

Meilenstein 2 - JavaScript

Abalo soll eine moderne dynamische Weboberfläche erhalten, damit der Komfort für die Nutzer:innen steigt. Dazu erlernen wir in diesem Meilenstein die Grundlagen von JavaScript.

Aufgabe 1 (optional)

Übung. Ziel dieser Aufgabe ist das Einüben von JavaScript anhand kurzer praktischer Beispiele. Schreiben Sie die Lösungen jeweils in eine Datei mit dem vorgegebenen Namen vor der Aufgabe. Schreiben Sie ein JavaScript-Programm, wobei Sie stets den „strict mode“ verwenden, ...

- **2-js1-hallowelt.html:** ..., das „Hallo Welt“ in einem Dialogfenster ausgibt.
- **2-js2-hallokonsole.html:** ..., das „Hallo Fehler“ als Fehler, „Hallo Warnung“ als Warnung, „Hallo Information“ als Information auf der Konsole ausgibt.
- **2-js3-funktion.html:** ..., mit einer Funktion, die zwei Werte addiert. Geben Sie das Ergebnis auf der Konsole aus.
- **2-js4-funktionalparameter.html:** ..., mit einer Funktion, die als Parameter eine Funktion erhält, die zwei Zahlen multipliziert. Rufen Sie die Funktion auf und geben Sie das Ergebnis auf der Konsole aus.
- **2-js5-array.html:** ..., das
 - a) ein Array mit den Elementen („erstes“, „zweites“, 3, 4) enthält und mit drei unterschiedlichen Schleifenkonstrukten auf der Konsole vollständig (Index und Wert) ausgibt.
 - b) Fügen Sie dem Array Elemente hinzu. Entfernen Sie das letzte Element. Verändern Sie das Element an der 2. Position. Geben Sie das Array nach jeder Operation auf der Konsole aus.
 - c) Verwenden Sie die Funktionen length, sort, find, filter, map, slice an kurzen Beispielen.
- **2-js6-kontrollstrukturen.html:** ...das kurze Beispiele der Verwendung von Kontrollstrukturen if-elseif-else, switch, for, while, do..while zeigt.
- **2-js7-operatoren.html:** ... das die Operatoren verwendet und insbesondere die Typ- und Wertvergleiche beinhaltet.
- **2-js8-strings.html:** ... das eine Zeichenkette ihrer Wahl nach Anwendung jeder der Funktionen length, indexOf, lastIndexOf, substr, replace, trim, concat, toLowerCase und toUpperCase auf der Konsole ausgibt.

Hinweis: Die Aufgabe ist optional, damit Sie diese bei vorhandenen Vorkenntnissen im Bereich JavaScript überspringen können.

Aufgabe 2

Recherche und Effizienzsteigerung. Entdecken Sie drei neue, für Sie sinnvolle Features in der von Ihnen eingesetzten IDE. Benennen und beschreiben Sie jedes Feature kurz. Beschreiben Sie, wie Sie dadurch Arbeitszeiten einsparen oder Ergebnisse in höherer Qualität erzielen.

Aufgabe 3

Browserunterstützung. In der Vorlesung haben wir erfahren, dass nicht alle Webbrowser alle Features in JavaScript unterstützen. Recherchieren Sie die Verbreitung von Webbrowsern:

- Welcher wird am meisten auf Desktops und auf mobilen Geräten eingesetzt?
- Worauf achten Sie bei der Auswahl Ihrer Quelle?
- Welcher Webbrowser unterstützt welche Version von JavaScript?
- Für welche Webbrowser würden Sie Abalo umsetzen? Nennen Sie Ihre Top 3 Webbrowser mit einer kurzen Begründung.

Aufgabe 4

Vergleich mit bekannter Programmiersprache. Sie haben die ersten Grundlagen von JavaScript kennengelernt. Sie kennen bereits mindestens eine weitere Programmiersprache wie C/C++ oder PHP (z.B. durch Ihr Studium).

Nennen Sie 6 Gemeinsamkeiten sowie 6 Unterschiede zwischen JavaScript und einer weiteren Ihnen bekannten Programmiersprache. Geben Sie kurze Beispiele.

Aufgabe 5

Übung Array Datenabfrage. Wir üben die Verwendung von JavaScript weiter ein, indem wir Funktionen schreiben, die Standardaufgaben beinahe jeder JavaScript-Anwendung durchführen. Wir lesen unterschiedliche Informationen aus JSON-formatierten Daten aus. Sie erhalten die Daten **data.js** als Datei.

Schreiben Sie JavaScript. Schreiben Sie jeweils eine Funktion ...

- getMaxPreis*, die als Parameter die Daten erhält und den Namen des ersten Produkts mit dem höchsten Preis zurückgibt.
- getMinPreisProdukt*, die als Parameter die Daten erhält und die vollständigen Produktdaten (das Objekt) des ersten Produkts mit dem niedrigsten Preis zurückgibt.
- getPreisSum*, die als Parameter die Daten erhält und die Gesamtsumme aller Produkte berechnet und zurückgibt.
- getGesamtWert*, die als Parameter die Daten erhält und die den Gesamtwert aller Produkte berechnet und zurückgibt. Der Gesamtwert eines Produkts berechnet sich aus Preis * Anzahl.
- getAnzahlProdukteOfKategorie*, die als Parameter die Daten sowie einen Kategorienamen erhält und die Gesamtanzahl aller Produkte dieser

Kategorie berechnet und zurückgibt.

Woche 2

Aufgabe 6

Übung DOM. Ziel dieser Aufgabe ist das weitere Einüben von JavaScript anhand kurzer praktischer Beispiele. Schreiben Sie die Lösungen jeweils in eine Datei mit dem vorgegebenen Namen vor der Aufgabe. Schreiben Sie ein JavaScript-Programm, wobei Sie stets den „strict mode“ verwenden, ...

- **2-js9-dom-browser.html:** ... das den Namen des verwendeten Webbrowsers („Firefox“, „Internet Explorer“, „Safari“, ...) auf der Konsole ausgibt. Verwenden Sie Stringoperationen und Kontrollstrukturen, um nur den Namen auszugeben.
- **2-js10-dom.html:** ... das die folgenden Informationen auf der Konsole ausgibt:
 - 1) den Titel des Dokuments,
 - 2) die Breite und Höhe des Darstellungsbereichs,
 - 3) die aktuelle URL des Dokuments,
 - 4) die gesetzten Cookies (Setzen Sie selbst Cookies z.B. über die Entwicklerkonsole)Verwenden Sie Quellen wie:
https://www.w3schools.com/jsref/dom_obj_document.asp für die Recherche der notwendigen Attributnamen des DOM.
- **2-js11-dom-get.html:** ... das die Funktionen `getElementBy{Id,ClassName,TagName,Name}` an einem Beispiel zeigt und die Seite über `innerHTML` und `innerText` manipuliert.
- **2-js12-dom-createElement.html:** ... das eine Form mit zwei Eingabeelementen (ein Text sowie ein Dropdown mit zwei beliebigen Optionen) in JavaScript erzeugt und die erzeugte Form einem Element auf der Seite hinzufügt.
- **2-js13-dom-overlay.html:** ... das eine Fläche mit einem Text am unteren einblendet. Recherchieren und erklären Sie dazu die CSS-Eigenschaften/Werte: „`position: absolute`“, „`bottom: 0px`“, sowie `width` und `height`.
- **2-js14-cookie.html:** ... das ein Cookie mit dem Namen „counter“ mit dem Wert 0 initialisiert. Jedes Neuladen der Seite inkrementiert den Wert von counter um eins. Der aktuelle Wert soll als Information in der Konsole ausgegeben werden.
- **2-js15-objekte.html:** ..., in dem ein Objekt auf drei unterschiedlichen Arten (JSON, Function, Class) erzeugt wird. Nennen Sie das Objekt „car“. Alle drei Objekte haben die Attribute „name“ und „type“, die beide über Getter- und Setter-Methoden gesetzt und gelesen werden können. Zudem gibt es eine Methode `writeToConsole()`, die Attribute in einem String „name=<name>, type=<type>“ als Information auf der Konsole ausgibt. Geben Sie ebenfalls die vollständigen Objekte auf der Konsole aus.

- **2-js16-class.html:** ..., in dem eine Klasse (mit dem Schlüsselwort class) „User“ mit den Attributen firstname und lastname mit passenden Getter- und Setter-Methoden implementiert ist. Setzen Sie die Attribute ebenfalls über den Constructor. Leiten Sie die Klasse „Buyer“ von User ab. Erweitern Sie Buyer um ein Attribut „lastBuy“, das den letzten Einkauf beinhaltet.
- **2-js17-click.html:** ... das bei einem Klick auf eine Schaltfläche die Hintergrundfarbe eines Elements wechselt: Rot -> Gelb -> Grün -> Rot ...
- **2-js18-mouseover.html:** ... das bei einer Mausüberfahrt die Hintergrundfarbe einer beliebigen Fläche verändert. Bei der Mauseinfahrt soll die Fläche grün und bei der Mauseinfahrt weiß werden.
- **2-js19-timeout.html:** ... das jede Sekunde die Hintergrundfarbe eines Elements alternierend Weiß und Rot schaltet.

Hinweis: Die Aufgabe ist optional, damit Sie diese bei vorhandenen Vorkenntnissen im Bereich JavaScript überspringen können.

Aufgabe 7

Navigationsmenü. Abalo soll ein Navigationsmenü für die Benutzerführung erhalten. Konstruieren Sie rein in JavaScript ein Navigationsmenü für Abalo nach dem folgenden Aufbau:

- Home
- Kategorien
- Verkaufen
- Unternehmen
 - Philosophie
 - Karriere

Alle Einträge des Navigationsmenüs sollen direkt sichtbar sein.

- a) Legen Sie für die Menüstruktur eine passende Struktur in JavaScript an. Konstruieren Sie auf Basis Ihres Arrays das Menü in HTML. Nutzen Sie verschachtelte , -Elemente, die Sie mit CSS stylen.
- b) (optional) Stellen Sie die in der Datenbank gespeicherten Artikelkategorien (ab_articlecategory) unter dem Menüpunkt „Kategorien“ dar.
- c) (optional) Untermenüs, wie die Punkte „Philosophie“ und „Karriere“ unter „Unternehmen“, sollen nicht direkt sichtbar sein, sondern erst durch einen Klick sichtbar werden. Finden Sie eine komfortable Umsetzung für Nutzer:innen.

Aufgabe 8

Cookie-Genehmigung. Seit Mai 2018 gilt die EU DSGVO (EU-Datenschutzgrundverordnung). Die DSGVO verlangt, dass bei Verwendung von Cookies Nutzer:innen über die Verwendung selbst sowie über die geltende Rechtsgrundlage unterrichtet werden. Ein Beispiel finden Sie unter <https://www.fh-aachen.de>.

Erweitern Sie Abalo um eine einfache Rückfrage, ob Nutzer:innen mit der Verwendung von Cookies einverstanden ist. Ist der/die Nutzer:in einverstanden, so erscheint die Abfrage nicht wieder. Die Abfrage erscheint auch nach einem Neuladen der Seite nicht wieder. Entwickeln Sie dazu selbst eine Lösung in JavaScript.

Legen Sie die JavaScript-Datei in Laravel unter `/public/js/cookiecheck.js` ab und inkludieren Sie die Datei aus der View heraus.

Aufgabe 9

Artikeleingabe. Erweitern Sie Abalo. Implementieren Sie in JavaScript eine Artikeleingabe unter `/newarticle`. Konstruieren Sie die Form rein in JavaScript. Erzeugen Sie den notwendigen HTML-Code also mit JavaScript.

Überprüfen Sie die Dateneingabe auf Richtigkeit (`price > 0`, `name` nicht leer) ebenfalls in JavaScript. Schicken Sie die Artikeldaten über die Parameter `name`, `price` und `description` zum Endpunkt `/articles` mit POST. Verwenden Sie für das Abschicken ein: `document.getElementById("...").submit()`. Reagieren Sie dafür den Klick auf die Schaltfläche „Speichern“. Serverseitig sollen die Daten nochmals geprüft und in der Datenbank persistiert werden. Schlägt das Einfügen serverseitig fehl, soll für den/die Benutzer:in eine sprechende serverseitig erzeugte Fehlermeldung dargestellt werden.

Aufgabe 10

Warenkorb. Erweitern Sie Abalo um einen clientseitigen Warenkorb, in dem Artikel zusammengestellt werden können. Es handelt sich um eine reine clientseitige Implementierung (ohne den Einsatz von AJAX). D.h. beim Verlassen der Seite sind die Daten im Warenkorb verloren. Später erweitern wir den Warenkorb um die serverseitige Funktionalität.

Erweitern Sie dazu die Artikelübersicht unter `/articles`. Fügen Sie eine Darstellung des Warenkorbs oberhalb der Artikel in die Webseite ein. Jeden Artikel gibt es in Abalo nur einmal.

- Der/die Nutzer:in kann einen Artikel über ein + (Plus) dem Warenkorb hinzufügen. Der Warenkorb soll sofort, ohne Serveranfrage, aktualisiert sein.
- Finden Sie eine Lösung dafür, dass ein Artikel nicht mehrfach im Warenkorb vorkommt.
Beachten Sie, dass keine Anzahl von Artikeln benötigt wird, da in Nutzer:innen in Abalo in der Regel Einzelstücke anbieten.
- Jeder Artikel im Warenkorb soll über ein – (Minus) entfernt werden können. Der Artikel soll sofort, ohne Serveranfrage, entfernt und in der Artikelliste wieder verfügbar sein.

Aufgabe

Abgabe. Laden Sie Ihre Ergebnisse als ZIP in ILIAS hoch. Der Name des ZIPs soll sein:

<TeamNr>_<Meilensteinnummer>.zip

Die TeamNr finden Sie in der Teamzuordnung. TeamNr und Meilensteinnummer schreiben Sie bitte ohne führende Null. Also z.B. 4_2.zip für Team 4 Meilenstein 2.

Das ZIP soll alle Ergebnisse des Meilensteins (Antworten zu Freitextaufgaben, Grafiken, Quelltexte, ...) enthalten. Fügen Sie zusätzlich ein README ein, wo Sie Vor-, Nachname und Matrikelnummer der beteiligten Teammitglieder hinterlegen.