

JavaScript基础

- 第1章 初识 JavaScript
- 第2章 JavaScript 基础知识
- 第3章 控制语句
- 第4章 函数
- 第5章 对象
- 第6章 浏览器对象
- 第7章 文档对象
- 第8章 DOM事件
- 第9章 JavaScript 代码优化

第1章 初识 JavaScript

1.1 什么是 JavaScript

JavaScript(简称 JS)是一门用于前端开发的计算机语言,在不与服务器交互的情况下,可修改 HTML 内容,为网页添加各种动态效果。它是网页动画与交互的效果之源, 也是构建 Web 应用的核心部分。

随着 5G 移动网络的普及,在各种新技术、新标准的推动下,今天的 Web 前端技术已 形成了一个大的技术系统,涉及知识点如图 1-1 所示。



图 1-1 Web 前端开发技术

(1) Web 前端泛指在 Web 应用中,用户可以看得见的内容,包括 Web 页面的结构、Web 的外观表现以及 Web 的交互实现。HTML 语言属结构,决定网页的内容是什么; CSS 样式属表现,展示网页的效果是什么样子;而 JavaScript 技术属行为,它是一种基于对象和事件驱动的脚本语言,用于控制网页的交互行为。在 Web 前端开发技术中,JavaScript 技术是最重要的,它的语法和 Java 类似,属于解释型语言,边执行边解释。JavaScript 的最大特点是"一切皆对象",它是一种拥有属性、方法的特殊数据类型。

(2) Web 后端更多的是与数据库进行交互、处理相应的业务逻辑,需要考虑的是数据的存取、平台的稳定性与性能等。Web 后端编程语言是百花齐放,有 PHP、JSP、 node.js等编程语言,但 Web 前端的脚本是一枝独秀,处理前端效果,非 JavaScript 语言不可。不过,JavaScript 也可以应用在其他方面,比如 node.js 使 JS 变成了服务器端脚本,类似 PHP。

1.2 JavaScript 的发展史

1994年,网景公司(Netscape)发布 Navigator 浏览器,但是这款浏览器只能浏览页面,为了解决这个问题,网景决定发明一种全新的语言。

1995年,网景公司的布兰登·艾奇(Brendan Eich)为 Navigator 浏览器开发了一种 名为 LiveScript 的脚本语言。当时,Netscape 为了搭上媒体热炒 Java 的"顺风车",临时 将 LiveScript 改名为 JavaScript,所以从本质上讲,JavaScript 和 Java 是没有什么关系的。

因为网景公司开发的 JavaScript 1.0 获得了巨大的成功,一年后微软迫于竞争对手的压力模仿 JavaScript 开发了 JScript。为了让 JavaScript 标准化,微软、网景、ScriptEase等公司联合 ECMA(欧洲计算机制造商协会)组织定制了 JavaScript 语言的标准,称为 ECMAScript 标准。虽然 JavaScript 和 ECMAScript 通常被人们用于表达相同的意思,但 JavaScript 的含义比 ECMA-262 中规定的多得多。一个完整的 JavaScript 实现由 ECMAScript(语法标准,核心)、BOM 浏览器对象模型、DOM 文档对象模型 3 个部分组成。

由于 ECMA-262 定义的 ECMAScript 只是这门语言的基础,与 Web 浏览器没有依赖关系,Web 浏览器只是 ECMAScript 实现可能的宿主环境之一,因此,在 ECMAScript 基础上,Web 浏览器就可能构建出更完善的脚本语言,不仅提供了基本的 JavaScript 的实现,还提供了该语言的扩展,如 BOM、DOM。

JavaScript 的 ECMAScript 标准在不断发展,平时讲到 JavaScript 版本,实际上就是 说它是实现了 ECMAScript 标准的哪个版本。

1997 年 7 月, ECMAScript 1.0 发布, 实质上与 Netscape 的 JavaScript 1.0 相同, 只 是做了一些小改动, 支持 Unicode 标准, 对象与平台无关。

2009年12月,ECMAScript 5.0正式发布。它添加了新功能,包括原生 JSON 对象、 继承的方法和高级属性定义以及严格模式。

2011年6月, ECMAScript 5.1发布并且成为 ISO 国际标准。

2015 年 6 月, ECMAScript 6.0 正式发布, 并更名为 ECMAScript 2015。ECMAScript 6.0 的目标是使 JavaScript 语言可以用来编写大型的、复杂的应用程序,称为企业级开发语言。

2020年6月,发布 ECMA-262(第11版),即 ECMAScript 2020通用编程语言的标准。

1.3 JavaScript 的功能

通过 JavaScript 脚本,主要实现两项功能:对 DOM(HTML)的增删改查、对事件 (event)的响应和处理。例如,单击图 1-2 购物车中的"删除"图标,可以删除选购的商品。 光标滑过图 1-3 所示的选项卡,可以切换扫码登录或账户登录的界面。

下面通过案例来体验一下 JavaScript 的用途。

【例 1-1】 打开页面,输出"hello world!",如图 1-4 所示。

分析:

最简单的实现方式,就是把输出内容写在<body>标签内。

4

.

1 Ait	*0		m/A	10-5L	4.14	18.00-
M #22	Man		497	53.88	744	SRIF
✔ 京东自营	Í.	优惠券 ~	0	还差18.18元免运费 🗄	E湊单 > 开通PLUS	每月锁5张运费
満成 B	满2件,已减8.98元 去凑单 >				¥89.80 减: -¥8.98	
"	■生軽」結構进口紅心火龙果 4个装 红肉中果 单果約330-420g 新鮮水果	红心中果 4个	¥29.90 促钠 ~	- 1 + 有货	¥29.90	删除 移入关注
"	國主計 數項38号果态程子相接 種子 格子 2.5kg装 単果130g以上新起水	類還2.5kg	¥59.90 促的~	- 1 + 有質	¥59.90	删除 移入关注
♥ 全选 動除	赤中的商品 移入关注 書理時物车		Bi	5坪2件商品へ 总切	¥80.82 🤉	去结算

图 1-2 购物车的删除功能

扫码登录 账户登录	
打开手机京东 扫描二维码	③ 127.0.0.1:8848/example/1/1- x + ● $-$ □ × $\leftarrow \rightarrow$ C ③ 127.0.0.1:8848/example/1/1 $\mathbb{H} \Leftrightarrow \bigcirc$:
② 免編入 多 更快 ③ 更安全 ○ QQ ● 微信 ● 立即注册	hello world!
图 1-3 登录切换功能	图 1-4 页面输出一条信息

参考代码:

【例 1-2】 在例 1-1 静态页面的基础上,输出 1 万条"hello world!"的信息,如图 1-5 所示。



图 1-5 输出多条记录

分析:

如果直接用制作静态网页的方式去实现,必然存在一定困难,需要在< body >标签内 输入1万条"hello world!"。显然,这种方式是极其低效的。但是用 JavaScript 循环语句 实现就简单多了,只需保存几行代码,而且文件相对比较小。需要注意,在 HTML 内嵌 入 JavaScript 语句,需要用< script >标签声明一下。内容用 document. write()语句输出。

参考代码:

1.4 JavaScript 运行环境

浏览器内置 JavaScript 代码解释器,所以安装浏览器后就可以运行 JavaScript 代码。 不同的浏览器,解释器可能存在差异。本书的案例是使用 Google Chrome 浏览器进行测试与调试的。

JavaScript的执行流程如图 1-6 所示。用户通过浏览器的地址栏输入页面的地址,向服务器发出请求,请求查找包含 JavaScript 代码的页面。服务器接到请求后,返回保存在服务器上包含 JavaScript 的源文件。需要注意的是,服务端不进行 JavaScript 代码的处理,其只完成查找、传递 JavaScript 源文件的任务。当请求的网页返回浏览器端后,浏览器进行解析、渲染 HTML+CSS+JavaScript 代码。JavaScript 代码运行在浏览器端,故属于客户端的脚本。



HTML+CSS+JavaScript

图 1-6 JavaScript 执行流程

1.5 JavaScript 代码位置

在网页中引用 JavaScript 有 3 种方式。

(1) 内嵌。将内容直接写在 HTML 标签中,该方式不常用。例如:

<input type = "button" value = "点击试试" onclick = " javascript: alert('试试就试试'); "/>
(2) 内置。在< head >和< body >标签的任何位置嵌入"< script > JS 脚本</ script >"

代码。

(3)外联。导入外部 JavaScript 文件。在 HTML 文档中,用< script src="×××.js"></script >标签引用。其中×××.js 是单独保存的 JavaScript 文件。使用外部脚本的优 点是可实现页面内容和功能的分离,代码结构清晰,易于维护。

【例 1-3】 在网页中用 3 种方式引用 JavaScript。

参考代码:

```
<! DOCTYPE html >
< html >
  < head >
   < meta charset = "utf - 8">< title > </title >
   <!-- JavaScript 外联方式 -->
    < script type = "text/ javascript" src = "jsFile.js"></script>
  </head>
  < body >
   <!-- JavaScript 内嵌方式 -->
    < input type = "button" value = "点击试试" onclick = "javascript:alert('试试就试试');"/>
   <!-- JavaScript 内置方式 -->
    < script type = "text/ javascript">
        alert("欢迎进入 JS 学习!");
    </script>
  </body>
</html>
```

jsFile.js文件内容如下。

//外部 JavaScript 文件 alert("成功导入 JavaScript 文件");

注意:HTML 文档的解析顺序是自上向下,此时因为 JavaScript 代码放置的位置不

同,会产生不同的运行结果。

(1) 放置在<head>标签内。由于 HTML 文档是由浏览器从上往下依次载入的,故 在执行 JavaScript 代码时,由于 HTML 元素还未载入,此时是无法操作 HTML 元素的。

(2)放置在< body >的结束标签前,可改善页面显示速度,因为脚本编译会拖慢显示。

一般情况下,建议把 JavaScript 代码放在</body>之前。如果要放置在< head >标签内,最佳解决方案是用 window 对象的 onload 事件控制 JavaScript 代码的执行时机。当 网页代码加载完成时,才触发 JavaScript 运行代码。

1.6 JavaScript 代码调试

代码调试是将编写好的程序进行测试,对测试结果中出现的错误进行分析,找出原因和具体的位置,并进行纠正,这是保证代码正确性必不可少的步骤。调试 PC 端 JavaScript 代码的方式有很多种,可以通过以下方式进行。

1. 通过 Console(控制台)查看代码运行情况

专业开发人员编写代码时,需要不停调试与测试。JavaScript 代码运行后,可以切换 到 Console(控制台)进行调试。打开 Google Chrome 控制台的快捷键是 F12。

假设在 file. js 源文件的末尾添加一句代码:

alert(a); //a 为变量,但没赋值

运行例 1-3. html 代码,在 Console 控制台就会显示对应错误的提示,如图 1-7 所示。 需要注意的是,不是所有错误都能在控制台中正确显示出来。



图 1-7 控制台错误提示

2. 通过 console. log()在 Console 控制台打印信息

假设在调试过程中需要查看中间过程的信息,可通过在 JavaScript 中添加 console. log (msg)语句,把输出结果显示在控制台。msg 为需要在控制台打印的信息(如变量值)。运行程序后,打开浏览器并按 F12 键,即可在控制台查看到打印的信息。

【例 1-4】 获取 id="zcool"的元素,并通过 console. log 打印该元素、元素的 href 属性以及元素的文本内容,如图 1-8 所示。

参考代码:

<! DOCTYPE html >

DevTool:	5 - 127.0.0.1:8848,	/example/1/1-4	4.html					-		×
	Elements	Console	Sour	ces	Network	»			🌣	:
D O	top	*	0	Fil	ter		Defa	ult leve	ls ▼	\$
<a l<="" td=""><td>href="<u>https:</u></td><td>//www.zcool</td><td></td><td><u>:n/</u>"</td><th>id="zcool"</th><td>'>站酷</td><td></td>	href=" <u>https:</u>	//www.zcool		<u>:n/</u> "	id="zcool"	'>站酷		<u>1-4.</u>	html:1	0
https:	//www.zcool	.com.cn/						1-4.	html:1	1
站酷								1-4.	html:1	2
>										

图 1-8 在控制台打印信息

3. 通过 alert 弹出窗口查看信息

alert 与 console. log 一样, alert 通过在 JavaScript 中添加 alert(msg)进行调试, msg 为需要在弹出窗口中显示的信息。

例如,例 1-4 的 JavaScript 代码可修改如下。

```
< script type = "text/javascript">
    var obj = document.getElementById('zcool');
    alert(obj.href);
    alert(obj.innerHTML);
</script>
```

需要注意的是,弹窗是强制阻塞,只有关闭窗口,才能解除阻塞,因此需要谨慎使用。

4. 通过 Sources 选项卡设置断点进行程序调试

Sources 选项卡的左边是内容源,包括页面中的各种资源。中间区域展示左边资源文件的内容。右边是调试功能区,最上面的一排按钮分别是暂停/继续、单步执行、单步跳入、单步跳出、禁用/启用所有断点。下面是各种具体的功能区,如图 1-9 所示。注意,左右两边的区域默认收缩在两侧没有显示出来,单击两侧的"伸缩"按钮 【 可展示出来。

通过左边的内容源,打开对应的 JavaScript 文件。单击文件的行号,就可以设置和删除断点。添加的每个断点都会出现在右侧调试区的 Breakpoints 列表中,单击列表中的断点就会定位到内容区的断点上。对于每个已添加的断点都有两种状态:激活和禁用。 刚添加的断点都是激活状态,禁用状态就是保留断点但临时取消该断点功能。

9

DevTools - 127.0.0.1:8848/ex	ample/1/1-	0.html								-	0	×
🕞 🚹 📔 Elements	Console	Sources	Network	Performance	Mem	ory	App	licatio	on	»	1	: :
Page »	1	-0.html ×				₽,	â	+	1	*	7/2	0
r 🗖 top	6					▼ Wa	atch				+	C
▼ △ 127.0.0.1:8848	8	<body> <script :<br="" type="text/javascript">sum=0;</td><td colspan=6>sum: 0</td></tr><tr><td>▼ 📄 example/1</td><td>9</td><td colspan=5>▼ Call Stack</td></tr><tr><td>1-0.html</td><td>10</td><td colspan=4>for(1=0;1<10000;1++){ sum=sum+i;</td><td colspan=3> (anonymous) 1-0.htm </td><td>.html:1</td><td>1</td></tr><tr><td></td><td>12</td><td rowspan=3 colspan=4><pre>} document.write(sum); </script> <script></script></body>										

图 1-9 Google Chrome 的 Sources 选项卡

也可以设置条件断点。在断点位置的右键菜单中选择 Edit Breakpoint 命令设置触发断点的条件,就是写一个表达式,表达式的值为 true 时才触发断点。

5. 通过 Network 选项卡查看页面加载过程

查看 HTTP 请求后可得到的各个请求资源的详细信息,如状态、资源类型、大小、所 用时间等,可以根据这个进行网络性能优化,如图 1-10 所示。

DevTools - 127.0.0.1:8848	/example/10/10-2	html				- 0	×
🕞 🚹 🛛 Elements	Console	Sources Netwo	ork Performan	ce Memory	/ »	1.0	ŧ. (۲
• • • •	Preserve log	Disable cach	e Online	<u>±</u>			•
Filter	Media Font	e data URLs Doc WS Manif	fest Other 🗌 I	Has blocked co	okies 🗌 B	locked Reque	ests
20 ms	40 ms	60 ms	80 ms	100 ms	1	20 ms	14(
Name	Status	Туре	Initiator	Size	Time	Waterfall	
10-2.html	304	document	Other	284 B	10 ms		1 -
jquery-3.5.1.js	304	script	<u>10-2.html</u>	286 B	9 ms	4	
5 requests 41.0 kB tra	ansferred 322	kB resources	Finish: 114 ms [OOMContentLo	baded: 97 m	is Load: 97	ms

图 1-10 Google Chrome 的 Network 选项卡

但在开发移动端应用时,不像开发 PC 端那样可以在控制台中查看各种网络请求,以 及打印的日志,但可以借助微信团队开发的移动端调试器 vconsole。