



Betrieb 4.0 machen!

Produkte gestalten!

Nachgelesen

Moderne App-Entwicklung mit Javascript

M. Sc. Christoph Neumann

Mittelstand-
Digital



**Bundesministerium
für Wirtschaft
und Energie**

aufgrund eines Beschlusses
des Deutschen Bundestages

In dieser Ausgabe unserer Nachgelesen-Reihe erfahren Sie:

- Warum Javascript eine gute Wahl für Softwareprodukte ist,
- Was mit javascriptbasierten Apps möglich ist,
- Wie man bei der App-Entwicklung mit Javascript vorgeht,
- Wie zukunftsfähig Javascript ist.

Neue Herausforderungen

Die Gestaltung der eigenen Produkte stellt, insbesondere bei Softwareprodukten, ein zentrales Alleinstellungsmerkmal für Unternehmen gegenüber dem Kunden dar. Besonders im Softwarebereich nimmt die Digitalisierung und die fortlaufende Weiterentwicklung einen großen Stellenwert ein.

Digitale Produkte und Dienstleistungen sind daher hohen Anforderungen bezüglich ihrer Interaktivität, Modernität und Fähigkeiten ausgesetzt. Um diesen Ansprüchen der Kunden an das Softwareprodukt oder die Dienstleistungen gerecht zu werden, muss auch die Entwicklung sich diesem Trend entsprechend anpassen.

Aus diesem Grund ist die Wahl der Programmiersprache und Programmierumgebung, sowie der verfügbaren Dokumentation und Ent-

wicklungswerkzeuge, das Fundament einer flexiblen und zukunftsfähigen Softwareentwicklung.

Der typische Entwicklungsweg für Software besteht in der Regel aus vier Schritten:

- Anforderungsanalyse,
- Design/Mockup,
- Programmierung,
- Evaluation.

In dieser Nachgelesen-Ausgabe werden wir uns gesondert auf den Bereich der Programmierung und der praxisrelevanten Problemstellungen beziehen.

Plant man sein Produktangebot durch den Einsatz einer App zu erweitern oder bestehende Softwarelösungen zu modernisieren, steht man zu Beginn vor der Entscheidung, welche Programmiersprache am geeignetsten für das Produkt ist.

Das große Angebot an Programmiersprachen ist oft für den Erstanwender oder alteingesessenen Entwickler schwer überschaubar. Die hohe Weiterentwicklungsgeschwindigkeit und Spezialisierung der einzelnen Sprachen erschwert die Auswahl.

Für uns im Kompetenzzentrum hat sich durch grundsätzlich verschiedene Softwareprojekte herausgestellt, dass besonders Javascript, in Verbindung mit HTML, viele Anforderungen an eine moderne und flexible Softwareentwicklung erfüllt. Für die kommerzielle Nutzung scheint Javascript geeignet.

Daher soll im Folgenden auf die praktischen Implikationen für die Verwendung von Javascript als Programmiersprache für eigene Softwareprodukte eingegangen werden.



Abbildung 1: Eine kleine Auswahl gängiger Programmiersprachen

Was Javascript ausmacht

Grundsätzlich ist Javascript eine web-basierte Programmiersprache. Ursprünglich ist sie für die Interaktivität von Webseiten gedacht gewesen, hat sich aber besonders in den letzten Jahren sehr stark gewandelt. Man kann Javascript verwenden als:

- Clientbasierte Sprache für Interfaces
- Serverbasierte Sprache für Serverapplikationen.

Als serverbasierte Sprache lassen sich mit Hilfe von Node.js komplexe Serveranwendungen schreiben.

Als clientbasierte Sprache sind dynamische Interfaces programmierbar, die viele weitere Schnittstellen bieten.

Beispielhaft ist in Abbildung 2 der Aufbau einer clientbasierten Anwendung dargestellt.

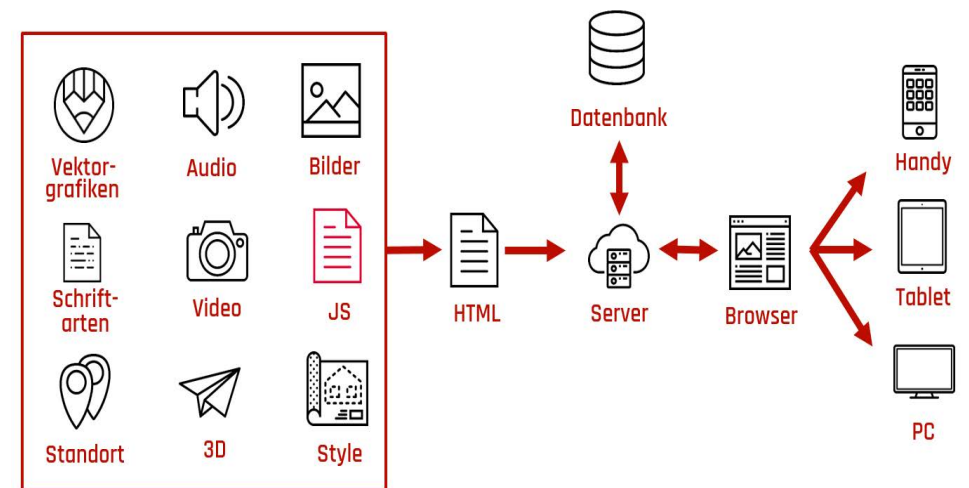


Abbildung 2: Was Javascript eigentlich ist und wo man es findet (vereinfacht)

Beginnend auf der rechten Seite der Darstellung befindet sich der Endnutzer. Dieser kann verschiedene Geräte nutzen, um mit einer in Javascript geschriebenen Anwendung zu interagieren. Es ist lediglich ein Browser notwendig, um die Applikation darzu-

stellen. Der Browser ruft von einem Server die Applikation auf und verarbeitet die erhaltenen Daten zu einer interaktiven Oberfläche. Der Server kann über das Internet erreichbar sein (webbasiert), sich im lokalen Netzwerk im Unternehmen befinden

oder auf dem Gerät selbst. Hier ist es bereits möglich, Schnittstellen zu beispielsweise Datenbanken oder anderen Datenanbietern einzubinden.

Der Server selbst gibt in erster Linie eine textbasierte HTML-Datei an den Browser des Nutzers aus. Innerhalb der HTML-Datei wird auf den Javascript-Code verwiesen.

Neben dem Javascriptcode kann auch auf andere Medieninhalte wie Bilder, Audio und Video verwiesen werden. Weitere Möglichkeiten sind Vektorgrafiken für hochauflösende Darstellungen, Schriftarten für bessere Lesbarkeit, Standortdaten für ortsbezogene Services, 3D-Daten für Tiefendarstellungen und Stildaten (CSS) für ein durchgängiges Design des Interfaces.

Diese Daten werden mit dem Javascriptcode an den Browser gesendet, der diese dann interpretiert und interaktiv nutzbar macht. Der Code wird auf dem Gerät des Nutzers verarbeitet. Daher die Bezeichnung clientbasiert (nutzerbasiert).

Wichtige Eigenschaften sind:

- Nahezu geräteunabhängig (es wird nur ein Browser benötigt)
- Code wird beim Endnutzer ausgeführt
- Verschiedene Medien sind nutzbar

- Modularisierung ist möglich
- Internet-, intranet- oder lokalbasiert

Ist bekannt, wie Javascript funktioniert, ergeben sich unter anderem folgende Vor- und Nachteile:

Vorteile

- Die Syntax ist leicht zu erlernen
- Kein Compiler¹ notwendig, ein Texteditor reicht
- Verwendung auf verschiedenen Plattformen (Web, lokal, Programmierschnittstellen, ...)
- Aufgrund der Beliebtheit gibt es eine große Community
- Ständige Codeverbesserungen und -erweiterungen (Normungsorganisation ECMA)
- Geringe Entwicklungszeit, da viele Standardbibliotheken bereits vorhanden sind

Nachteile

- Interpreter² notwendig (Browser, Javascript-Engine)
- Kein direkter System- oder Hardwarezugriff möglich
- Vereinzelt Caching-Probleme (Laden von Aktualisierungen)
- Abhängigkeit von Bibliotheken

- Asynchrones Verhalten manchmal komplex
- Ungeeignet für ressourcenlastige Anwendungen (RAM).

Es lässt sich schnell feststellen, ob Javascript für das eigene Softwareprodukt geeignet ist. Für das Entwickeln eigener Javascript-Applikationen ist lediglich ein gewöhnlicher Texteditor nötig. Damit kann auf leistungsschwachen Computern mit

Javascript entwickelt werden. Die meisten für die Entwicklung notwendigen Bibliotheken sind als Open Source frei verfügbar und werden kontinuierlich gepflegt. Ein Großteil steht unter freier Lizenz.

Nützliche Werkzeuge

Für die schnelle und flexible Entwicklung neuer Softwareprodukte stehen für Javascript zahlreiche Bibliotheken zur Verfügung. Für das Kompetenzzentrum sind die Bibliotheken, die den größten Einfluss auf die Entwicklung nehmen, folgende:

■ React

Mit dem Framework React kann man das Interface dynamisch verändern und auf eingehende Daten „reagieren“. Man kann Code in Komponenten einteilen und damit wiederverwenden. React steht als Open Source unter BSD-Lizenz zur Verfügung und bietet viele freie Erweiterungen und Module.

■ Bootstrap

Bootstrap bietet Standardlayouts und -stile für übliche Elemente eines Interfaces. Es gibt eine umfangreiche Anzahl unterstützter Elemente und Komponenten, die gut dokumentiert sind. Bootstrap wird von Twitter unter MIT-Lizenz entwickelt.

■ Webpack

Dieses Werkzeug bündelt alle benötigten Daten für die App. Es kann Codes optimieren, minimieren und verschleiern. Dadurch verbessern sich die Ladezeiten, da alle Daten sich im RAM befinden.

■ Cordova

Mit Cordova ist es möglich, die eigene App direkt als App für Plattformen, wie Android, iOS oder Windows 10 herauszubringen. Es bietet Zugriff auf die hardware- und gerätespezifischen Schnittstellen, wie zum Beispiel das Telefonbuch oder die Kamera.

■ Electron

Ist Cordova ähnlich, nur mit x86-basierten Computern als Ziel, wie Windows 10, MacOS und Linux.

Was passiert in Zukunft?

Aufgrund der hohen Beliebtheit von Javascript und der hohen Abdeckung von Betriebssystemen steht die Weiterentwicklung von Javascript besonders im Fokus der großen Internetunternehmen wie zum Beispiel Google. Dadurch wurden bereits viele neue Funktionen testweise implementiert und können selbst ausprobiert werden (<https://experiments.withgoogle.com>).

Mit WebVR können über Javascript direkt im Browser virtuelle Welten mit Headup-Displays erkundet werden. Der Endkunde muss dafür nur die Hardware besitzen und muss keine zusätzliche Software installieren.

Mit WebASM können bereits in Maschinencode übersetzte Programme bereitgestellt werden. Damit können firmeneigene Algorithmen geschützt werden.

WebGL ermöglicht grafikbeschleunigte Darstellungen im Browser. Besonders 3D-Darstellungen profitieren davon und es sind höhere Detailgrade möglich.

Progressive Web Apps sind browserbasierte Anwendungen, die keine Installation des Endnutzers benötigen. Die Grenze zwischen einer Webseite und einer App verschwimmt.

Ein erweiterter Zugriff auf Hardwarefunktionen wird möglich. Zum Beispiel können Javascript-Apps dann direkt auf das Bluetooth des Endnutzers zugreifen. Damit können Apps entwickelt werden, die direkt mit Geräten beim Endkunden kommunizieren können.

Maschinelles Lernen ist in Javascript bereits verfügbar. Dadurch können neuronale Netze in der App betrieben werden, die mehr Funktionalitäten ermöglichen.

Überblick

Mit Javascript können wir:

- Eine interaktive Oberfläche mit verschiedenen Medien gestalten.
- Einfache Prototypen bis fertige Apps entwickeln.
- Software plattformübergreifend entwickeln.
- An der zukünftigen Entwicklung teilhaben.
- Viele umfangreiche und schnelle Tools und Bibliotheken nutzen.

Anmerkungen

Diese Nachgelesen-Ausgabe beschreibt das komplexe Thema Javascript mit vielen Vereinfachungen und Verkürzungen, um Ihnen das Thema auf eine einfache Art und Weise darzulegen. Einige Fachbegriffe seien dennoch an dieser Stelle kurz erläutert.

- ¹ Ein Compiler übersetzt den auch für Menschen lesbaren Quellcode in Maschinensprache.
- ² Ein Interpreter verarbeitet den Quellcode zur Laufzeit.

Aus unserer Sicht eignet sich Javascript sehr gut dazu, neue und innovative Software zu entwickeln und gleichzeitig für die zukünftigen Entwicklungen auf dem Markt gerüstet zu sein. Moderne Funktionen und Möglichkeiten bieten einen besonderen Mehrwert für die eigenen Produkte.

Autoren

M. Sc. Christoph Neumann ist wissenschaftlicher Mitarbeiter an der Professur für Arbeitswissenschaft und Innovationsmanagement der TU Chemnitz. Im Kompetenzzentrum beschäftigt er sich mit dem Thema Industrial Engineering und Programmierung.

info@betrieb-machen.de

Weitere Informationen

Das Mittelstand 4.0-Kompetenzzentrum Chemnitz gehört zu Mittelstand-Digital. Mit Mittelstand-Digital unterstützt das Bundesministerium für Wirtschaft und Energie die Digitalisierung in kleinen und mittleren Unternehmen und dem Handwerk.

Was ist Mittelstand-Digital?

Mittelstand-Digital informiert kleine und mittlere Unternehmen über die Chancen und Herausforderungen der Digitalisierung. Die geförderten Kompetenzzentren helfen mit Expertenwissen, Demonstrationszentren, Best-Practice-Beispielen sowie Netzwerken, die dem Erfahrungsaustausch dienen. Das Bundesministerium für Wirtschaft und Energie (BMWi) ermöglicht die kostenfreie Nutzung aller Angebote von Mittelstand-Digital.

Der DLR Projektträger begleitet im Auftrag des BMWi die Kompetenzzentren fachlich und sorgt für eine bedarfs- und mittelstandsgerechte Umsetzung der Angebote. Das Wissenschaftliche Institut für Infrastruktur und Kommunikationsdienste (WIK) unterstützt mit wissenschaftlicher Begleitung, Vernetzung und Öffentlichkeitsarbeit.

Weitere Informationen finden Sie unter www.mittelstand-digital.de

IMPRESSUM:

Herausgeber:

Mittelstand 4.0-Kompetenzzentrum Chemnitz
Geschäftsstelle
c/o Technische Universität Chemnitz
Prof. Dr.-Ing. habil. Ralph Riedel
DE – 09107 Chemnitz
Tel: 0371 531 19935
Fax: 0371 531 819935
E-Mail: info@betrieb-machen.de
Web: www.betrieb-machen.de
www.kompetenzzentrum-chemnitz.digital

Redaktion & Gestaltung

M. Sc. Christoph Neumann & Anikó Lessi

Bildnachweis Titel:

Florian Olivo auf Unsplash