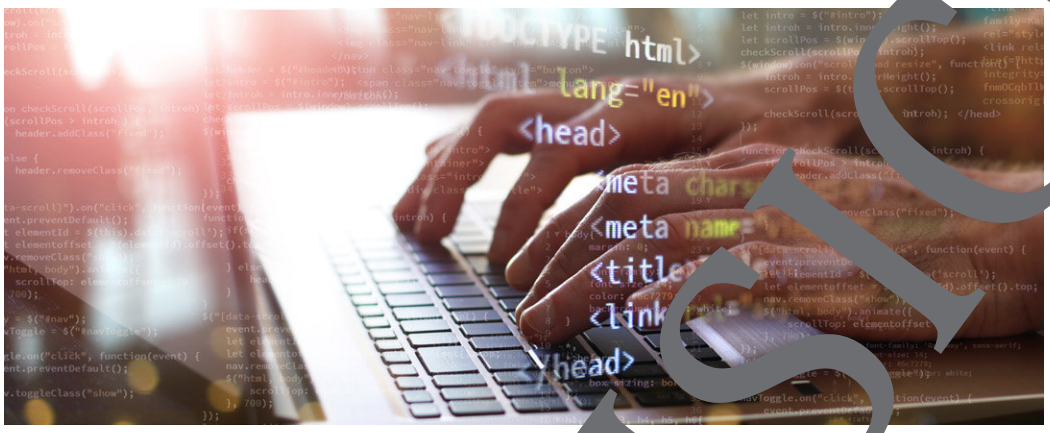
B.IV.9

Algorithmen – Objektorientierte Programmierung

Einheit: JavaScript-Grundlagen in HTML – Teil 2

Mirko Schiller



Um Besucherinnen und Besuchern einer Webseite ein „Gefällt mir“ oder „Teilen“ zu bieten, müssen Webentwicklerinnen und Webentwickler J... In unserer zweiten Einheit zu JavaScript liegen die Schwerpunkte auf Algorithmen und Schleifen sowie als Vertiefung auf Callbacks. Ihre Klasse erlernt den vertieften Umgang mit JavaScript anhand von zahlreichen Übungen.

KOMPETENZPROFIL

Klassenstufe:	9/10, Sek II
Dauer:	...errichtsstunden
Lernziele:	Die Lernenden ... 1. entwickeln auf Grundlage von JavaScript ein Verständnis für dynamische Benutzerinteraktionen, 2. implementieren algorithmische Grundstrukturen für Webseiten, 3. übertragen deren Kenntnisse auf andere Entscheidungsprozesse.
Thematische Bereiche:	Internet, HTML, JavaScript, Webentwicklung, interaktives Verhalten von Webseiten, Algorithmen, Kontrollstrukturen
Kompetenzbereiche:	Implementieren, Darstellen und Interpretieren, Produzieren und Präsentieren, Analysieren und Reflektieren

Fachliche Hinweise

Was sollten Sie zum Thema wissen?

JavaScript bietet unzählige Anwendungsszenarien, hat aber stellenweise hohe Anfangsh[ürden] und Fallstricke. Grundlagen und typische Einstiegsbeispiele werden in dieser Einheit gezeigt, sodass anfängliche Herausforderungen minimiert werden. Weiterführende und ergänzende Informationen findet man vor allem auf SELFHTML, w3school und Mozilla Developers.

Welches Vorwissen sollten die Lernenden mitbringen?

Es ist ratsam, dass Grundlagen von HTML bereits vorhanden sind. Speziell die Funktionsweise von HTML-Tags und der Aufbau einer HTML-Grundstruktur sollten vorhanden sein (siehe dazu A.I.15). Notwendig ist, dass die Lernenden Dateitypen kennen und diese mit den dazugehörigen Anwendungsumgebungen anwenden (z. B. HTML-Datei mit Browser öffnen).

Wie kann die Erarbeitung des Themas im Unterricht erfolgen?

Vorbereitung

- Stellen Sie ausreichend Laptops/PCs/mobile Endgeräte im Klassenzimmer zur Verfügung, idealerweise ein Gerät pro Schüler/-in oder mindestens ein Gerät pro Schülerpaar.
- Sorgen Sie für die Bereitstellung von Internet im Klassenzimmer.
- Stellen Sie einen alternativen Text-Editor zur Verfügung (vorzugsweise Notepad++)
- Sorgen Sie für Notizmöglichkeiten bei Partner-/Gruppenarbeit (digitales Endgerät, Blätter, farbige Stifte).

Einstieg/Erarbeitung/Übung, etc.

Die Materialien sind so erstellt, dass diese sowohl im Unterricht mit der Lehrkraft zusammen oder von jeder Schülerin bzw. jedem Schüler einzeln erarbeitet werden können. Jedes Material einzeln betrachtet hält in der Regel einführende, erarbeitende oder übende Elemente bereit. Diese werden daher in Lektionen zusammengefasst. Lektionen können also flexibel nach den jeweiligen Bedürfnissen eingesetzt werden. Es wird empfohlen, dass die unterrichtende Person zunächst das Material selbst in der Vorbereitung sichtet und auf dieser Grundlage entscheidet, welches Material eher in Einzel-/Partnerarbeit oder in Form eines instruierenden/demonstrierenden Unterrichtsstils vermittelt wird.

Weiterführende Medien

- SELFHTML e. V.: JavaScript-Wiki: https://wiki.selfhtml.org/wiki/JavaScript [Letzter Abruf aller Links am 15.07.2024]
- Mozilla Developer Docs (englisch): https://developer.mozilla.org/en-US/docs/Web/JavaScript [Letzter Abruf aller Links am 15.07.2024]
- W3School: JavaScript Tutorial (englisch): https://www.w3schools.com/js/ [Letzter Abruf aller Links am 15.07.2024]

Auf einen Blick

Benötigte Materialien

☐ PC/Laptop/mobiles Endgerät
☐ Stifte/Notizzettel

Lektion 6

Thema:	Algorithmen – Teil 1 (90 Min.)
M 8	**Algorithmen in JavaScript**
M 9	**Falls-Dann-Sonst**
Benötigt:	☐ M9.2.Karls-Kino-vorlage.zip ☐ M9.3.zahlenraten-vorlage.zip

Lektion 7

Thema:	Algorithmen – Teil 2 (90 Min.)
M 10	**Schleifen**
M 11	**Übung: Kontrollstrukturen**

Lektion 8

Thema:	Spezifika in Ja[illegible]t (45 Min.)
M 12	**Vertiefung: Callb[illegible]ks in Jav[illegible]**

Benötigte Dateien

- **Programmcode:** M9.2 Karls-Ki[illegible]vorlage.zip
- **Programmcode:** M[illegible]zahlenrat[illegible]n-vorlage.zip
- **Link:** Sammlung aller [illegible]ram[illegible]code-[illegible]gen und Lösungen – https://editor.p5js.org/mi[illegible]iller/collections/h_oQOYblN

Ergänzendes Material

- ☐ **Programmcode:** M9.4.muenzwurf1-lsg.html
- ☐ **Programmcode:** M10.1.countdown-lsg.html
- ☐ **Programmcode:** M10.2.muenzwurf2-lsg.html
- ☐ **Programmcode:** M10.3.muenzwurf3-while-lsg.html
- ☐ **Programmcode:** M10.4.tresor1-lsg.zip
- ☐ **Programmcode:** M10.5.schatzsuche-lsg.html
- ☐ **Programmcode:** M11.2.tresor2-lsg.zip
- ☐ **Programmcode:** M11.3.altersabfrage-alternative-lsg.zip
- ☐ **Programmcode:** M11.3.altersabfrage-lsg.zip
- ☐ **Programmcode:** M11.4.formularabfrage-Zeichen-lsg.zip
- ☐ **Programmcode:** M11.5.formularabfrage-senden-lsg.zip
- ☐ **Programmcode:** M11.6.quiz-lsg.html

Erklärung zu den Symbolen

 Dieses Symbol markiert differenzi[illegible]tes Materia[illegible]n nicht anders ausgewiesen, befinden sich die Materialien auf mi[illegible]Niveau.		
 leichtes Niveau	 mi[illegible]res Niveau	 schwieriges Niveau
 Zusatzaufgabe	 Alternative	 Selbsteinschätzung

Algorithmen in JavaScript

M 8

Puzzeln

Angenommen, du hast ein riesiges Puzzle vor dir liegen. Um dieses Puzzle zu lösen, benötigst du eine klare Schritt-für-Schritt-Anleitung, die dir sagt, welche Teile zuerst gelegt werden sollten, welche Teile zusammenpassen und welche Schritte du wiederholen musst, bis das gesamte Bild vollständig ist. Diese systematische Anleitung zum Lösen des Puzzles ist dem, was wir in der Informatik einen Algorithmus nennen, sehr ähnlich. Ein Algorithmus ist wie ein Rezept: eine klare Liste von Anweisungen, die dir hilft, ein Problem oder eine Aufgabe zu lösen.

Was ist ein Algorithmus?
Ein Algorithmus ist eine Reihe von klaren Anweisungen, die ausgeführt werden, um eine bestimmte Aufgabe oder ein Problem nach endlich vielen Schritten zu lösen. In der Programmierung helfen uns Algorithmen, Computer dazu zu bringen, Aufgaben in einer logischen und effizienten Reihenfolge zu erledigen.

Algorithmen und JavaScript

JavaScript, die Sprache, mit der wir Webseiten zum Leben erwecken, bietet uns Werkzeuge zur Erstellung von Algorithmen. Einige dieser Werkzeuge sind:

- *Entscheidungsstrukturen*: Mit `if-else` können wir dem Computer sagen: „Wenn dies zutrifft, mache das, sonst mache jenes."
- *Wiederholungsstrukturen*: Mit Schleifen wie `for` oder `while` können wir Anweisungen mehrmals wiederholen, ohne sie immer wieder neu schreiben zu müssen.

Hierbei sollten Algorithmen diverse Kriterien erfüllen:

Kriterium	Beschreibung
Eindeutigkeit	Jeder Schritt des Algorithmus muss klar und unmissverständlich formuliert sein. Es sollte keinen Raum für mehrere Interpretationen eines Schritts geben.
Vollständigkeit	Der Algorithmus muss für jede mögliche Eingabe eine Antwort oder eine Lösung liefern. Er darf keine Eingabe auslassen.
Determiniertheit	Bei gleicher Eingabe muss der Algorithmus immer das gleiche Ergebnis produzieren, unabhängig davon, wie oft er ausgeführt wird.
Endlichkeit	Der Algorithmus muss nach einer endlichen Anzahl von Schritten zu einem Ergebnis kommen. Das bedeutet, er darf nicht in einer unendlichen Schleife enden.
Allgemeingültigkeit	Ein Algorithmus muss in der Lage sein, alle Probleme der gleichen Problemklasse zu lösen, nicht nur ein spezifisches Beispiel des Problems.
Effizienz (optional, aber oft erwünscht)	Ein guter Algorithmus sollte in der Lage sein, ein Problem in einer effizienten und zeitnahen Weise zu lösen, besonders bei großen Datenmengen.
Korrektheit	Der Algorithmus sollte bei jeder Eingabe das korrekte und erwartete Ergebnis liefern.

Aufgabe: Alltag trifft Programmierung

Stell dir vor, du musst morgens für die Schule aufstehen und dich fertig machen. Dieser alltägliche Prozess kann als Algorithmus dargestellt werden:

1. Morgendliche Routine: **Schreibe** alle Schritte auf, die du unternimmst, vom Aufwachen bis zum Verlassen des Hauses für die Schule. Versuche, so detailliert wie möglich zu sein.
2. Programmier-Perspektive: **Überlege**, welche dieser Schritte Entscheidungen beinhalten. Zum Beispiel: „Wenn es regnet, dann nehme ich einen Regenschirm mit." Formuliere diese Entscheidungen in einer „Wenn-Dann"-Struktur.
3. Wiederholungen erkennen: Gibt es Schritte, die du regelmäßig wiederholst? Zum Beispiel: „Putze jeden Zahn für 10 Sekunden." Dies könnte als Schleife interpretiert werden. **Notiere** alle wiederkehrenden Aktionen und überlege, wie oft sie durchgeführt werden.
4. Zum Nachdenken: Wie würdest du deine morgendliche Routine optimieren, um Zeit zu sparen? Welche Schritte könnten effizienter gestaltet oder sogar übersprungen werden?

Vom einfachen Algorithmus zu maschinellen Entscheidungsprozessen

Wenn wir von Algorithmen in einem einfachen Kontext wie unserem morgendlichen Aufstehen sprechen, denken wir an eine strukturierte Abfolge von Schritten. Wenden wir dieses Konzept jedoch auf maschinelle Prozesse erweitern, vor allem auf jene, die sich mit komplexen Entscheidungen auseinandersetzen, wird die Tiefe und Komplexität der Algorithmen erheblich größer.

- **Autonomes Fahren:** Ein autonomes Fahrzeug nutzt komplexe Algorithmen, um Entscheidungen in Echtzeit zu treffen. Es muss kontinuierlich Daten von seinen Sensoren auswerten, um die Umgebung zu interpretieren. Wenn zum Beispiel ein Fußgänger die Straße betritt, muss das Auto diesen erkennen und entsprechend reagieren. Diese Entscheidung basiert nicht nur auf einem einfachen Wenn-Dann, sondern auf einer tiefen Analyse vieler Datenpunkte und möglicher Szenarien.
- **Wahlcomputer:** In einigen Ländern werden Wahlcomputer eingesetzt, um Wählerstimmen elektronisch zu erfassen und auszuzählen. Diese Maschinen verwenden Algorithmen, die darauf ausgelegt sind, die Wahlentscheidungen korrekt und sicher zu registrieren. Dabei gilt es, Manipulationsversuche zu erkennen und abzuwehren. Ein Algorithmus in einem Wahlcomputer muss in der Lage sein, unklare oder mehrdeutige Eingaben zu erkennen und entsprechend zu flaggen, sodass sie manuell geprüft werden können.
- **Gesichtserkennung:** Gesichtserkennungsalgorithmen sind in der Regel komplex und nutzen neuronale Netzwerke, um ein menschliches Gesicht in einem Bild oder Video zu erkennen und es mit einer Datenbank von Gesichtern abzugleichen. Diese Algorithmen zerlegen das Gesicht in Tausende Datenpunkte, die bestimmte Merkmale oder Eigenschaften repräsentieren. Selbst Veränderungen wie Beleuchtung oder Alter muss der Algorithmus in der Lage sein, das Gesicht zu identifizieren. Solche Technologien haben vielfältige Anwendungen, von der Sicherheit über die Smartphone-Entsperrung bis hin zu personalisierter Werbung.

In allen diesen Beispielen müssen die Algorithmen nicht nur eine Aufgabe erfüllen, sondern auch robust und sicher gegen Fehler oder Manipulationen sein. Im Kern mag das Prinzip dasselbe sein – eine Folge von Anweisungen – aber die Komplexität, Tiefe und die Anforderungen an die Genauigkeit sind in solchen maschinellen Entscheidungsprozessen enorm gesteigert.

Das Wissen über einfache Algorithmen hilft uns, die Grundlagen zu verstehen. Aber wenn wir die Technologien der Zukunft entwickeln und verbessern wollen, müssen wir die Fähigkeiten und Kenntnisse zur Gestaltung und Analyse von Algorithmen weiter vertiefen.

RAAbits Online
@RAABE

Mehr Materialien für Ihren Unterricht mit RAAbits Online

Unterricht abwechslungsreicher, aktueller sowie nach Lehrplan gestalten – und dabei Zeit sparen.
Fertig ausgearbeitet für über 20 verschiedene Fächer, von der Grundschule bis zum Abitur: Mit RAAbits Online stehen redaktionell geprüfte, hochwertige Materialien zur Verfügung, die sofort einsetz- und editierbar sind.

- Zugriff auf bis zu **400 Unterrichtseinheiten** pro Fach
- Didaktisch-methodisch und **fachlich geprüfte Unterrichtseinheiten**
- Materialien als **PDF oder Word** herunterladen und individuell anpassen
- Interaktive und multimediale Lerneinheiten
- Fortlaufend **neues Material** zu aktuellen Themen

Testen Sie RAAbits Online 14 Tage lang kostenlos!

www.raabits.de

RAAbits Online – Das Portal für Unterrichtsmaterialien und Schulorganisation