Name Auszubildende/Auszubildender:

LITERATURLISTE • JAVASCRIPT

Animationen
Die Methoden *setTimeout(), setInterval()* und *requestAnimationFrame()* können verwendet werden, um Inhalte im Browser-Fenster zu animieren.
https://www.mediaevent.de/javascript/animation.html

Ausgabe
alert() öffnet ein Fenster mit einer Ausgabe, die auf Nutzereingaben wartet. Mit *print()* lassen sich Ausgaben an den Drucker senden.
https://www.mediaevent.de/javascript/output.html

Cookies und Sessions
JavaScript kann verwendet um in sogenannten *Cookies* Daten zu speichern, die bestehen bleiben, wenn der Nutzer die Seite zu einem anderen Zeitpunkt erneut aufruft oder der Nutzer durch mehrere Unteseiten eines Webauftritts navigiert.
Typische Anwendungsbeispiele sind, Formular-Eingaben auch bei Nutzung des Zurück-Buttons verfügbar zu halten oder den Login-Status eines Nutzers über verschiedene Unterseiten aufrecht zu erhalten und dadurch zu verhindern, dass sich der Nutzer beim Aufruf einer neuen Seite jedes Mal neu einloggen muss.
https://www.mediaevent.de/javascript/cookies.html
https://wiki.selfhtml.org/wiki/JavaScript/Tutorials/cookies
https://www.kostenlose-javascripts.de/tutorials/cookies/

CSS ändern
JavaScript kann auch verwendet werden, um die Darstellung der Webseite durch CSS zu verändern.
https://www.mediaevent.de/javascript/DOM.html

Currying und funktionale Programmierung
Bei der funktionalen Programmierung handelt es sich um ein Programmierparadigma, das in den letzten Jahren einen großen Popularitätsschub erfahren hat und helfen soll, Komplexität begreifbarer zu machen und wartungsfreundlichen Code fördert.
https://de.wikipedia.org/wiki/Currying#Anwendung
http://webkrauts.de/artikel/2015/grundlagen-funktionaler-programmierung
https://entwickler.de/online/windowsdeveloper/funktionales-javascript-579832242.html
http://currybuch.de

Dateien laden und senden
XMLHttpRequest 2 macht es möglich, das mittels JavaScript Dateien vom Server geladen werden und auch auf den Server gespeichert werden können.
https://www.html5rocks.com/de/tutorials/file/xhr2/

Datentypen

JavaScript kann Daten unterschiedlicher Beschaffenheit speichern. Unter anderen werden Daten als *Number*, *String*, Objekt oder boolscher Wert gespeichert.

https://www.mediaevent.de/javascript/Javascript-Basis-Datentypen.html
https://www.mediaevent.de/javascript/typeof.html
https://www.mediaevent.de/javascript/Javascript-Strings.html
https://www.mediaevent.de/javascript/array.html
https://wiki.selfhtml.org/wiki/JavaScript/Datentyp
https://developer.mozilla.org/de/docs/Web/JavaScript/Guide/Grammatik_und_Typen#Datenstrukturen_und_-typen

Debugging

Mittels dem Befehls *console.log()* können Ausgaben an die Konsole generiert werden. Dies ist zum Beispiel hilfreich bei der Fehlersuche, indem der Inhalt einer Variablezu einem bestimmten Zeitpunkt über die Konsole überprüft werden kann. Chrome, Firefox und Safari bieten außerdem Debugging-Tools, mit denen der Code schrittweise ausgeführt werden kann.

https://www.mediaevent.de/javascript/fehler-suchen.html
https://medium.com/appsflyer/10-tips-for-javascript-debugging-like-a-pro-with-console-7140027eb5f6
https://www.sgalinski.de/typo3-agentur/technik/javascript-im-google-chrome-debuggen/
https://developer.mozilla.org/de/docs/Tools/Debugger

DOM-Manipulation

JavaScript Scripte können direkt auf einzelne HTML-Baum-Knoten zugreifen, diese verändern, entfernen oder neue erstellen.

https://wiki.selfhtml.org/wiki/JavaScript/DOM
https://www.mediaevent.de/javascript/DOM.html
https://developer.mozilla.org/de/docs/Web/API/ChildNode/remove

Event-Handling

Üblicherweise werden JS-Scripte erst ausgeführt, wenn ein definiertes Ereignis eintritt – zum Beispiel das Anklicken eines Buttons, das Verändern der Größe eines Browserfensters, das vollständige Laden einer HTML-Seite etc.
JavaScript bietet zur Überwachung dieser Events unterschiedliche Eventhandler.

https://molily.de/js/konzepte.html#events
https://www.mediaevent.de/javascript/events.html

Funktionen

Möchten wir Code-Bestandteile mehrmals an unterschiedlichen Stellen im Code wiederverwenden, können wir diese in Funktionen auslagern. Das spart nicht nur Schreibarbeit sondern macht spätere Änderungen einfacher und hilft, Fehler zu vermeiden. Das zugrunde liegende Prinzip nennt sich *Don't repeat yourself*.

https://developer.mozilla.org/de/docs/Web/JavaScript/Guide/Funktionen
https://wiki.selfhtml.org/wiki/JavaScript/Funktion
https://de.wikipedia.org/wiki/Don't_repeat_yourself

if- & switch-Abfrage

Möchten wir, dass unser Code auf unterschiedliche Bedingungen unterschiedlich reagiert, können wir dies mit einer If-Else-Abfrage erstellen. *switch* tut das gleiche, bietet nur eine komfortablere Schreibweise. Der *Ternary-Operator* bietet eine noch reduziertere Schreibweise.

https://wiki.selfhtml.org/wiki/JavaScript/Verzweigung
https://developer.mozilla.org/de/docs/Web/JavaScript/Guide/Kontrollfluss_und_Fehlerbehandlung#Bedingte_Statements
https://www.mediaevent.de/javascript/Javascript-Anweisungen-Bedingungen.html
https://www.mediaevent.de/javascript/switch.html
https://www.mediaevent.de/javascript/ternary-if-then-else.html

JavaScript Code einbinden

Um JavaScript in HTML-Dokumenten ausführen zu können, muss der Code in eine HTML-Datei eingebunden werden. Dazu kann der Code direkt in die HTML-Datei geschrieben werden oder es wird eine externe Datei eingebunden.

https://www.mediaevent.de/javascript/

Laufzeitfehler

Mit *try*, *catch* und *error* gibt es verschiedene Möglichkeiten, Fehler in der Ausführung – sogenannte Laufzeitfehler – abzufangen.

https://www.mediaevent.de/javascript/try-catch-throw-exceptions.html
https://developer.mozilla.org/de/docs/Web/JavaScript/Guide/Kontrollfluss_und_Fehlerbehandlung#Statements_zur_Fehler-_bzw._Ausnahmebehandlung

Objektorientierte Programmierung

Objektorientierte Programmierung soll helfen, Code übersichtlicher zu gestalten und ermöglicht damit die Umsetzung sehr komplexer Anforderungen. Durch das Konzept der Kapselung wird der Code zudem robuster und eine kollaborative Entwicklung einfacher. Auch steigert die Modularität objektorientierter Software die Verständlichkeit des Codes.

https://developer.mozilla.org/de/docs/Web/JavaScript/Introduction_to_Object-Oriented_JavaScript
https://wiki.selfhtml.org/wiki/JavaScript/Objekte_-_Eigenschaften_und_Methoden
https://wiki.selfhtml.org/wiki/JavaScript/Module_und_Kapselung
https://wiki.selfhtml.org/wiki/JavaScript/Vererbung
https://wiki.selfhtml.org/wiki/JavaScript/this
https://www.mediaevent.de/javascript/Javascript-Objekte-1.html
https://www.mediaevent.de/javascript/object.html
https://www.mediaevent.de/javascript/Javascript-Objekte-3.html
https://www.mediaevent.de/javascript/object-prototype.html
https://www.mediaevent.de/javascript/Javascript-Objekte-2.html

Name Auszubildende/Auszubildender:

Operatoren

Um zwei oder mehr Operanden miteinander zu vergleichen oder einem Operanden einen Wert zuzuweisen, benötigen wir Operatoren. Auch um Operanden miteinander zu multiplizieren, addieren oder subtrahieren etc. werden Operatoren benötigt. Wir unterscheiden dabei u. a. zwischen Zuweisungs- und Vergleichs-Operatoren sowie arithmetischen Operatoren.

https://developer.mozilla.org/de/docs/Web/JavaScript/Guide/Ausdruecke_und_Operatoren
https://www.mediaevent.de/javascript/javascript-basis-vergleichen.html
https://www.mediaevent.de/javascript/assignments.html
https://www.mediaevent.de/javascript/berechnung.html
https://www.mediaevent.de/javascript/Javascript-Basis-Operatoren.html
https://www.mediaevent.de/javascript/delete.html
https://wiki.selfhtml.org/wiki/JavaScript/Operatoren

Reguläre Ausdrücke

Reguläre Ausdrücke ermöglichen es, Zeichenketten nach unterschiedlichen Vorgaben zu überprüfen. So kann zum Beispiel verifiziert werden, ob eine Nutzereingabe zum Beispiel eine E-Mail-Adresse oder eine Postleitzahl sein kann.

https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Global_Objects/RegExp#grouping-back-references
https://wiki.selfhtml.org/wiki/JavaScript/Objekte/RegExp
https://www.mediaevent.de/javascript/Javascript-Regulaere-Ausdruecke-2.html

Schleifen

while, do-while, for, forEach, for...of und *for...in* sind verschiedene Möglichkeiten, Codebestandteile in einer Schleife laufen zu lassen, solange eine vordefinierte Bedingung nicht erfüllt wurde.

https://developer.mozilla.org/de/docs/Web/JavaScript/Guide/schleifen_und_iterationen
https://developer.mozilla.org/de/docs/Web/JavaScript/Reference/Statements/for...of
https://www.mediaevent.de/javascript/while-schleife.html
https://www.mediaevent.de/javascript/Basis-Schleifen-for-while.html
https://www.mediaevent.de/javascript/for-in-foreach.html
https://www.mediaevent.de/javascript/break-continue-label.html

Server-Interaktion

Mithilfe des XMLHttpRequest Objektes können im Hintergrund Daten an den Server gesendet oder vom Server empfangen werden. So kann zum Beispiel eine Anfrage an den Server gestellt werden, ob eine eingegebene Kombination von Nutzername und Kennwort valide ist.

https://wiki.selfhtml.org/wiki/JavaScript/XMLHttpRequest
https://developer.mozilla.org/de/docs/Web/API/XMLHttpRequest
https://www.html5rocks.com/de/tutorials/file/xhr2/

Standortbestimmung

getCurrentPosition() ermöglicht, den momentanen Standort des Nutzers zu ermitteln.

https://www.mediaevent.de/javascript/geolocation.html

Unit-Tests und End-To-End-Tests

Der Quellcode moderner Software-Produkte wird laufend verändert. Manche Änderungen verursachen Fehler an unerwarteten Stellen, weshalb nach erfolgter Änderung das Software-Produkt möglichst Lückenlos getestet werden muss. Dabei unterscheidet man zwischen Unit-Test, bei denen jede einzelne Funktion im Code getestet wird und End-To-End-Tests, bei denen die Units der Software im Zusammenhang getestet werden. Da die manuelle Ausführung sehr zeitaufwendig und Fehleranfällig ist, gibt es Frameworks, die automatisierte Tests ermöglichen.

https://blog.codecentric.de/2017/06/javascript-unit-tests-sind-schwer-aufzusetzen-keep-calm-use-jest/

https://jasmine.github.io/

Variablen

JavaScript kennt drei verschiedene Möglichkeiten, Variablen zu deklarieren: *const, let* und *var.* Diese können entweder eine lokale oder eine globale Gültigkeit haben.

https://www.mediaevent.de/javascript/variable.html

https://www.mediaevent.de/javascript/globale-lokale-variablen.html

https://www.mediaevent.de/javascript/let.html