JavaScript

aktuelle Generation

- Die aktuelle Version ist die **Version 2017**, im Juni 2017 als „ECMAScript 2017“ veröffentlicht
- der als **ECMAScript** (ECMA 262) standardisierte Sprachkern von JavaScript beschreibt eine **dynamisch typisierte, objektorientierte, aber klassenlose Skriptsprache**
- **wird allen objektorientierten Programmierparadigmen gerecht**, ähnlich wie bei klassenbasierten Programmiersprachen üblich
- es sollen jährliche Updates folgen

Versionsgeschichte von ECMAScript (ECMA-262)

Die aktuelle Version ist die Version 2017, die im Juni 2017 als „ECMAScript 2017“ veröffentlicht wurde.[21] Dieser sollen jährliche Updates folgen.[22] Die Entwicklung der Standards erfolgt auf GitHub.[23]

Version	publiziert am	Unterschiede zur Vorgängerversion	Editor
1	Juni 1997	erste Version	Guy L. Steele, Jr.
2	Juni 1998	Änderungen zwecks Kompatibilität zum internationalen Standard ISO/IEC 16262	Mike Cowlishaw
3	Dezember 1999	Neu sind reguläre Ausdrücke, bessere Verarbeitung von Zeichenketten, Kontrollfluss, Fehlerbehandlung mit try/catch, bessere Fehlerbehandlung, bessere Formatierung bei der Ausgabe von Zahlen usw.	Mike Cowlishaw
4	abgebrochen	Wegen Uneinigkeit in Bezug auf die Zukunft der Sprache wurde die weitere Entwicklung des komplexen Entwurfes zu ECMAScript 4 eingestellt. Einige Ideen werden in ES6 wieder aufleben.	
5	Dezember 2009	Im „strict mode“ wird eine erweiterte Fehlerprüfung eingeschaltet. Unklare Sprachkonstrukte von ECMAScript 3 werden entschärft und neue Features wie getter- und setter-Methoden, Unterstützung von JSON usw. hinzugefügt.[24]	Pratap Lakshman, Allen Wirfs-Brock
5.1	Juni 2011	Entspricht dem internationalen Standard ISO/IEC 16262:2011, Version 3	Pratap Lakshman, Allen Wirfs-Brock
2015[25][26]	Juni 2015	Neue Syntax für komplexe Applikationen wie Klassen und Module, die aber mit ähnlicher Terminologie wie in ECMAScript 5 (strict mode) definiert werden können.[27] Neue Sprachbestandteile wie `for` / `of` -Schleifen, teilweise an Python angelehnte Syntax usw. Der Codename lautet “Harmony” und wurde bis kurz vor Verabschiedung als „ECMAscript 6“ bezeichnet.[28]	Allen Wirfs-Brock
2016[21][29]	Juni 2016	`**` (Potenzfunktion), `Array.prototype.includes`, diverse Anpassungen an Generatoren, destruktiven Zuweisungen[30]	Brian Terlson
2017[31]	Juni 2017	`async` / `await`, diverse `Object` -Funktionen[30][32][33]	
2018	Juni 2018[34]	`global`, `import()`, Rest/Spread Properties, `for - await - of` (Asynchronous Iterators), String-Padding, …[35]	

Sprachtyp

- **vollwertige Programmiersprache** im Gegensatz zur Auszeichnungssprache HTML, somit flexible einsetzbar
- **über einen Interpreter ausgeführt**, verzichtet auf Sprachelemente, deren Nutzen erst bei der Bearbeitung komplexerer Aufgaben zum Tragen kommt
- kann im Browser (clientseitig) und auf den Webserver (serverseitig) eingesetzt werden
- DHTML, **dynamisches HTML** oder DOM-Scripting mit Webdesign-Methoden, bei denen während der Anzeige einer Webseite diese selbst verändert wird, ausgelöst durch Benutzereingaben

- programmieren **je nach Bedarf objektorientiert, prozedural oder funktional**, objektbasierend heißt es dazu im Lehrbuch S. 16
- **multiparadigmatisch**, Programmierparadigmen unterscheiden sich durch ihre Konzepte für die Repräsentation von statischen (wie Objekte, Methoden, Variablen, Konstanten) und dynamischen (wie Zuweisungen, Kontrollfluss, Datenfluss) Programmelementen. Es können „viele Programmiersprachen **mehrere Paradigmen gleichzeitig** unterstützen“[3] wie in JavaScript

Datentypen

JavaScript	
Paradigmen:	multiparadigmatisch
Erscheinungsjahr:	1995
Entwickler:	Brendan Eich
Aktuelle Version	ECMAScript 2017[1]
Typisierung:	schwach, dynamisch, duck
Wichtige Implementierungen:	SpiderMonkey, Rhino, SquirrelFish, V8
Beeinflusst von:	Self, C, Scheme, Perl, Python, Java, Lua
Beeinflusste:	ActionScript, Haxe, CoffeeScript, Dart, TypeScript

- **dynamisch typisiert** und **schwach typisiert**, das heißt die Zuweisung von Werten an Variablen unterliegt keinen typbasierten Einschränkungen, Typprüfungen (etwa des Datentyps von Variablen) finden vorrangig zur Laufzeit eines Programms statt, (bei der statischen Typisierung wird die Typprüfung bereits zum Zeitpunkt der Kompilierung durchgeführtbei diversen Erweiterungen von JavaScript, wie TypeScript von Microsoft, wird die eine statische Typisierung optional beziehungsweise zwingend erfordert
- der Datentyp ist keine Eigenschaft einer Variablen, sondern Laufzeit-bezogen die Eigenschaft ihres aktuellen Wertes (oder auch die Eigenschaft eines Literals), der Datentyp eines Wertes lässt sich mit dem unären Operator typeof ermitteln
- eine **unäre oder monadische Verknüpfung** ist eine Verknüpfung mit nur einem Operanden, einstellige Verknüpfungen werden üblicherweise als Funktionen auf einer gegebenen Menge angesehen
- **Duck-Typing,** Konzept der objekt-orientierten Programmierung, bei dem der Typ eines Objektes nicht durch seine Klasse beschrieben wird, sondern durch das Vorhandensein bestimmter Methoden oder Attribute, Was aussieht und sich bewegt wie eine Ente, ist eine Ente. Duck-Typing ist charakteristisch auch für objektorientierte Skriptsprachen wie Python oder PHP.

Entwicklung

- JavaScript (kurz JS) ist eine Skriptsprache, die ursprünglich 1995 **von Netscape** für **dynamisches HTML in Webbrowsern entwickelt**, um Benutzerinteraktionen auszuwerten, Inhalte zu verändern, nachzuladen oder zu generieren und so die Möglichkeiten von HTML und CSS zu erweitern
- ursprünglich LiveScript entstand 1996 aus einer **Kooperation von Netscape mit Sun Microsystems**, deren Java-Applets, erstellt mit der gleichfalls 1995 veröffentlichten Programmiersprache Java, wurden mithilfe von LiveScript in den Netscape Navigator integriert
- **LiveScript wurde in JavaScript umbenannt**, obwohl die beiden Sprachen nur wenige Gemeinsamkeiten aufweisen, um die Popularität von Java aus Marketinggründen zu nutzen

- nach dem Ende der Firma Netscape übernahm die Mozilla Foundation (Firefox) die Weiterentwicklung der Skriptsprache

Versionsgeschichte von JavaScript

Versionsgeschichte[20]

Version	Veröffentlichung	Entsprechung	Netscape Navigator	Mozilla Firefox	Internet Explorer	Opera	Safari	Google Chrome
1.0.0	März 1996		2.0		3.0			
1.1.0	August 1996		3.0					
1.2.0	Juni 1997		4.0-4.05					
1.3.0	Oktober 1998	ECMA-262 1st edition / ECMA-262 2nd edition	4.06-4.7x		4.0			
1.4.0			Netscape Server					
1.5.0	November 2000	ECMA-262 3rd edition	6.0	1.0	• 5.5 (JScript 5.5) • 6 (JScript 5.6) • 7 (JScript 5.7) • 8 (JScript 6)	• 6.0 • 7.0 • 8.0 • 9.0		
1.6.0	November 2005	1.5 + Array extras + Array & String generics + E4X		1.5			• 3.0 • 3.1	
1.7.0	Oktober 2006	1.6 + Pythonic generators + Iterators + let + destructuring assignments		2.0			• 3.2 • 4.0	1.0
1.8.0	Juni 2008	1.7 + Generator expressions + Expression closures		3.0				
1.8.1	Juni 2009	1.8 + geringfügige Updates		3.5				
1.8.2	Januar 2010	1.8.1 + geringfügige Updates		3.6				
1.8.5	Juli 2010	1.8.1 + ECMAScript 5 Compliance		4	9.0 (JScript 9.0)			

Verwendung

- heute auch außerhalb von Browsern angewendet, so **auch auf Servern und in Microcontrollern**
- als Skriptsprache **für Spiele und Anwendungsprogramme** eingesetzt, da der Sprachkern nur **wenige Objekte** enthält und dadurch der zur Ausführung von in JavaScript formulierten Skripten **erforderliche Interpreter relativ klein** gehalten werden kann
- JavaScript wird als Verkehrssprache in der **Datenbank MongoDB** eingesetzt

Typische Anwendungsgebiete von JavaScript im Webbrowser

- **dynamische Manipulation von Webseiten** über das Document Object Model (**DOM**)
- Anzeige von Dialogfenstern
- **Senden und Empfangen von Daten**, ohne dass der Browser die Seite neu laden muss (**Ajax**)
- Vorschlagen von Suchbegriffen während der Eingabe
- Werbebanner oder Laufschriften
- Verschleierung von E-Mail-Adressen zur **Bekämpfung von Spam**
- mehrere Frames auf einmal wechseln oder die Seite aus dem Frameset lösen
- **Schreib- und Lesezugriff auf Cookies** und den Web Storage innerhalb des Browsers

Vorzüge

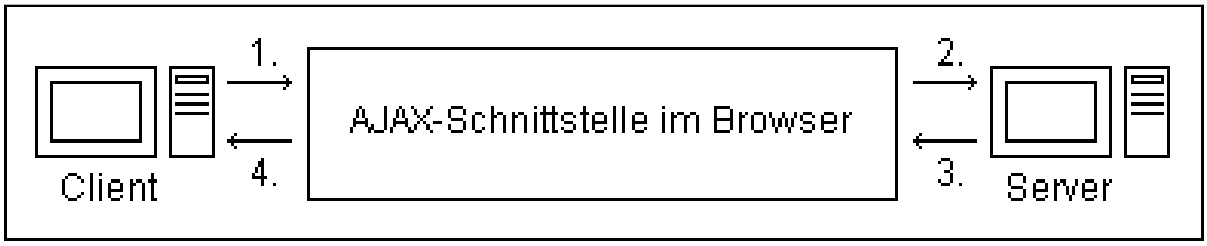
- auf Deklarationszwang von Variablen wird meist verzichtet, vorteilhaft zur schnellen **Erstellung kleiner Programmen**, wie Taschenrechner, Währungsumrechner, Kalender…
- Plausibilitätsprüfung (**Datenvalidierung**) von Formulareingaben z.B. die Postleitzahl besteht nur aus 5 Zahle bei deutschen Anschriften, das entlastet den Webserver
- **Darstellung von dynamischen Inhalten ohne Webserver** oder spezielle Browsererweiterungen, wie die Ausgabe von Messdaten in Diagrammen
- Skripte fast ausschließlich in Form von Quelltextdateien ausgeliefert, um so ein **einfaches Bearbeiten und Anpassen des Programms** zu ermöglichen
- Speicherung von Daten auf dem Client wie in Cookies
- Anpassung der Seiten und Inhalte an die Nutzergegebenheiten

Sandbox-Prinzip

- im Browser in einer sogenannten Sandbox ausgeführt, bewirkt, dass man in JavaScript im Allgemeinen nur Zugriff auf die Objekte des Browsers hat und somit nicht auf das Dateisystem zugreifen und dadurch keine Dateien lesen oder schreiben kann, eine Ausnahme stellt der Lesezugriff auf eine Datei dar, die mittels eines mit dem HTML-Element <input type="file"> gestarteten Dateiauswahl-Dialogs vom Benutzer ausgewählt wurde
- Schreib- und Lesezugriff auf Cookies und den Web Storage innerhalb des Browsers
- jede Website oder **Webanwendung innerhalb des Browsers isoliert ausgeführt,** Sicherheitsprobleme, wie das sogenannte Cross-Site-Scripting zu verhindern, (ohne diesen Schutz wäre es möglich, über eine Seite Schadcode auszuführen, der beispielsweise Bank- oder Logindaten in anderen parallel geöffneten Browserfenstern ausliest oder manipuliert)
- bestimmte sicherheitsrelevante Browserfunktionen, wie das Schließen des Browserfensters, das Aus- und Einblenden von Symbolleisten, das Ändern der Browserstartseite, der Zugriff auf die Zwischenablage oder das Auslesen der zuletzt besuchten Webseiten des Anwenders, werden durch obligatorische Nutzereingaben geschützt

AJAX - Das Prinzip

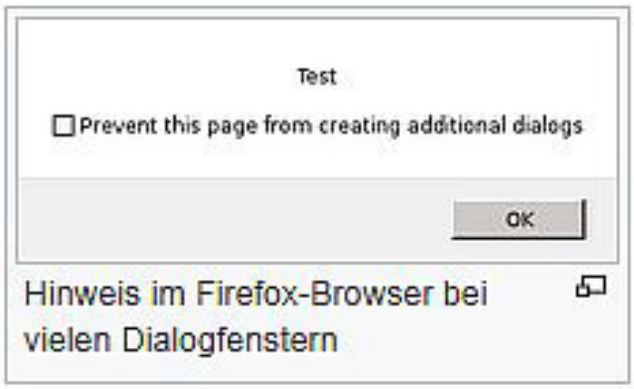
Ajax steht für Asynchronous JavaScript And XML. Damit sind serverseitige Anfragen im Hintergrund möglich, ohne dass die jeweilige Seite neu geladen werden muss, geschieht dabei im Hintergrund, so dass der Anwender davon nichts mitbekommt. Erfunden hat das Microsoft.



- (1.) Im Browser wird, wie auch immer, eine Aktion ausgelöst, die einen Ajax-Request zur Folge hat.
- (2.) Das Ajax-Objekt sendet im Hintergrund eine Anfrage an ein serverseitiges Script.
- (3.) Das Ajax-Objekt nimmt das Ergebnis der Anfrage (normalerweise) als XML-Dokument entgegen.
- (4.) Anhand des Ergebnisses wird mit normalem JavaScript die jeweilige Seite geändert.

Nachteile

Standardmäßig wird ein Skript innerhalb eines Browsers in Form eines einzigen Threads ausgeführt. Warteschleifen oder **lange Berechnungen sind daher in JavaScript-Programmen zu vermeiden**.



Hinweis im Firefox-Browser bei vielen Dialogfenstern

Missbrauch

Einige Anwendungen, die mit JavaScript möglich sind, agieren teilweise gegen den Wunsch des Benutzers oder **widersprechen dem Prinzip der geringsten Verwunderung**, dass besagt, dass eine Benutzerschnittstelle so ausgelegt werden sollte, dass der Benutzer möglichst wenige Überraschungen erlebt. (einige Browser bieten daher Funktionen an, die derartige JavaScript-Funktionen unterdrücken)

- Verschleiern von Internetadressen, auf die ein Link verweist
- Deaktivieren des Kontextmenüs, um zu erschweren, dass Bilder oder die gesamte Seite abgespeichert werden können
- Deaktivieren der Kopierfunktion, um zu erschweren, dass Texte oder Bilder kopiert werden können
- Unaufgeforderte (Werbe-)Pop-ups oder Pop-unders oder aufeinanderfolgende Dialogfenster, die den Benutzer behindern
- Ungewolltes Schließen des Browserfensters
- Ungewollte Größenänderung des Browserfensters
- barrierearme Webseiten zeichnen sich dadurch aus, dass sie auch bei abgeschaltetem JavaScript möglichst uneingeschränkt nutzbar bleiben. Teilweise schränkt das deaktivierte JavaScript die Benutzbarkeit einer Webseite ein.
- Maßnahmen, die an den Sicherheitseinstellungen des Browsers vorbei ein Wiedererkennen eines Benutzers bei einem späteren Besuch einer Website erlauben (siehe Anonymität im Internet)
- JavaScript kann auch von Dritten missbraucht werden, etwa per XSS (**Codeeinschleusung**)

Danke für Ihr Interesse.