

JavaScript

維基百科，自由的百科全書

JavaScript，通常縮寫為**JS**，是一種進階的，解釋執行的程式語言^[5]。JavaScript是一門基於原型、函式先行的語言^[6]，是一門多範式的語言，它支援物件導向編程，指令式程式設計，以及函數語言程式設計。它提供語法來操控文字、陣列、日期以及正規表示式等，不支援I/O，比如網路、儲存和圖形等，但這些都可以由它的宿主環境提供支援。它已經由ECMA（歐洲電腦製造商協會）通過ECMAScript實現語言的標準化^[5]。它被世界上的絕大多數網站所使用，也被世界主流瀏覽器（Chrome、IE、Firefox、Safari、Opera）支援。

雖然JavaScript與Java這門語言不管是在名字上，或是在語法上都有很多相似性，但這兩門程式語言從設計之初就有很大的不同，JavaScript的語言設計主要受到了Self（一種原型程式設計語言）和Scheme（一門函數語言程式設計語言）的影響^[6]。在語法結構上它又與C語言有很多相似（例如if條件語句、while迴圈、switch語句、do-while迴圈等）^[7]。

在用戶端，JavaScript在傳統意義上被實現為一種解釋語言，但在最近，它已經可以被即時編譯（JIT）執行。隨著最新的HTML5和CSS3語言標準的推行它還可用於遊戲、桌面和行動應用程式的開發和在伺服器端網路環境執行，如Node.js。

目錄

- 歷史
 - 肇始於網景
 - 微軟採用
 - 標準化
- 概論
- 特性
 - 指令式與結構化
 - 動態化
 - Eval()
- 編程
- 範例
- 版本
- 參見
- 參考文獻
- 外部連結

歷史

JavaScript	
	
編程範型	程式語言，物件導向（基於原型），指令式，函數式
設計者	布蘭登·艾克
實作者	網景通訊公司，Mozilla基金會
面市時間	1995年12月4日 ^[1]
穩定版本	ECMAScript 2016^[2] （2016年6月17日）
型態系統	動態型別、duck
網站	developer.mozilla.org/en-US/docs/Web/JavaScript (https://developer.mozilla.org/en-US/docs/Web/JavaScript)
	
主要實作產品	
V8、JavaScriptCore、SpiderMonkey、Chakra	
啟發語言	
C、Java、Perl、Python、Scheme、Self	
影響語言	
ActionScript、CoffeeScript、Dart、JScript、.NET、Objective-J、QML、TIScript、TypeScript	
 維基教科書中有關JavaScript的文字	

JavaScript	
<div><pre>String.prototype.trim = function () { return this .replace (/^s+/, "") .replace (/s+\$/, ""); }</pre></div>	
副檔名	.js
網路媒體型式	application/javascript <div> <div>text/javascript</div> <div>(obsolete)^[3]</div> </div>
統一類型標識	com.netscape.javascript-source ^[4]
格式類型	程式語言

肇始於網景

1993年，伊利諾大學厄巴納-香檳分校的國家超級電腦應用中心（NCSA）發表了NCSA Mosaic，這是最早流行的圖形介面網頁瀏覽器，它在全球資訊網的普及上發揮了重要作用。1994年，一家名為Mosaic Communications的公司在加州山景城成立了，並雇用了許多原來的NCSA Mosaic開發者用來開發Mosaic Netscape，該公司的目標是取代NCSA Mosaic成為世界第一的網頁瀏覽器。第一個版本的網頁瀏覽器Mosaic Netscape 0.9於1994年底發布。在四個月內，已經佔據了四分之三的瀏覽器市場，並成為1990年代網際網路的主要瀏覽器。為避免NCSA的商標所有權問題，該瀏覽器於同年更名為Netscape Navigator，該公司命名為Netscape Communications。網景預見到網路需要變得更動態。公司的創始人馬克·安德森認為HTML需要一種膠水語言，讓網頁設計師和兼職程式設計師可以很容易地使用它來組裝圖片和外掛程式之類的元件，且程式碼可以直接編寫在網頁標記中。

1995年，網景招募了布蘭登·艾克，目標是把Scheme語言嵌入到Netscape Navigator瀏覽器當中^[8]。但更早之前，網景已經跟昇陽合作在Netscape Navigator中支援Java，這時網景內部產生激烈的爭論^[9]。後來網景決定發明一種與Java搭配使用的輔助指令碼語言並且語法上有些類似^[10]，這個決策導致排除了採用現有的語言，例如Perl、Python、Tcl或Scheme。為了在其他競爭提案中捍衛JavaScript這個想法，公司需要有一個可以運作的原型。艾克在1995年5月僅花了十天時間就把原型設計出來了。

最初命名為Mocha，1995年9月在Netscape Navigator 2.0的Beta版中改名為LiveScript，同年12月，Netscape Navigator 2.0 Beta 3中部署時被重新命名為JavaScript^{[1][11]}，當時網景公司與昇陽電腦公司組成的開發聯盟為了讓這門語言搭上Java這個程式語言「熱詞」，因此將其臨時改名為JavaScript，日後這成為大眾對這門語言有諸多誤解的原因之一^[12]。

微軟採用

JavaScript推出後在瀏覽器上大獲成功，微軟公司在不久後就為Internet Explorer 3瀏覽器推出了JScript，以與處於市場領導地位的網景產品同台競爭。JScript也是一種JavaScript實現，這兩個JavaScript語言版本在瀏覽器端共存意味著語言標準化的缺失，發展初期，JavaScript的標準並未確定，同期有網景的JavaScript，微軟的JScript雙峰並峙。除此之外，微軟也在網頁技術上加入了不少專屬物件，使不少網頁使用非微軟平台及瀏覽器無法正常顯示^{[13][14]}，導致在瀏覽器大戰期間網頁設計者通常會把「用Netscape可達到最佳效果」或「用IE可達到最佳效果」的標誌放在首頁^[15]。隨著Internet Explorer 4的發布，微軟引入了動態HTML的概念，但語言實現和不同專有化的文件物件模型的差異仍然存在，成為網路上普及使用JavaScript的阻礙^[13]。

標準化

1996年11月，網景正式向ECMA（歐洲電腦製造商協會）提交語言標準。1997年6月，ECMA以JavaScript語言為基礎制定了ECMAScript標準規範ECMA-262。JavaScript成為了ECMAScript最著名的實現之一^[16]。除此之外，ActionScript和JScript也都是ECMAScript規範的實現語言。儘管JavaScript作為給非程式人員的指令碼語言，而非作為給程式人員的程式語言來推廣和宣傳，但是JavaScript具有非常豐富的特性。

概論

一般來說，完整的JavaScript包括以下幾個部分：

- ECMAScript，描述了該語言的語法和基本物件
- 文件物件模型（DOM），描述處理網頁內容的方法和埠
- 瀏覽器物件模型（BOM），描述與瀏覽器進行互動的方法和埠

JavaScript的基本特點如下：

- 是一種解釋性程式語言（程式碼不進行預編譯）。
- 主要用來向HTML頁面添加互動行為。
- 可以直接嵌入HTML頁面，但寫成單獨的js檔案有利於結構和行為的分離。

JavaScript常用來完成以下任務：

- 嵌入動態文字於HTML頁面
- 對瀏覽器事件作出回應
- 讀寫HTML元素
- 在資料被提交到伺服器之前驗證資料
- 檢測訪客的瀏覽器資訊
- 控制cookies，包括建立和修改等

特性

不同於伺服器端指令碼語言，例如PHP與ASP，JavaScript主要被作為用戶端指令碼語言在用戶的瀏覽器上運行，不需要伺服器的支援。所以在早期程式設計師比較青睞於JavaScript以減少對伺服器的負擔，而與此同時也帶來另一個問題：安全性。而隨著伺服器的強壯，雖然現在的程式員更喜歡運行於伺服端的指令碼以保證安全，但JavaScript仍然以其跨平台、容易上手等優勢大行其道。同時，有些特殊功能（如AJAX）必須依賴JavaScript在用戶端進行支援。隨著引擎如V8和框架如Node.js的發展，及其事件驅動及異步IO等特性，JavaScript逐漸被用來編寫伺服器端程式。且在近幾年中，Node.js的出世，讓JavaScript也具有了一定的伺服器功能，且在某些方面比PHP的效果更為顯著。

指令式與結構化

JavaScript從支援許多C語言的結構化編程語法（例如if條件語句、while迴圈、switch語句、do-while迴圈等）。但作用域是一個例外：JavaScript只支援使用var關鍵字來定義變數的函式作用域。ECMAScript加入了let關鍵字來支援塊級作用域，意味著JavaScript現在既支援函式作用域又支援塊級作用域。正如C語言一樣，JavaScript中的運算式和語句是不同的。與C不同，JavaScript支援自動在語句末添加分號。

動態化

正如大部分程式語言，類型與值而不是與變數關聯。例如x變數可以為數值，隨後又可被賦值為字串。JavaScript提供了包括鴨子型別在內的方法來檢測變數類型。

Eval()

Eval() 函式可以直接執行一個JavaScript函式。例如：

```
eval("alert(\"Message\")");
```

編程

JavaScript是一種指令碼語言，其原始碼在發往用戶端執行之前不需經過編譯，而是將文字格式的字元程式碼發送給瀏覽器由瀏覽器解釋執行。直譯語言的弱點是安全性較差，而且在JavaScript中，如果一條執行不了，那麼下面的語言也無法執行。而其解決辦法就是於使用例外處理try{}catch(){}：

```

console.log("a");    // 這是正確的
console.log("b");    // 這是正確的
console.log("c");    // 這是錯誤的，並且到這裡會停下來
console.log("d");    // 這是正確的
console.log("e");    // 這是正確的

/* 解決辦法 */
try{console.log("a");}catch(e){}    // 這是正確的
try{console.log("b");}catch(e){}    // 這是正確的
try{console.log("c");}catch(e){}    // 這是錯誤的，但是到這裡不會停下來，而是跳過
try{console.log("d");}catch(e){}    // 這是正確的
try{console.log("e");}catch(e){}    // 這是正確的

```

JavaScript被歸類為直譯語言，因為目前主流的引擎都是每次執行時載入程式碼並解譯。V8是將所有程式碼解譯後再開始執行，其他引擎則是逐行解譯（SpiderMonkey會將解譯過的指令暫存，以提高效能，稱為即時編譯），但由於V8的核心部份多數用JavaScript撰寫（而SpiderMonkey是用C++），因此在不同的測試上，兩者效能互有優劣。

與其相對應的是編譯語言，例如C語言，以編譯語言編寫的程式在執行之前，必須經過編譯，將程式碼編譯為機器碼，再加以執行。

範例

以下是一個簡單的JavaScript Hello World：

```

<!DOCTYPE HTML>
<html>
  <head>
    <title>簡單的JavaScript Hello World</title>
    <script type="text/javascript">
      document.write("Hello, world!");    // 於瀏覽器視窗內直接顯示
      alert("Hello, world!");             // 開啟對話視窗顯示
      console.log("Hello, world!");        // 於console裡顯示，需要先開啟開發工具控制台
    </script>
  </head>
  <body>
    HTML內容.....
  </body>
</html>

```

或是在瀏覽器的網址列中使用javascript:，以互動方式表示：

```

javascript:alert("Hello world!");

```

版本

JavaScript最初開發於1996年，被使用於Netscape Navigator網頁瀏覽器。同年微軟在Internet Explorer發布了一個實作。由於商標問題，這項實作被命名為JScript。1997年，JavaScript已經被網景公司提交給ECMA制定為標準，稱之為ECMAScript，標準編號ECMA-262。目前最新版為ECMA-262 5th Edition^[17]。目前ECMA-262 4th Edition正在開發的過程中，與其對應的JavaScript的版本為JavaScript 2.0，目前也正在開發過程中。

下列表格的資訊基於多個參考來源^{[18][19][20][21]}：

版本	發布日期	基於	Netscape Navigator	Mozilla Firefox	Internet Explorer	Opera	Safari	Google Chrome
1.0	1996年3月		2.0		3.0			
1.1	1996年8月		3.0					
1.2	1997年6月		4.0 - 4.05			3		
1.3	1998年10月	ECMA-262 1st + 2nd edition	4.06 - 4.7x		4.0	5 ^[22]		
1.4			Netscape Server			6		
1.5	2000年11月	ECMA-262 3rd edition	6.0	1.0	5.5 (JScript 5.5) 6 (JScript 5.6) 7 (JScript 5.7) 8 (JScript 5.8)	7.0	3.0-5	1.0 - 10.0.666
1.6	2005年11月	1.5 + Array extras + Array and string generics + <u>E4X</u>		1.5				
1.7	2006年10月	1.6 + <u>Pythonic generators</u> (https://developer.mozilla.org/en-US/docs/JavaScript/New_in_JavaScript/1.7?redirectlocale=en-US&redirectslug=New_in_JavaScript_1.7#Generators) + Iterators + Let		2.0				28.0.1500.95
1.8	2008年6月	1.7 + Generator expressions + <u>Expression closures</u>		3.0		11.50		
1.8.1		1.8 + <u>native JSON</u> support + Minor updates		3.5				
1.8.2	2009年6月22日	1.8.1 + Minor updates		3.6				
1.8.5	2010年7月27日	1.8.2 + New features for ECMA-262 5th edition compliance		4.0				

參見

- 指令碼語言
- DMDScript

- [ECMAScript](#)
- [JavaScript引擎](#)
- [JScript](#)
- [AJAX](#)
- [JavaScript函式庫](#)

參考文獻

1. Press release announcing JavaScript (<https://web.archive.org/web/20070916144913/http://wp.netscape.com/newsref/pr/newsrelease67.html>), "Netscape and Sun announce JavaScript", PR Newswire, December 4, 1995
2. [Standard ECMA-262](#). Ecma International. 2015-06-17.
3. [RFC 4329](#) (<http://www.apps.ietf.org/rfc/rfc4329.html#sec-7.1>) 網際網路檔案館的存檔 (<https://web.archive.org/web/20140316122853/http://www.apps.ietf.org/rfc/rfc4329.html>)，存檔日期2014-03-16.
4. [System-Declared Uniform Type Identifiers](#). Mac OS X Reference Library. Apple Inc. [2010-03-05].
5. David, Flanagan. JavaScript: The Definitive Guide 6th. O'Reilly & Associates. ISBN 978-0-596-80552-4.
6. [ECMAScript Language Overview](#) (PDF): 4. 2007-10-23 [2009-05-03]. （[原始內容](#) (PDF)存檔於2010-07-13）.
7. [JavaScript: The World's Most Misunderstood Programming Language](#). www.crockford.com. [2016-08-17].
8. Chapter 4. How JavaScript Was Created.
9. Severance, Charles. JavaScript: Designing a Language in 10 Days. Computer (IEEE Computer Society). February 2012, **45** (2): 7–8 [23 March 2013]. doi:10.1109/MC.2012.57.
10. [TechVision: Innovators of the Net: Brendan Eich and JavaScript](#). 2009-03-19. （[原始內容](#) (HTML)存檔於2008-02-08）.
11. [TechVision: Innovators of the Net: Brendan Eich and JavaScript](#). web.archive.org. （[原始內容](#)存檔於2008-02-08）.
12. Fin JS, [Brendan Eich - CEO of Brave](#), 2016-06-17 [2018-02-07]
13. Champeon, Steve. [JavaScript, How Did We Get Here?](#). oreilly.com. 6 April 2001 [16 July 2016]. （[原始內容](#)存檔於2016-07-19）.
14. [Microsoft Internet Explorer 3.0 Beta Now Available](#). microsoft.com. Microsoft. 29 May 1996 [16 July 2016].
15. McCracken, Harry. [The Unwelcome Return of "Best Viewed with Internet Explorer"](#). technologizer.com. 16 September 2010 [16 July 2016].
16. Nicholas, Zakas. Professional JavaScript for Web Developers 3rd. Wrox. 2012. ISBN 978-1-118-02669-4.
17. https://developer.mozilla.org/en/JavaScript/Language_Resources
18. [New in JavaScript](#). developer.mozilla.org. 2014 [2016-07-16].
19. JavaScript – JScript – ECMAScript version history. Webmasterworld.com. [2009-12-17].
20. John Resig. [Versions of JavaScript](#). Ejohn.org. [2009-05-19].
21. [Version Information \(JScript\)](#). Msdn.microsoft.com. [2009-12-17].
22. [What Version of JavaScript](#).

外部連結

- [關於JavaScript介紹](https://developer.mozilla.org/docs/Web/JavaScript) (<https://developer.mozilla.org/docs/Web/JavaScript>) [Mozilla Developer Network](#)上的資源

取自 "<https://zh.wikipedia.org/w/index.php?title=JavaScript&oldid=51760443>"

本頁面最後修訂於**2018年10月24日 (週三) 08:47**。

本站的全部文字在創作共用 姓名標示-相同方式分享 3.0 協議之條款下提供，附加條款亦可能應用（請參閱[使用條款](#)）。
Wikipedia®和維基百科標誌是維基媒體基金會的註冊商標；維基™是維基媒體基金會的商標。
維基媒體基金會是按美國國內稅收法501(c)(3)登記的非營利慈善機構。