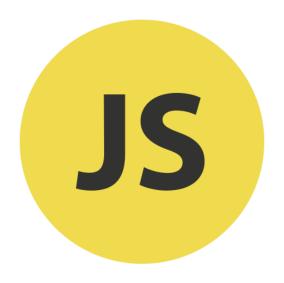
JavaScript

- Grundlegendes -



Grundlegendes zu JavaScript

JavaScript ist eine Sprache zur dynamischen Webseitengestaltung. JavaScript wurde erfunden, um Webseiten dynamisch im Browser zu ändern und Interaktionen mit dem Benutzer zu realisieren. Das ist heute immer noch der Haupteinsatz. Für die meisten Web-User ist der Browser einfach nur ein Programm zum Betrachten von Web-Seiten. Für den Web-Entwickler nimmt der Browser jedoch eine ganz andere Stelle ein: Er ist Laufzeit-, Debugging- und Testumgebung in einem und wichtiger Bestandteil damit ein Entwicklungsprozesses. Weil Web-Seiten möglichst auf allen Plattformen funktionieren sollen, wird JavaScript



nicht in Maschinencode kompiliert, sondern in Klartextform ausgeliefert und dann vom Browser interpretiert. JavaScript ist die einzige Sprache, die von fast allen vorhandenen Browsern nativ unterstützt wird.

Bei JavaScript handelt es sich um eine Programmiersprache, die speziell für Internetseiten entworfen wurde. Sie bietet viele Elemente, die auch aus anderen Programmiersprachen bekannt sind, wie zum Beispiel

- Schleifen zur schnellen Wiederholung von Programmteilen,
- Verzweigungen zur unterschiedlichen Behandlung verschiedener Situationen und
- Funktionen zur Zerlegung eines Programms in übersichtliche Bestandteile.











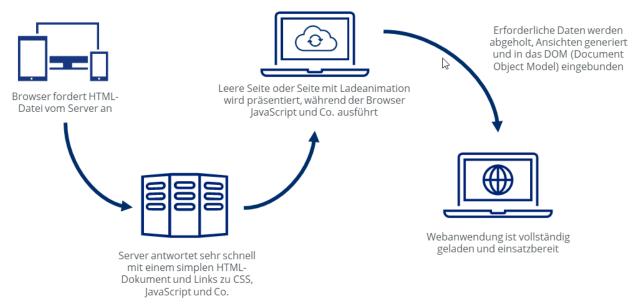


Außerdem hat man mit Hilfe von Objekten und dem Document Object Model (DOM) Zugriff auf alle Elemente der Internetseiten, so dass man sie dynamisch verändern kann. JavaScript-Programme werden den Benutzern innerhalb von Internetseiten zusammen mit HTML zur Verfügung gestellt. Sie werden auf dem Browser des Benutzers ausgeführt und können die Inhalte einer Internetseite dynamisch verändern. Dies

geschieht entweder sofort nach dem Laden der Internetseite oder nach dem Eintreten eines Ereignisses, zum Beispiel der Betätigung eines Buttons durch den Benutzer. JavaScript ermöglicht somit den Entwurf komplexer Anwendungen mit einer Benutzeroberfläche. JavaScript wurde entworfen, um dem Benutzer zusätzliche Möglichkeiten und Hilfen zu bieten, die er allein mit HTML nicht hat. JavaScript-Programme werden im Browser des Benutzers ausgeführt und nicht auf dem Webserver, von dem sie geladen werden.

_

Client-Side-Ausführung von JavaScript



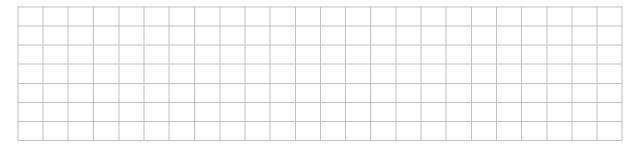
 $\label{eq:Quelle:partial} \textbf{Quelle:} \ \underline{\text{https://www.ionos.de/digitalguide/websites/web-entwicklung/server-side-und-client-side-scripting-die-unterschiede/digitalguide/websites/web-entwicklung/server-side-und-client-side-scripting-die-unterschiede/digitalguide/websites/web-entwicklung/server-side-und-client-side-scripting-die-unterschiede/digitalguide/websites/web-entwicklung/server-side-und-client-side-scripting-die-unterschiede/digitalguide/websites/web-entwicklung/server-side-und-client-side-scripting-die-unterschiede/digitalguide/websites/web-entwicklung/server-side-und-client-side-scripting-die-unterschiede/digitalguide/websites/web-entwicklung/server-side-und-client-side-scripting-die-unterschiede/digitalguide/websites/web-entwicklung/server-side-und-client-side-scripting-die-unterschiede/digitalguide/websites/web-entwicklung/server-side-und-client-side-scripting-die-unterschiede/digitalguide/websites/web-entwicklung/server-side-und-client-side-scripting-die-und-client-side-scripting-die-und-client-scripting-die-und-clie$

Die Technik des Client-Side-Renderings bzw. Client-Side-Scriptings wird von Webentwicklern vorrangig dazu verwendet, **Projekte mit dynamischen Inhalten** zu realisieren. Die programmierten Skripte werden in diesem Fall nicht vom Server, sondern **vom zugreifenden Browser ausgeführt** und verarbeitet. Zu diesem Zweck bettet man die Skripte entweder in das HTML-Dokument ein oder schreibt sie in eine separate Datei, die man mit dem Dokument verknüpft.

Ruft der Nutzer eine Webseite mit clientseitigen Skripten auf, sendet der Webserver das HTML-Dokument sowie die Skripte an den Browser, der diese selbst ausführt und das Endergebnis präsentiert. Clientseitige Skripte können darüber hinaus **konkrete Instruktionen für den Webbrowser** beinhalten, wie dieser auf bestimmte Aktionen des Nutzers, z. B. auf einen Button-Klick, reagieren soll. Oftmals muss der Client dazu keinen erneuten Kontakt zum Webserver aufbauen.



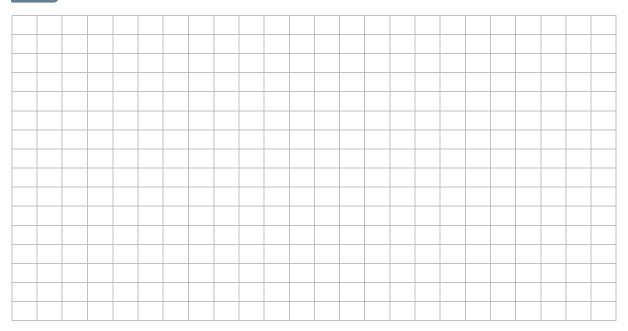
Nennen Sie einige typische Anwendungsgebiete von JavaScript. Recherchieren Sie bei Bedarf im Internet.



LF10a: Benutzerschnittstellen gestalten und entwickeln

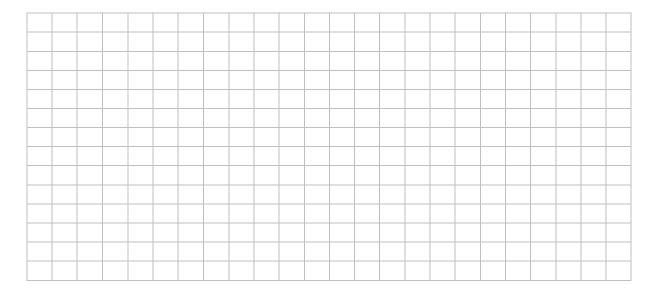


Sie wissen nun, dass JavaScript eine clientseitige Sprache ist. Nennen Sie einen Vorund einen Nachteil von client-seitigen-Sprachen.

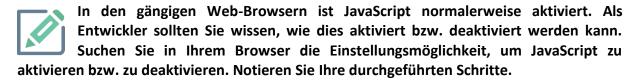




JavaScript – Eine Programmiersprache? Oder doch eine Skriptsprache? Erklären Sie den Unterschied der beiden Begriffe in eigenen Worten und ordnen Sie JavaScript nach Möglichkeit richtig ein.



_





Einbindung von JavaScript in HTML

Zur Erinnerung: Um JavaScript einsetzen zu können, müssen die Grundlagen von HTML bekannt sein. Denn JavaScript tritt im Regelfall zusammen mit HTML (und CSS) auf. Eine HTML Datei ist typischerweise in DOCTYPE, HEAD und BODY aufgeteilt.

JavaScript kann man auf 2 Optionen in eine HTML-Seite einbinden. Mit der ersten Art ist das JavaScript direkt auf der Seite.

Aufbau HTML5 Seite – alle Bereiche



Mit der zweiten Art wird JavaScript über eine externe Datei eingebunden.

Option 1: Einbinden von JavaScript direkt auf HTML-Seite (= Inline JavaScript)

Das Einbinden geschieht über folgenden Aufbau (i.d.R. im Head-Bereich der HTML-Seite)

```
<script language="javascript" type="text/javascript">
  <!--
    ... hier kommt das JavaScript-Programm
    -->
    </script>
```

Ab HTML5 ist das Einbinden von JavaScript noch einfacher. Dann reicht die Angabe <script> aus, also:

```
<script>
... hier kommt das JavaScript-Programm
</script>
```

Option 2: Einbinden von JavaScript in HTML-Seite als externe Datei (=External JavaScript)

Dazu wird im Head-Bereich der HTML-Datei folgender Zeile integriert:

Kürzer ist schöner - daher folgt hier die Kurzfassung. Wichtig ist einfach, dass man auch mal die längere gesehen hat, wenn man fremde Seiten analysiert.

```
<!DOCTYPE html>
<html lang="de">

<head>
  <title>JavaScript als externe Datei einbinden</title>
  <script src="dateiname.js"></script>
  </head>
  <body>
  ...
```



Sie kennen nun zwei Arten wie JavaScript in HTML eingebunden werden kann. Welche Vor- und Nachteile bringen diese beiden Optionen für Anwendungsentwickler mit sich? Beschreiben Sie dies in eigenen Worten.

| | Inline JavaScript | External JavaScript |
|-----------|-------------------|---------------------|
| Vorteile | | |
| Nachteile | | |

Gehen Sie auf die Internetseite Ihres Ausbildungsbetriebs. (Oder zum Beispiel auf www.passau.de) Lassen Sie sich dort im Browser den Quellcode anzeigen. Was fällt Ihnen bei der Einbindung von JavaScript auf?

Ein erstes JavaScript Programm erstellen



weggeklickt werden können.

Ihnen liegt der HTML-Code (siehe Abbildung 1) einer Webseite vor. Diese HTML-Seite soll um eine JavaScript-Funktion ergänzt werden.



Sobald die Webseite durch einen Anwender geöffnet wird, sollte eine Popup-Meldung erscheinen in welcher steht: "Diese Seite befindet sich noch im Aufbau". Die Meldung soll durch einen Klick auf "OK" vom User wieder

Arbeitsauftrag:

- 1. Tippen Sie den HTML-Code ab und speichern Sie diesen als *.html-Datei ab.
- 2. Öffnen Sie die HTML-Datei im Browser, um die Funktionalität der Seite zu prüfen.
- 3. Schreiben Sie den dazugehörigen JavaScript-Code und binden Sie diesen auf zwei Optionen in die HTML-Seite ein.
 - 1. Inline JavaScript
 - 2. External JavaScript

Sie haben am Ende der Übung also zwei HTML-Dateien.

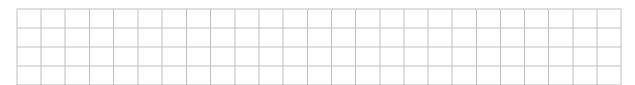


Tipp: Orientieren Sie sich für die Implementierung der Popup-Meldung an der JavaScript Methode "alert()". Recherchieren Sie bei Bedarf im Internet über deren Funktionalität und Syntax.

```
<!DOCTYPE html>
<html lang="de">
  <head>
    <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Titel</title>
 </head>
 <body>
    "Das ist die neue Webseite der Firma UI GmbH"
  </body>
</html>
```

Abbildung 1: HTML-Code der Webseite

Was fällt Ihnen bei der Einbindung von External JavaScript besonders auf?

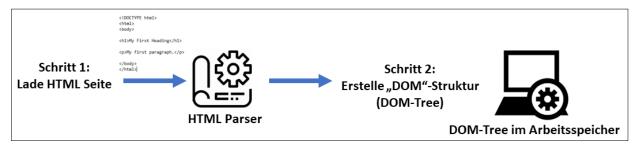


Dr 10a. Denatzersemmustenen gestatten und entwicken

Das Document Object Model (DOM)

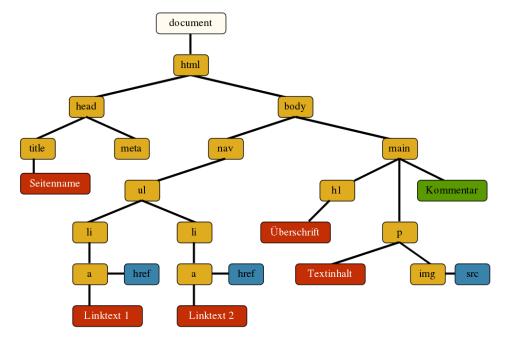
Das DOM (Document Object Model) ist die Schnittstelle zwischen HTML und dynamischem JavaScript. Alle Elemente werden zu Objekten, die dynamisch aufgerufen, verändert, hinzugefügt und gelöscht werden können.

Eine Webseite liegt dem Browser zunächst als bloßer Text, der mit der Auszeichnungssprache HTML formatiert ist, vor. Noch während der Browser den Code über das Netz empfängt, verarbeitet er ihn Stück für Stück. Diese Aufgabe übernimmt der sogenannte Parser (englisch parse = einen Satz in seine grammatikalischen Einzelteile zerlegen).



Der Parser überführt den HTML-Code in eine Objektstruktur, die dann im Arbeitsspeicher vorgehalten wird. Diese Objektstruktur besteht aus verschachtelten Knoten, allen voran Elementknoten, Attributknoten und Textknoten, die in einer Baumstruktur angeordnet sind.

Der Browser nutzt für alle weiteren Operationen diese Objektstruktur, nicht den HTML-Quellcode, an dem der Webautor üblicherweise arbeitet. Insbesondere CSS und JavaScript beziehen sich nicht auf den HTML-Code als Text, sondern auf den entsprechenden Elementen-Baum im Speicher.



Die einzelnen Bestandteile einer solchen Baumstruktur werden als Knoten bezeichnet. Das zentrale Objekt des DOM ist deshalb das Objekt Node (node = Knoten). Es gibt verschiedene Knotentypen. Innerhalb eines gewöhnlichen HTML-Dokuments gibt es auf jeden Fall drei wichtige Knotentypen, die Sie unterscheiden müssen:

- Elementknoten
- Attributknoten
- Textknoten



In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.



Betrachten Sie den nachfolgenden HTML-Code. Skizzieren Sie das dazugehörige Document Object Model mit seinen einzelnen Knoten.

Anwendungsentwicklung und Programmierung LF10a: Benutzerschnittstellen gestalten und entwickeln

| DOM: | |
|------|--|
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

-



Üben Sie den praktsichen Umgang mit dem Document Object Model, indem Sie die Praxisbeispiele unter folgenden Links testen.

| DOM Methoden | https://www.w3schools.com/js/js htmldom methods.asp |
|--------------|--|
| DOM Document | https://www.w3schools.com/js/js htmldom document.asp |

Während Sie die verschiedenen Code-Beispiele testen, beantworten Sie folgende Fragen.

Was bewirkt die Methode "getElementbyld()"? Notieren Sie ein passendes Code-Bespiel.

The most common way to access an HTML element is to use the id of the element.

<script>document.getElementById("demo").innerHTML = "Hello World!";</script>

Wozu kann die Eigenschaft "innerHTML" in JavaScript genutzt werden?

The innerHTML property can be used to get or change any HTML element, including httml and <body>.

Welchen Zweck erfüllt das "Document" Object?

The document object represents your web page.
If you want to access any element in an HTML page, you always start with accessing the document object.

Sie wollen aus einer HTML Seite (DOM) alle ** Elements** finden. Welchen Befehl verwenden Sie?

const element = document.getElementsByTagName("img");