. 在 table 的所有子标签中,数据行标签是 _____,数据列标签是 _____。
4. 在 option 标签中用 _____ 属性表示选中,在单选和多选按钮控件中用 _____ 属性表示选中。
5. 在 form 标签中用 _____ 属性指定表单提交的资源 URL 地址,用 _____ 属性指定表单数据传输方式。
6. CSS 代码由 _____、属性和 _____ 构成。
7. CSS 的类别选择器以符号 _____ 开头,id 选择器以符号 _____ 开头。
8. JavaScript 有两种特殊数据类型,分别是 _____、_____。
9. 如果已知 HTML 页面中的某标签对象的 `id="username"`,在 Javascript 中用 _____ 获得该标签,在 JQuery 中用 _____ 获得该标签。
10. JQuery 选择器主要分为两大类,分别是 _____ 和 _____。

四、简答题

1. 简述用 GET、POST 传输数据的异同。
2. 简述在 HTML 代码中引入 CSS 的 3 种方式。
3. 简述 JavaScript 对象、函数、数组的定义。
4. JQuery 如何来设置和获取 HTML 和文本的值?