

Einführung in JavaScript

Basics und Aufgaben

Salome Treichler

26.05.2020

Version 1.0

Inhaltsverzeichnis

Abbildungsverzeichnis	2
Einführung.....	3
Basics.....	4
Was ist JavaScript?	4
Document Object Model (DOM)	5
Datentypen und Variablen in JavaScript	6
Datentypen	6
Dynamische Typisierung.....	6
Vorbereitung für Aufgaben.....	7
Download von Beispiel Projekt	7
Installation Visual Studio Code.....	9
Aufgaben	11
Hello World	11
Number Guessing Game	16
Math.random()	16
parseInt()	16
isNaN()	16
alert()	16
Array.push()	17
setAttribute() und removeAttribute()	17
Drumkit.....	18
Event-Listener keydown	18
Event-Listener transitioned	18
Document.querySelector()	18
Document.querySelectorAll().....	18
element.classList.add() und .remove()	19
audio.play().....	19
Wissens Checkliste	20
Quellenverzeichnis.....	21

Abbildungsverzeichnis

Abbildung 1 – DOM Baumstruktur (Quelle: https://wiki.selfhtml.org/wiki/JavaScript/DOM)	5
Abbildung 2 - Repository herunterladen Schritt 1 (Quelle: Eigene Aufnahme)	7
Abbildung 3 - Repository herunterladen Schritt 2 (Quelle: Eigene Aufnahme)	7
Abbildung 4 - Download Ordner mit Zip-File (Quelle: Eigene Aufnahme)	8
Abbildung 5 - Menü Zip-File Rechtsklick (Quelle: Eigene Aufnahme)	8
Abbildung 6 - Extract File Fenster (Quelle: Eigene Aufnahme)	8
Abbildung 7 - Download Links Visual Studio Code (Quelle: Eigene Aufnahme)	9
Abbildung 8 - File Menü Visual Studio Code (Quelle: Eigene Aufnahme)	10
Abbildung 9 - Hello World neues File (Quelle: Eigene Aufnahme)	11
Abbildung 10 - JavaScript Link in HTML (Quelle: Eigene Aufnahme)	12
Abbildung 11 - Hello World ID in HTML Elementen (Quelle: Eigene Aufnahme)	12
Abbildung 12 - Hello World Variablen Deklaration (Quelle: Eigene Aufnahme)	12
Abbildung 13 - Hello World Event-Listener (Quelle: Eigene Aufnahme)	13
Abbildung 14 - Go Live Funktion (Quelle: Eigene Aufnahme)	14
Abbildung 15 - Hello World Ergebnis (Quelle: Eigene Aufnahme)	14
Abbildung 16 - Menü bei Rechtsklick auf Webseite (Quelle: Eigene Aufnahme)	14
Abbildung 17 - DevTolls Fenster (Quelle: Eigene Aufnahme)	15

Einführung

Dieses Dokument ist eine kleine Einführung in die Welt der Website Entwicklung mit JavaScript. Damit Sie diese Anleitung verstehen, wird vorausgesetzt, dass Sie bereits Erfahrungen haben mit den Grundprinzipien des Programmierens und bereits Wissen haben in den folgenden Punkten:

- HTML/CSS
- If Verzweigungen
- Verschiedenen Loops wie for, foreach, dowhile, while
- Verständnis von Funktionen / Methoden
- Grundvokabular der Informatik

Falls sie diese Punkte erfüllen ist dieses Dokument perfekt für einen ersten Einblick in die Skriptsprache JavaScript.

Das Dokument ist aufgeteilt in einen kurzen Baisc Teil von Wissen rundum JavaScript. Danach wird es einen Aufgaben Teil geben, mit dem sie sich praktisch mit JavaScript vertraut machen können. Alle Unterlagen und auch das Beispiel Projekt für die Aufgaben befinden sich auf meinem GitHub-Repository und sind für jeden kostenlos verfügbar. Den Link dazu finden sie in der Anleitung für den Download der Files.

Falls sie zu irgendetwas Fragen haben, stehe ich ihnen gerne über mein LinkedIn Profil zur Verfügung:

<https://www.linkedin.com/in/salome-treichler-ab560a1b0/>

Basics

Was ist JavaScript?

JavaScript oder kurz JS ist eine Skript Sprache. Sie wurde ursprünglich 1995 von Netscape entwickelt. JavaScript wurde entwickelt, um die Möglichkeiten von HTML und CSS zu erweitern. Es wurde vor allem entwickelt, um HTML dynamisch zu machen, um Benutzerinteraktionen auszuwerten und Inhalte zu verändern.

JavaScript ist eine objektorientierte, jedoch klassenlose Programmiersprache. Somit hat JavaScript trotz der ähnlichen Namensgebung nicht viel gemeinsam mit Java. Der Syntax ähnelt eher der prozeduralen Programmiersprache C.

Ein paar Typische Anwendungsgebiete von JavaScript sind...

- Dynamische Manipulation von Webseiten über das **Document Object Model (DOM)**
- Datenvalidierung (Richtigkeit der eingegebenen Daten überprüfen) von Formulareingaben vor der Übertragung zum Server
- Anzeigen von Dialogfenstern
- Verschleierung von E-Mail-Adressen zur Bekämpfung von Spam
- Senden und Empfangen von Daten, ohne dass der Browser die Seite neu laden muss
- ...

Mit JavaScript kann aber auch viel ungutes gemacht werden. Es gibt Webseiten, die in ihrem JavaScript Code gegen den Wunsch des Benutzers agieren. Deshalb bieten manche Browser auch an, JavaScript-Funktionen zu unterdrücken.

Ein paar Beispiele für solche Missbräuche sind...

- Unaufgeforderte Pop-ups oder Pop-unders
- Aufeinanderfolgende Dialogfenster die den Benutzer behindern
- Verschleiern von Internetadressen, auf die ein Link verweist
- ...

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.

Ein Browser analysiert ein HTML Dokument welches er öffnet und erstellt daraus im Arbeitsspeicher das DOM. Das DOM ist wie ein Baum aufgebaut. Hier ein Beispiel:

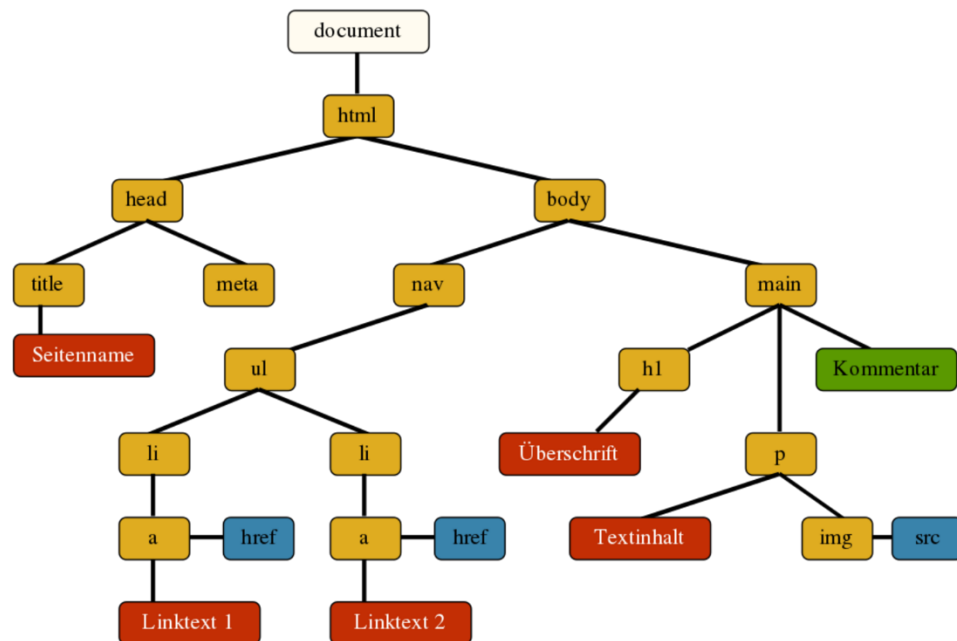


Abbildung 1 – DOM Baumstruktur (Quelle: <https://wiki.selfhtml.org/wiki/JavaScript/DOM>)

Dieser Elementbaum der nun erstellt wurde ist jedoch nicht statisch.

Durch DOM-Manipulation mit JavaScript können Daten ausgelesen und die Struktur verändert und erweitert werden

Diese Methoden können dazu verwendet werden:

- `getElementById()`: kann auf Elemente zugreifen, die ein dokumentweit eindeutiges `id`-Attribut enthalten
- `getElementsByName()`: kann auf Elemente zugreifen, die einen Namen besitzen (er muss nicht unbedingt eindeutig sein)
- `getElementsByTagName()`: kann auf alle Elemente zugreifen in der Form: "liefere mir das 27. `td`-Element im Dokument".
- `querySelector()`: gibt das erste Element zurück, das dem angegebenen CSS-Selektor entspricht.
- `querySelectorAll()`: gibt eine Liste von Elementen zurück, die dem angegebenen CSS-Selektor (auch mehrere, durch Komma getrennte, Angaben möglich) entsprechen

Die Methoden, die wir brauchen werden, werden später bei den Aufgaben erklärt.

Datentypen und Variablen in JavaScript

Datentypen

- Primitive Datentypen:
 - Boolean (logische Einheit, kann zwei Werte annehmen: true und false)
 - Null (hat nur einen Wert: null)
 - Undefined (Wert einer Variablen die noch keinen Wert zugewiesen hat)
 - Number (mögliche Werte: Gleitkommazahlen, +Infinity, -Infinity, NAN(not a number))
 - String (Menge von Elementen, textuelle Daten)
- Objekte

Dynamische Typisierung

JavaScript ist eine schwach typisierte oder dynamische Programmiersprache. Datentypen werden bei einer Variablen nicht explizit deklariert und jede Variable kann mit Werten jedes Typen beschrieben (überschrieben) werden:

```
var foo = 42; // foo ist jetzt eine number
```

```
var foo = "bar"; // foo ist jetzt ein string
```

```
var foo = true; // foo ist jetzt ein boolean
```

Vorbereitung für Aufgaben

Download von Beispiel Projekt

Für den Download des Beispiel Projekts, welches Sie später für die Aufgaben brauchen, befolgen Sie bitte folgende Schritte:

1. Öffnen Sie den Link zu meinem GitHub Repository über den QR Code in der oberen rechten Ecke dieser Seite oder über den folgenden Link:

<https://github.com/SalomeTreichler/javascriptbasicstutorial>



2. Es wird sich die folgende Ansicht öffnen:

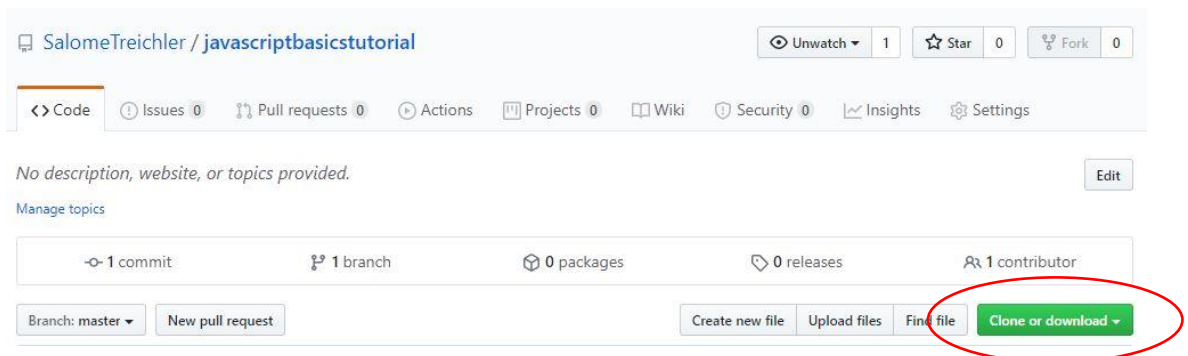


Abbildung 2 - Repository herunterladen Schritt 1 (Quelle: Eigene Aufnahme)

Klicken Sie nun auf den Grünen "Clone or Download" Button (rot markiert)

3. Klicken Sie nun auf "Download Zip" um alle Files als Zip herunterzuladen.

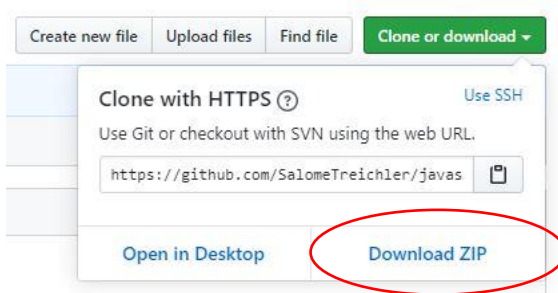


Abbildung 3 - Repository herunterladen Schritt 2 (Quelle: Eigene Aufnahme)

4. Gehen Sie nun in Ihren Download-Folder im File Explorer, dort sollten Sie nun ein neues Zip-File finden.

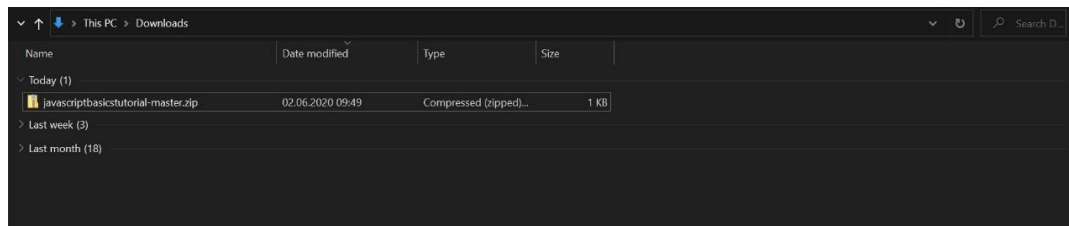


Abbildung 4 - Download Ordner mit Zip-File (Quelle: Eigene Aufnahme)

5. Nun müssen Sie das File nur noch extrahieren. Dafür machen Sie einen Rechtsklick auf das File wobei sich dann folgendes Menü öffnet:

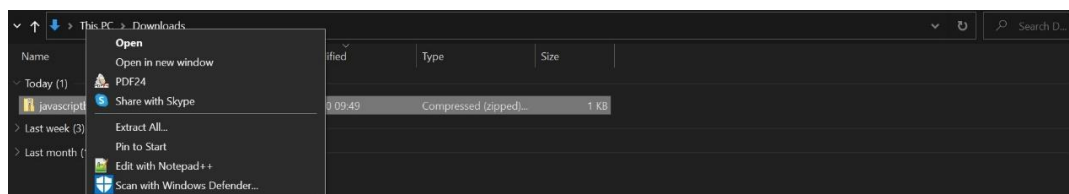


Abbildung 5 - Menü Zip-File Rechtsklick (Quelle: Eigene Aufnahme)

Klicken sie nun auf "Extract All..."

6. Nun wird ein neues Fenster geöffnet werden, wo Sie den gewünschten Folder für das Extrahieren der Files wählen können. Wählen Sie nun den gewünschten Folder aus und Klicken sie auf Extract

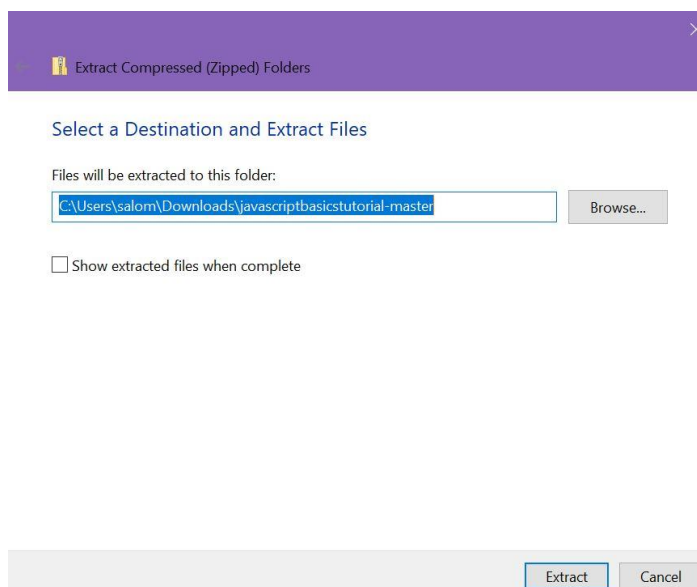


Abbildung 6 - Extract File Fenster (Quelle: Eigene Aufnahme)

Installation Visual Studio Code

Ich habe für mein Projekt Visual Studio Code als Editor verwendet. Es gibt auch viele andere Editoren für HTML, CSS und JavaScript. Wählen Sie den Editor aus mit dem Sie sich am wohlsten fühlen. Ich werde nun kurz den Installation Prozess von Visual Studio Code erklären und zeigen, wie man ein Projekt Folder öffnen kann.

1. Folgen Sie dem folgenden Link auf die offizielle Download Seite von Visual Studio Code:

<https://code.visualstudio.com/download>

2. Wählen Sie nun Ihr Betriebssystem aus (Ich nehme Windows) indem sie auf die Blaue Fläche unter dem Logo ihres Betriebssystems klicken.

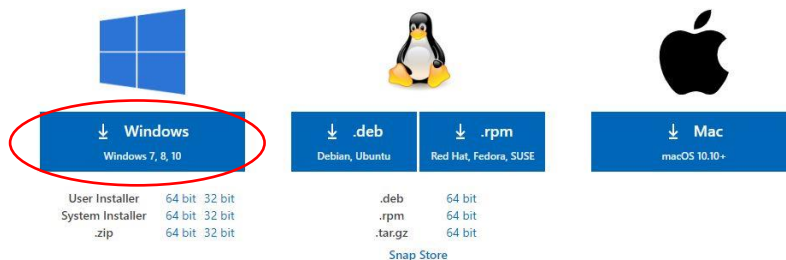


Abbildung 7 - Download Links Visual Studio Code (Quelle: Eigene Aufnahme)

Nun wird ein neuer Tab geöffnet mit Infos zu Visual Studio Code und ein paar Start Tipps. Diesen können Sie wieder schliessen.

3. Sobald der Download des ".exe" Files abgeschlossen ist, können Sie es öffnen. Nun wird der Installer gestartet. Sie können alle Installationseinstellungen beim Standard belassen. Schliessen Sie den Vorgang ab und öffnen Sie das Programm nach der Installation.

Nach der Installation können wir nun das Beispiel Projekt mit Visual Studio Code öffnen.

1. Öffnen Sie das Programm Visual Studio Code, falls Sie dies noch nicht gemacht haben.
2. Klicken Sie Oben Links auf File. Und dann auf "open Folder". Nun geht ein Dialog Fenster auf in welchem sie den gewünschten Folder auswählen können. Wählen Sie den Folder aus in dem Sie zuvor das Beispielprojekt abgelegt haben.

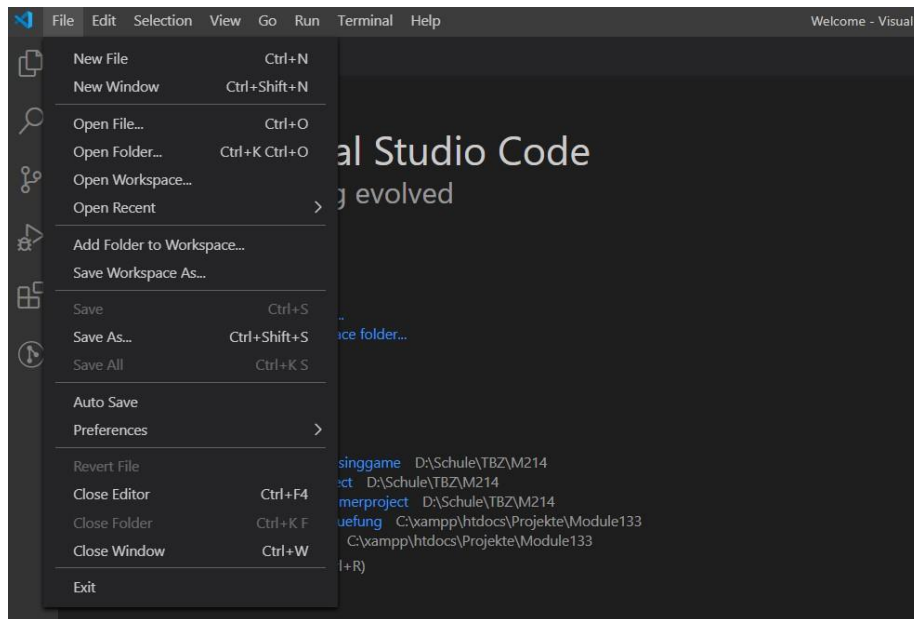


Abbildung 8 - File Menü Visual Studio Code (Quelle: Eigene Aufnahme)

Aufgaben

Hier beginnt der Aufgaben Teil. Die erste Aufgabe werde ich Ihnen Schritt für Schritt erklären, damit Sie sich ein bisschen besser in das ganze einfinden können. Die weiteren Aufgaben habe ich von einer Website mit Projekt Ideen. Ich werde Ihnen dazu den Link der originalen Aufgabe geben, damit Sie einen Überblick haben, was das Ziel der Aufgabe ist. Danach werde ich Ihnen einzelne Elemente und Funktionen erklären, welche sie brauchen werden, um die Aufgabe zu lösen.

Die Lösungen für die Aufgaben finden Sie unter dem folgenden Link:

<https://github.com/SalomeTreichler/javascriptbasictutorialsolutions>

Hello World

Wie Sie wahrscheinlich schon wissen, ist es eine Tradition, in einer neuen Programmiersprache zuerst ein Hello World Programm zu programmieren, das ist auch das um was es in dieser ersten Aufgabe gehen wird.

Das Ziel der Aufgabe ist es, dass beim klick auf einen Button "Hello World" auf der Seite ausgegeben wird. Ausserdem soll in der Konsole ein Logging gemacht werden, dass der Button gedrückt wurde.

1. Als erstes erstellen Sie ein neues JavaScript File mit rechtsklick auf den "HelloWorld" Folder und nennen es "helloWorld.js"

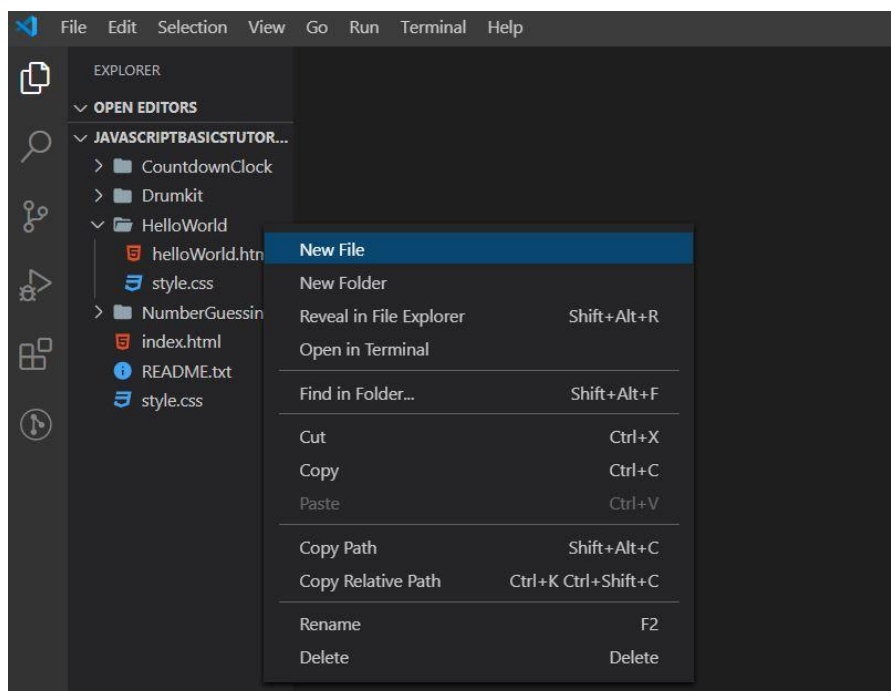


Abbildung 9 - Hello World neues File (Quelle: Eigene Aufnahme)

2. Öffnen sie nun das File "helloWorld.html". Sie müssen nun das JavaScript File, welches Sie gerade erstellt haben im HTML-File verlinken, dass es der Browser dann auch ausführt. Um dies zu tun, fügen sie folgende Linie Code zuunterst im HTML-Body ein.

```
47     <!--Add Scripts here-->
48     <script src="app.js"></script>
49 </body>
```

Abbildung 10 - JavaScript Link in HTML (Quelle: Eigene Aufnahme)

3. Nun widmen wir uns den beiden Komponenten, die wir für diese kleine Programm brauchen werden. Um mit den Komponenten aus dem DOM in JavaScript arbeiten zu können, müssen Sie ihnen eine Eindeutige ID geben.
Die zwei Elemente, die Sie verwenden werden, sind der leere Paragraph und der Button, also müssen Sie diesen zwei Elementen nun eine eindeutige ID geben.

```
41     <!--Content-->
42     <h3 class="page-title">Hello World</h3>
43     <p id="hello-world"></p>
44     <button class="btn btn-primary" id="button">Click Me!</button>
```

Abbildung 11 - Hello World ID in HTML Elementen (Quelle: Eigene Aufnahme)

Ich habe für mein Beispiel diese beiden IDs gewählt, Sie können aber auch eigene wählen. Wichtig ist nur, dass Sie dann im JavaScript File die gleichen wie hier verwenden.

4. Jetzt haben Sie die nötigen Vorbereitungen getroffen und Sie können ins "helloWorld.js" File wechseln.
5. Zuerst holen Sie die Elemente aus dem DOM und speichern sie in zwei Variablen.

```
1 //Variables with elements from DOM
2 let paragraph = document.getElementById("hello-world");
3 let button = document.getElementById("button");
4
```

Abbildung 12 - Hello World Variablen Deklaration (Quelle: Eigene Aufnahme)

Deklarieren Sie die Variablen wie Sie es im Kapitel "Basics – Datentypen und Variablen in JavaScript" erklärt bekommen. Um die Elemente in den Variablen zu speichern, müssen Sie die Methode "getElementById()" verwenden. Als Parameter geben Sie die ID, die Sie dem Element im HTML File gegeben haben, als String mit. Die Methode "getElementById()" greift auf ein HTML Element zu, das ein eindeutiges ID-Attribut besitzt.

6. Der nächste Schritt ist einen Event-Listener auf den Button zu machen. Es gibt verschiedene Event-Listener, in diesem Fall sollten Sie einen Click-Listener verwenden. Das bedeutet, dass die Funktion, die Sie dem Event-Listener mitgeben, ausgeführt wird, sobald der Benutzer auf den Button klickt. So ein Event-Listener kann man wie folgt hinzufügen:

```
5 button.addEventListener('click', function (event) {  
6     event.preventDefault();  
7     //Console log for button clicked  
8     console.log("Button has been clicked");  
9     //Display Hello World in paragraph tag  
10    paragraph.innerHTML = "Hello World";  
11 });
```

Abbildung 13 - Hello World Event-Listener (Quelle: Eigene Aufnahme)

Schauen wir uns diesen Code nun Zeile für Zeile an.

Auf Zeile 5 wird der Event-Listener dem Button hinzugefügt. In der Klammer wird in Anführungs- und Schlusszeichen mitgegeben, dass es ein "click" Event ist. Dann wird eine Funktion mitgegeben, mit dem Parameter "event". Diesen Parameter brauchen wir in der Nächsten Zeile.

Interaktive Elemente, sowie ein Button, haben oft ein browsereigenes Standartverhalten. Da Sie aber nicht die browsereigene Funktion, sondern Ihre eigene Funktion ausführen möchten, brauchen Sie den Code auf Zeile 6. Die Zeile 6 sorgt dafür, dass die Standartaktion des Browsers abgebrochen wird.

Als Nächstes soll in der Konsole ausgegeben werden, dass der Button gedrückt wurde. Dazu verwenden Sie "console.log()" und geben als Parameter einen String mit. Das Logen in der Konsole hilft zum Beispiel bei der Fehlersuche. Zum besseren Verständnis ein kleines Beispiel:

Hello World wird nicht auf der Seite angezeigt. Um herauszufinden, ob die Funktion überhaupt aufgerufen wird, machen Sie einen Log in der Konsole mit einer sinnvollen Message, wie zum Beispiel "Der Button wurde geklickt". Wenn dann der Button betätigt wird und die Nachricht nicht in der Konsole erscheint wissen Sie, dass die Funktion gar nicht aufgerufen wurde und können das Problem eingrenzen.

In der letzten Zeile der Funktion wird nun der Hello World Text auf die Seite gesetzt. Dafür benutzen Sie das "innerHTML" Attribut vom "paragraph" Element und setzen es auf "Hello World". Nun wird alles was sich zwischen den beiden "<p>" Tags befindet ersetzt mit "Hello World".

7. Nun können Sie überprüfen ob der Code, den Sie gerade geschrieben haben auch wirklich funktioniert.

Klicken Sie dafür in der unteren rechten Ecke, in der Blauen Navigation vom Visual Studio Code Fenster auf "Go Live".



Abbildung 14 - Go Live Funktion (Quelle: Eigene Aufnahme)

8. Nun wird sich Ihr Standardbrowser öffnen und ein neuer Tab mit Ihrer HTML Seite erscheinen. Wenn Sie nun auf den Button Klicken erscheint oberhalb der Text "Hello World"

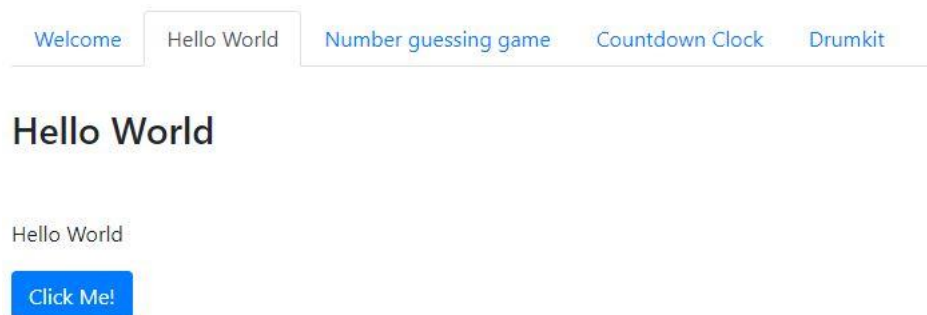


Abbildung 15 - Hello World Ergebnis (Quelle: Eigene Aufnahme)

9. Um zu überprüfen ob auch Ihr Log in der Konsole gemacht wurde, müssen Sie die Untersuchen Ansicht öffnen. Dafür machen Sie irgendwo auf der Seite einen rechts Klick und wählen im Menü den Punkt "Untersuchen" aus. Oder verwenden den Shortcut Strg + Umschalttaste + I

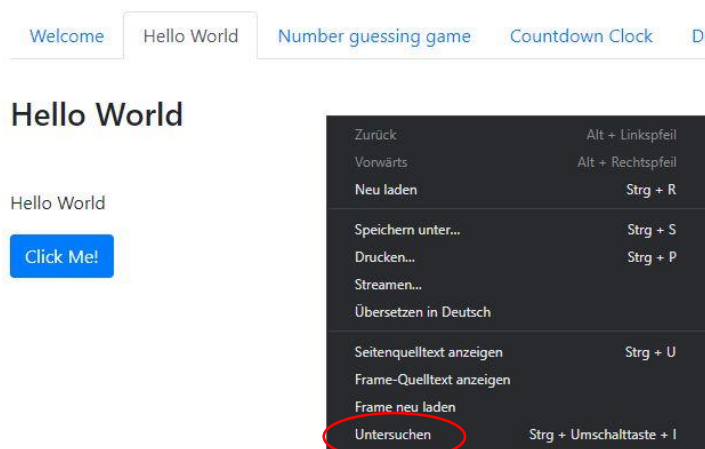


Abbildung 16 - Menü bei Rechtsklick auf Webseite (Quelle: Eigene Aufnahme)

10. Nun öffnet sich ein neues Fenster, das sogenannte "DevTools" Fenster. Falls Sie noch nicht im Console Tab sind, wechseln Sie auf den Console Tab. Nun sehen Sie in der Konsole, dass Ihr Text ausgegeben wurde.

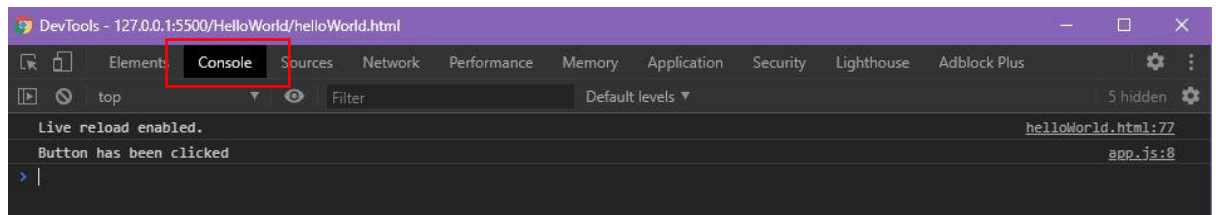


Abbildung 17 - DevTolls Fenster (Quelle: Eigene Aufnahme)

Number Guessing Game

Das Ziel dieser Aufgabe ist es, ein kleines Spiel zu programmieren. Im Spiel geht es darum, eine Zahl zwischen 1 und 100 zu erraten, die Zufällig generiert wurde. Während dem Raten bekommt der Benutzer Tipps, ob die gesuchte Zahl höher oder tiefer als die geratene Zahl ist. Ausserdem sieht man immer welche Zahlen man bereits eingegeben hat und wie viele Versuche noch übrig sind.

Link zur originalen Aufgabe: <https://jsbeginners.com/javascript-number-guessing-game/>

Math.random()

In JavaScript steht uns die Funktion Math.random() zur Verfügung, die eine zufällige Zahl zwischen 0 und 1 generiert. Da wir aber in vielen Anwendungsfällen eine Zahl in einem anderen Bereich brauchen, können wir mit dieser generierten Zahl ganz einfach eine Zahl im gewünschten Bereich berechnen

Allgemein gilt: $(\text{Math.random()} * \text{Maximum}) + \text{Minimum}$

Um aus der daraus entstandenen Dezimalzahl dann eine ganze Zahl zu machen verwendet man parseInt().

Beispiel: `let myRandom = parseInt((Math.random() * 100) + 1)`

parseInt()

Mit dieser Funktion kann man einen String zu einen Integer Parsen.

Beispiel: `let myInteger = parseInt("133");`

isNaN()

Die Funktion isNaN() (is not a number) überprüft ob der mitgegebene Wert keine Zahl ist, wenn das der Fall ist gibt sie true zurück

Beispiel: `isNaN("Hello")` → gibt true zurück

`isNaN(123)` → gibt false zurück

alert()

Mit dieser Funktion kann man eine Message für den Benutzer in einer Alert Box anzeigen. Man verwendet einfach diese Funktion und gibt die Message an den Benutzer als String mit.

Beispiel: `alert("Sie haben einen ungültigen Wert in das Textfeld eingegeben")`

Array.push()

Diese Funktion ist für Arrays. Sie fügt die mitgegebenen Werte ans Ende eines Arrays ein. Man kann beliebig viele Werte mitgeben.

Beispiel: `myArray.push(5, 88, 99)`

setAttribute() und removeAttribute()

Mit diesen beiden Funktionen kann man Attribute zu DOM Elementen hinzufügen und entfernen. Beim Setzen gibt man den Namen des Attributs an und danach den Wert und beim Löschen einfach den Attributnamen.

Beispiel: `element.setAttribute("style", "background-color: red;");`

`element.removeAttribute("style");`

Drumkit

In dieser Aufgabe geht es darum, ein Schlagzeug zu simulieren. Der Benutzer kann dann mit der Tastatur das Schlagzeug spielen.

Link zur originalen Aufgabe: <https://jsbeginners.com/drum-kit-javascript-project/>

Event-Listener keydown

Dieser Event-Listener hört auf alle Tastatur eingaben, also sobald irgendeine Taste auf der Tastatur gedrückt wird reagiert er. Wenn man diesen Event der Funktion mitgibt, hat man zugriff auf das Attribut keyCode, welches einem den Code der gedrückten Taste zurückgibt.

Beispiel:

```
eventTarget.addEventListener(keydown, event = {  
  if (event.keyCode === 229) {  
    //Do something  
  }  
});
```

Event-Listener transitionend

Dieser Event-Listener wird ausgelöst, wenn eine Transition im CSS eines Elements vorbei ist. Es gibt auch eine andere Lösung, um nach diesem Event etwas auszuführen:

```
const transition = document.querySelector('.transition');
```

```
transition.ontransitionend = () => {  
  console.log('Transition ended');  
};
```

Document.querySelector()

Die Funktion `querySelector()` von Document gibt das erste Element zurück welches dem angegebenen CSS-Selektor-String entspricht.

Beispiel: `var element = document.querySelector("div.user-panel.main input[name='login']");`

Document.querySelectorAll()

Die Funktion `querySelectorAll()` von Document gibt eine statische NodeList (Liste von Elementen aus dem DOM zurück. Auch hier gibt man ein CSS-Selektor String mit, wie bei der oberen Funktion.

`element.classList.add()` und `.remove()`

Mit dieser Funktion kann man zu den Klassen eines DOM Elements etwas hinzufügen oder entfernen. Als Parameter gibt man einfach den Namen der Klasse als String mit. Man kann auch mehrere auf einmal mitgeben, wenn man die Strings mit einem Komma trennt.

Beispiel: `div.classList.add("myClassName")`

`audio.play()`

Diese Funktion startet das Abspielen eines Audio Elements.

Wissens Checkliste

- ☐ Wissen was JavaScript ist und wo es verwendet wird
- ☐ Die Grundlagen des DOM verstehen
- ☐ Die Datentypen von JavaScript kennen
- ☐ Den Umgang mit Variablen in JavaScript verstehen und anwenden
- ☐ Ein Projekt von GitHub herunterladen können
- ☐ Wissen wie man JavaScript Files im HTML Code einbettet
- ☐ `document.getElementById()` anwenden können
- ☐ Event-Listener zu Elementen hinzufügen können
- ☐ `console.log()` verwenden können
- ☐ Die Konsole im Browser öffnen können
- ☐ `Math.random()` korrekt anwenden können
- ☐ Wissen was die Funktion `parseInt()` macht
- ☐ Wissen was die Funktion `isNaN()` macht
- ☐ Wissen wie man eine Alert Box anzeigen kann
- ☐ Die `array.push()` Methode anwenden können
- ☐ Attribute eines DOM Elements setzen und entfernen können
- ☐ Die Events `click`, `keydown` und `transitioned` kennen und anwenden können
- ☐ `document.querySelector()` anwenden können
- ☐ `document.querySelectorAll()` anwenden können
- ☐ Klassen einem Element hinzufügen und entfernen
- ☐ Die Funktion `audio.play()` kennen und anwenden können

Quellenverzeichnis

Aufgabestellungen:

- <https://jsbeginners.com/javascript-number-guessing-game/>
- <https://jsbeginners.com/javascript-countdown-clock-project/>
- <https://jsbeginners.com/drum-kit-javascript-project/>

Unterlagen Inhalt:

- <https://de.wikipedia.org/wiki/JavaScript>
- <https://wiki.selfhtml.org/wiki/DOM>
- <https://wiki.selfhtml.org/wiki/JavaScript/DOM>
- <https://developer.mozilla.org/de/docs/Web/JavaScript/Datenstrukturen>
- <https://wiki.selfhtml.org/wiki/JavaScript/DOM/Event/preventDefault>
- [https://www.w3schools.com/jsref/jsref_parseint.asp#:~:text=The%20parseInt\(\)%20function%20parses,number%20to%20a%20decimal%20number.](https://www.w3schools.com/jsref/jsref_parseint.asp#:~:text=The%20parseInt()%20function%20parses,number%20to%20a%20decimal%20number.)
- https://www.w3schools.com/jsref/met_element_setattribute.asp
- https://www.w3schools.com/jsref/met_element_removeattribute.asp
- https://developer.mozilla.org/en-US/docs/Web/API/Document/keydown_event
- https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/transitionend_event
- <https://developer.mozilla.org/de/docs/Web/API/Document/querySelector>
- <https://developer.mozilla.org/de/docs/Web/API/Document/querySelectorAll>
- <https://developer.mozilla.org/de/docs/Web/API/NodeList>
- <https://developer.mozilla.org/en-US/docs/Glossary/Node/DOM>
- <https://developer.mozilla.org/de/docs/Web/API/Element/classList>
- https://www.w3schools.com/jsref/met_audio_play.asp