

Analyse der didaktischen Auswirkungen von
Klassenteilungen auf die Unterrichtsplanung bei einer
Jahrgangsstufe 2 des Technischen Gymnasiums mit
Profilfach Technik und Management

Dokumentation des Forschungsprojekts in angewandter Fachdidaktik Informatik

Vorgelegt von:

Lorenz Bung

Datum:

20. Mai 2024

Semester / Matr.-Nr.:

SS24 / 5113060

Betreuende Lehrkraft:

OStR Benjamin Lempp

Ausbildungsschule:

RFGS Freiburg

Versicherung

Hiermit versichere ich, dass ich diese Dokumentation des Forschungsprojekts selbstständig und nur mit den angegebenen Hilfsmitteln angefertigt habe. Ich habe für alle Stellen und Materialien, die dem Wortlaut oder dem Sinn nach aus anderen Werken (auch aus elektronischen Medien) entnommen wurden, die Quellen angegeben.

(Ort, Datum)

(eigenhändige Unterschrift)

Inhaltsverzeichnis

1	Einleitung	1
2	Konzeption	2
2.1	Voraussetzungsanalyse	2
2.1.1	Anthropogene Voraussetzungen	3
2.1.2	Sozio-kulturelle Voraussetzungen	4
2.2	Lerninhalte und -ziele	5
2.2.1	Gliederung des Lernstoffs	6
2.2.2	Formulierung der Lernziele	7
2.3	Methodik und Medien	10
3	Umsetzung	12
3.1	Erste Doppelstunde	12
3.2	Zweite Doppelstunde	14
3.3	Dritte Doppelstunde	16
3.4	Vierte Doppelstunde	18
4	Reflexion	19
	Anhang	I
A	Unterrichtsverlaufspläne	I
B	Merkblätter	VIII

C Übungsblätter	XIII
D Codegerüste zu den Übungsaufgaben	XVIII
D.1 Aufgabe 0: Wiederholung HTML	XVIII
D.2 Aufgabe 1: Personenbeschreibung	XIX
D.3 Aufgabe 2: Promillerechner	XX
D.4 Aufgabe 3: Bildergalerie	XXI
D.5 Aufgabe 4: Song Shuffle	XXI
D.6 Aufgabe 5: To-Do-Liste	XXII
D.7 Aufgabe 6: Taschenrechner 1	XXIII
D.8 Aufgabe 7: Klickzähler	XXIII
D.9 Aufgabe 8: Taschenrechner 2	XXIV
E Ausdrücke der ersten Seiten der Onlineliteratur	XXV

1 Einleitung

2 Konzeption

2.1 Voraussetzungsanalyse

Die in dieser Arbeit durchgeführte Voraussetzungsanalyse orientiert sich am von Paul Heimann begründeten Berliner Modell [1, S. 41–70]. Wie in [Abbildung 1](#) zu

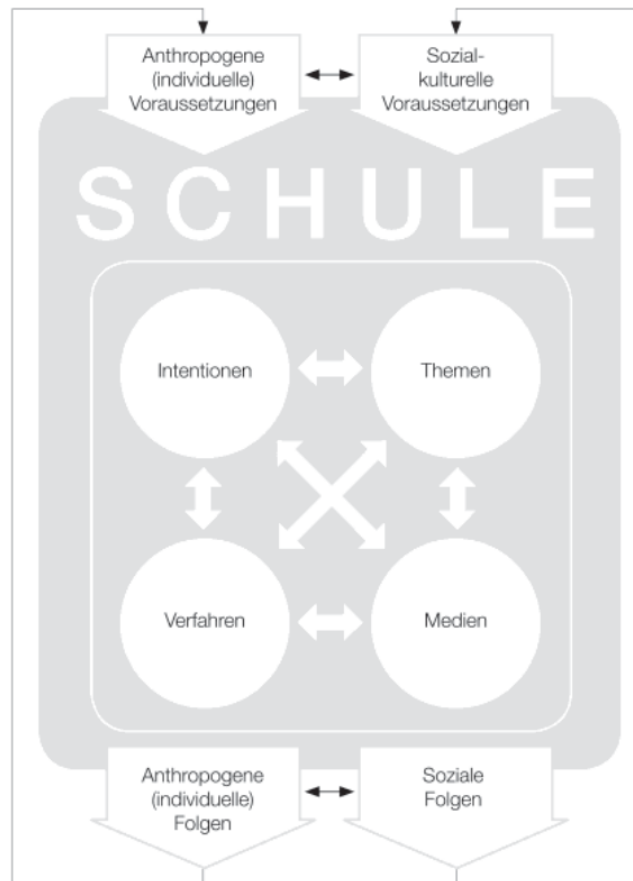


Abbildung 1: Schematische Darstellung des Berliner Modells [1, S. 68]

sehen ist, spielen nach diesem Modell sowohl die anthropogenen als auch die sozio-kulturellen Voraussetzungen der Lerngruppe eine entscheidende Rolle. Im Folgenden wird daher auf diese beiden Bedingungen genauer eingegangen, mögliche Auswirkungen auf die Lerngruppe genannt und daraus resultierende Konsequenzen auf die Planung des Unterrichts diskutiert.

2.1.1 Anthropogene Voraussetzungen

Bei der Analyse der anthropogenen Voraussetzungen der Lerngruppe werden Faktoren berücksichtigt, welche die Schülerinnen und Schüler in die Unterrichtssituation mitbringen. Fragen, die diese Analyse zu beantworten sucht, können unter Anderem die folgenden sein:

- Welches Vorwissen bringt die Lerngruppe zum Thema der Unterrichtsstunde mit?
- Wie sind die sozialen Vorbedingungen, die innerhalb der Klasse herrschen?
- Wie sind Selbstwahrnehmung und Interesse der Schülerinnen und Schüler in Bezug auf das Fach, insbesondere zum aktuellen Thema?

Die im Rahmen dieser Arbeit durchgeführte Unterrichtseinheit fand in der zweiten Jahrgangsstufe eines Technischen Gymnasiums mit dem Profulfach "Technik und Management" statt. Viele der Schülerinnen und Schüler haben zuvor eine Realschule besucht und dort die Mittlere Reife erhalten. Das technische Gymnasium bietet ihnen somit die Möglichkeit, statt einer Ausbildung einen höheren Bildungsabschluss zu erlangen. Für viele der Schülerinnen und Schüler ist die Perspektive nach dem Abitur ein Studium. Unter Anderem aus diesem Grund brachten die meisten Lernenden eine vergleichsweise hohe Eigenmotivation mit.

Die Wahl der Schule und des Profulfaches spricht dafür, dass der Großteil der Lerngruppe ein hohes Interesse an technisch-mathematischen Themenbereichen mitbringt. In Hospitationen im Unterricht mit der Klasse hat sich durch Gespräche und Beobachtungen jedoch herauskristallisiert, dass Informatik und informatische Inhalte von Teilen der Klasse als langweilig oder irrelevant empfunden wird. Andere Schülerinnen und Schüler brachten bereits Einiges an Vorwissen mit und hatten auch Spaß an der Auseinandersetzung mit Problemstellungen aus dem Informatikunterricht.

Der Grund für die Wahl des Themas dieser Forschungsarbeit war, dass die Klasse für den Unterricht in zwei etwa gleich große Gruppen geteilt wurde. Das Klassenklima in der Klasse lässt sich als durchgängig gut beschreiben, allerdings ist hierbei zu berücksichtigen, dass die Sicht der Lehrkraft auf das Klassenklima selbstverständlich nicht immer exakt der realen Situation entspricht. Auffällig an vielen Klassen am Technischen Gymnasium – insbesondere aber auch bei der in dieser Arbeit betrachteten Lerngruppe – ist das ungleiche Geschlechterverhältnis. In

diesem konkreten Fall waren im Unterricht nur zwei Schülerinnen, während der Rest der Lerngruppe männlich war. Aufgrund möglicher geschlechtsspezifischer Stereotypen, die insbesondere im Fach Informatik leider immer noch sehr prävalent sind, ist dieses ungleiche Geschlechterverhältnis ein Punkt, auf den bei der Durchführung des Unterrichts und der eventuell angepassten Konzeption der darauffolgenden Stunde ein besonderes Augenmerk gelegt werden muss.

Ein großes Problem vor der Durchführung der Unterrichtseinheit war die fehlende Klarheit über das Vorwissen der Schülerinnen und Schüler zum Thema "Webentwicklung". Auch die betreuende Lehrkraft konnte diesbezüglich keine Angaben machen, da diese die Klasse erst in der Jahrgangsstufe übernommen hatte, nicht jedoch in der Eingangsklasse. Eben dann steht allerdings das Thema "HTML und CSS" im Bildungsplan [6, BPE 2], was essentielles Vorwissen für die kommende Unterrichtseinheit ist. Diese Frage konnte bis zur ersten Stunde auch nicht mehr geklärt werden, was die Planung dieser ersten Stunde erheblich erschwerte. Klar war aber, dass alle Schülerinnen und Schüler Erfahrung im Bereich der Programmierung mit Python mitbrachten. Dies war das Thema, mit dem sich die Klasse bis dahin beschäftigt hatte; es lag also auch noch keine große zeitliche Distanz dazwischen.

Insgesamt kann festgehalten werden, dass die Lerngruppe in Bezug auf ihr Vorwissen zwar relativ homogen war, bezüglich der Motivation für das Fach und seine Lerninhalte jedoch große Heterogenität herrschte. Dies spiegelte sich auch deutlich in der Lerngeschwindigkeit der verschiedenen Schülerinnen und Schüler wider und war bei der weiteren Planung der Unterrichtseinheit zu berücksichtigen.

2.1.2 Sozio-kulturelle Voraussetzungen

Hinter der Frage nach den sozio-kulturellen Voraussetzungen stehen sowohl externe wie auch innere Faktoren, die das Unterrichtsgeschehen beeinflussen können. Ein Teil der dazu zählenden Punkte sind:

- Wie viel Zeit steht für den Unterricht zur Verfügung?
- Handelt es sich um eine Einzel- oder eine Doppelstunde?
- In welchen Räumlichkeiten findet die Stunde statt?
- Genügt die vorhandene technische Ausstattung den Anforderungen des geplanten Unterrichts?

Die Räumlichkeiten der Richard-Fehrenbach-Gewerbeschule, in der der im Rahmen dieser Arbeit durchgeführte Unterricht stattgefunden hat, sind durchgängig als technisch gut bis sogar hervorragend ausgestattet einzustufen. Sämtliche gehaltenen Stunden konnten in Computerräumen von ausreichender Größe durchgeführt werden, so dass jede Person an einem eigenen Computer arbeiten konnte. Zudem war in jedem Raum eine digitale Tafel mit Dokumentenkamera und drahtloser Verbindung zu digitalen Endgeräten vorhanden. Weiterhin war in jedem Computerraum ein Drucker platziert, so dass bei Bedarf Übungs- und Merkblätter in gedruckter Form an die Schülerinnen und Schüler weitergegeben werden konnten.

Alle Stunden der Unterrichtseinheit waren Doppelstunden, also in ununterbrochener Länge von 90 Minuten. Geplant waren für das gesamte Thema 8 Stunden bzw. 4 Doppelstunden, was sich mit den Vorgaben im Bildungsplan deckt [6, BPE 19]. Bei Bedarf wäre es allerdings möglich gewesen, dies um eine weitere Doppelstunde zu erweitern. Insgesamt wurden 32 Stunden á 45 Minuten gehalten, wobei jede Stunde einmal pro Gruppe stattgefunden hat. Je Gruppe fanden also 16 Stunden á 45 Minuten, jeweils in Blöcken von 90 Minuten statt.

2.2 Lerninhalte und -ziele

Vor der Aufstellung konkreter Lernziele ist es notwendig, sich klar zu machen, welche Lerninhalte im Unterricht eigentlich vermittelt werden sollen. Für die Planung einer vollständigen Bildungsplaneinheit, wie es in der vorliegenden Arbeit stattgefunden hat, muss dafür zunächst eine Aufteilung der Lernstoffes in einzelne Abschnitte durchgeführt werden, welche dann wiederum die inhaltliche Grundlage für die einzelnen Stunden bieten. Es ist also nötig, das Thema der Einheit in möglichst unabhängig voneinander stehende Abschnitte zu gliedern, welche aber gleichzeitig im Gesamtbild miteinander in Wechselwirkung stehen und ein umfassendes Bild der Thematik liefern.

Dass diese Trennung eine große Herausforderung darstellt, ist wohl selbstverständlich. Viel bedeutsamer ist für diese Arbeit jedoch, dass es nicht die eine unanfechtbar korrekte Lösung für dieses Problem gibt, sondern dass viele verschiedene Aufteilungen des Lernstoffes möglich sind. Je nach Lerngruppe, Vorwissen oder insbesondere auch nach verfolgtem Ziel haben diese verschiedenen Herangehensweisen unterschiedliche Vor- und Nachteile, über die man sich bei

der Planung des Unterrichts im Klaren sein muss.

2.2.1 Gliederung des Lernstoffs

Bei der Planung der in dieser Arbeit betrachteten Bildungsplaneinheit zum Thema "Javascript" gab es dabei zunächst zwei Ansätze, den Lernstoff zu gliedern. Der erste Ansatz war, analog zu bereits erfolgten Stunden zum Thema "Python" vorzugehen und Javascript einfach von Grund auf als neue Programmiersprache einzuführen. Dieser Ansatz hatte den Vorteil, dass das Vorgehen bereits in der Unterrichtseinheit zu Python erfolgreich durchgeführt worden war und man mit dieser Methode ein relativ sicheres Lernergebnis erreichen können würde. Der große Nachteil war jedoch, dass eine erneute Einführung einer neuen Programmiersprache nach demselben Schema und mit ähnlichen oder sogar gleichen Übungsaufgaben für die Schülerinnen und Schüler sehr langweilig gewesen wäre. Hier hätte sich schnell die Sinnfrage gestellt: "Wir haben das doch schon alles in Python gemacht, warum sollen wir das erneut in Javascript machen?" Sicherlich wäre diese Frage zumindest teilweise gerechtfertigt gewesen; außerdem wäre der Kernunterschied von Javascript zu Python nicht klar geworden: Nämlich, dass es sich bei Javascript um eine im Browser clientseitig ausgeführte Sprache handelt.

Im Gegensatz dazu stand der zweite Ansatz, der darauf aufbaute, direkt von Anfang an zu zeigen, dass das Ziel der Einheit die Webentwicklung ist. Hier würden Aufgaben so gestellt werden, dass die Schülerinnen und Schüler bereits vorgefertigte Webseiten inklusive Codegerüsten bekommen würden und diese entsprechend den Aufgabenstellungen vervollständigen müssten. Die Einführung bekannter Programmierkonzepte wie Verzweigungen, Schleifen und Funktionen würde nicht als abstraktes Konzept, sondern direkt anhand von Beispielen eingeführt werden. Ein entscheidender Vorteil wäre hierbei, dass sich die erlernten Konzepte direkt auf reale in der Webentwicklung auftretende Situationen anwenden ließe und sich so direkt erschließen würde, wofür das ganze benötigt wird. Dies würde im Optimalfall zu erhöhter Motivation bei den Schülerinnen und Schülern führen. Der entscheidende Nachteil dieser Variante wäre jedoch, dass die Struktur des Unterrichts weniger klar wäre. Da sich die Einführung neuer Konzepte an Beispielp Problemen wie vorgefertigten Webseiten orientieren würde, müssten viele Teile des Codes bereits fertig vorliegen. Dies könnte die Schülerinnen und Schüler ablenken und im schlechtesten Fall so verwirren, dass die eigentlich einfachen Aufgaben nicht mehr gelöst werden könnten. Andererseits böte sich hier

auch die Chance, interessierten Schülerinnen und Schülern einen Ausblick auf fortgeschrittenere Inhalte zu bieten.

2.2.2 Formulierung der Lernziele

Die in [Abschnitt 2.2.1](#) diskutierten Ansätze wurden mit der betreuenden Lehrkraft ausgiebig besprochen und die Vor- und Nachteile gegeneinander abgewägt. Schlussendlich wurde für die Planung der Unterrichtseinheit der zweite, problemorientierte Ansatz gewählt. Einer der ausschlaggebenden Gründe dafür war der motivationsaufbauende Aspekt von problemorientiertem Unterricht, insbesondere vor dem Hintergrund, dass die Programmiersprache Python erst kürzlich formal eingeführt worden war. Weiterhin bot die natürliche Komplexität der Aufgaben eine gute Möglichkeit, bereits erste Maßnahmen zur Differenzierung zu nehmen und durch zusätzliche bzw. weiterführende Inhalte auch besonders schnellen oder interessierten Schülerinnen und Schülern gerecht zu werden.

Ein Teil der im Bildungsplan zum Thema Javascript verankerten Lerninhalte [6, BPE 19] sind die folgenden:

- Deklaration von Variablen und Konstanten
- Einfache arithmetische Vergleichsoperationen
- Kontrollstrukturen: Verzweigungen, Schleifen

Eine Aufteilung dieser Inhalte auf die vier Doppelstunden lieferte die in [Tabelle 1](#) dargestellte Struktur. Wie sich im Verlauf der verschiedenen Stunden zeigen sollte, war diese Planung nicht final und musste im Laufe der Zeit noch weiteren Änderungen unterzogen werden. Weiteres hierzu wird in **?? TODO** genauer diskutiert.

Stunde	Inhalt
1	Überprüfung des Vorwissens zu HTML/CSS Variablen if-Verzweigung
2	Schleifen: for/while
3	Arrays, Funktionen
4	Entwicklung eines kleinen Spiels auf Basis der erlernten Konzepte

Tabelle 1: Aufteilung der Lerninhalte auf die Unterrichtsstunden vor der Durchführung

Aufgrund der Vorerfahrungen der Schülerinnen und Schüler bei der Programmierung mit Python konnte davon ausgegangen werden, dass die grundlegenden Konzepte der Programmierung wie Variablen, Schleifen und Verzweigungen nicht erneut auf theoretisch-abstraktem Niveau besprochen werden mussten. Aus diesem Grund und auch der geringen syntaktischen Unterschiede war nur eine Doppelstunde für die Themen Variablen und if-Verzweigungen geplant. Die auf diesen Überlegungen basierenden Lernziele sind in [Tabelle 2](#) genauer beschrieben.

Zielart	Lernziel
Grobziel	Die Schülerinnen und Schüler beschreiben die Struktur von HTML- und CSS-Dokumenten.
Feinziel	Die SuS nennen Beispiele für HTML-Tags.
Feinziel	Die SuS beurteilen die Korrektheit von CSS-Dokumenten bezüglich ihrer Syntax.
Grobziel	Die SuS identifizieren Unterschiede und Gemeinsamkeiten der Syntax von Variablen und Verzweigungen in Javascript und Python.
Feinziel	Die SuS identifizieren Code als Python- oder Javascript-Code.
Feinziel	Die SuS nennen Unterschiede in der Syntax von if-Verzweigungen in Javascript und Python.
Feinziel	Die SuS beurteilen die Qualität von Javascript-Code hinsichtlich seiner syntaktischen Korrektheit.

Tabelle 2: Planung der Lernziele für die erste Doppelstunde vor ihrer Durchführung

Ebenso sollten for- und while-Schleifen beide in einer Doppelstunde besprochen werden. Beide Strukturen waren bereits aus der Unterrichtseinheit zu Python bekannt und mussten daher nicht erst neu eingeführt werden. In [Tabelle 3](#) sind die dazu formulierten Lernziele zu finden.

Zielart	Lernziel
Grobziel	Die SuS verwenden for- und while-Schleifen, um Code wiederholt ausführen zu lassen.
Feinziel	Die SuS differenzieren zwischen korrekter und inkorrektter Syntax bei for- und while-Schleifen in Javascript.
Feinziel	Die SuS nennen Unterschiede und Gemeinsamkeiten von for- und while-Schleifen in Javascript und in Python.
Feinziel	Die SuS formulieren for-Schleifen in while-Schleifen um und umgekehrt.
Feinziel	Die SuS argumentieren, ob für ein gegebenes Problem eher eine for- oder eine while-Schleife geeignet ist.

Tabelle 3: Planung der Lernziele für die zweite Doppelstunde vor ihrer Durchführung

Die für die dritte Stunde vorgesehenen Inhalte Arrays bzw. Listen und Funktio-

nen waren so geplant, dass schnellere Schülerinnen und Schüler direkt in dieser Doppelstunde damit fertig werden würden, allerdings auch bei Bedarf die Möglichkeit bestand, noch die darauffolgende vierte Doppelstunde dafür zu nutzen. Die in [Tabelle 4](#) dargestellten Lernziele sind daher nur in Bezug auf die schnelleren Schülerinnen und Schüler gültig; sie können auch erst in der vierten Doppelstunde erfüllt werden.

Zielart	Lernziel
Grobziel	Die SuS implementieren die aus Python bekannte Struktur der Liste in der neuen Programmiersprache Javascript.
Feinziel	Die SuS nutzen die Methode <code>indexOf()</code> , um den Index eines Elements in einer Liste zu bestimmen.
Feinziel	Die SuS nutzen Indizes, um auf bestimmte Elemente in einer Liste zuzugreifen.
Feinziel	Die SuS identifizieren Unterschiede und Gemeinsamkeiten der Syntax von Listen bei Javascript und Python.
Grobziel	Die SuS verwenden Funktionen, um ihren Code zu strukturieren.
Feinziel	Die SuS unterscheiden korrekte von inkorrekt Syntax bezüglich vorgegebenen Codes zu Funktionen in Javascript.
Feinziel	Die SuS unterscheiden zwischen Funktionsaufruf und -rumpf.
Feinziel	Die SuS rufen vorgegebene Funktionen korrekt auf.
Feinziel	Die SuS implementieren Funktionen anhand eines vorgegebenen Funktionskopfes und Beschreibung der Funktionalität.

Tabelle 4: Planung der Lernziele für die dritte Doppelstunde vor ihrer Durchführung

Zielart	Lernziel
Grobziel	Die SuS implementieren die Logik eines browserbasierten Spiels mithilfe von Javascript.
Feinziel	Die SuS analysieren Problemstellungen in Bezug auf deren Umsetzung durch Programmierkonzepte.
Feinziel	Die SuS identifizieren algorithmische Lösungen für komplexe Programmierproblemstellungen.
Feinziel	Die SuS nutzen Schleifen, Listen, Verzweigungen und Funktionen, um ihren Code sinnvoll zu strukturieren.
Grobziel	Die SuS kombinieren erlernte Programmiertechniken und wenden diese im Zusammenspiel miteinander an.

Tabelle 5: Optionale und zusätzliche Lernziele für die vierte Doppelstunde vor ihrer Durchführung

Leistungsstarke Lernende würden die Zeit der vierten Doppelstunde dann nutzen können, um ihr in der Einheit gelerntes Wissen anzuwenden und ein kleines browserbasiertes Spiel wie Tic-Tac-Toe zu entwickeln. Für diese Schülerinnen und Schüler ergäben sich somit zusätzliche, über die reine Programmiertechnik hin-

ausgehende Lernziele, welche in [Tabelle 5](#) aufgelistet sind. Das Erreichen dieser Lernziele ist somit nicht für alle Schülerinnen und Schüler verpflichtend, sondern eine freiwillige Zusatzleistung, welche der Festigung und Vertiefung des Stoffes dient.

2.3 Methodik und Medien

Unter Anderem aufgrund der guten technischen Ausstattung der Schule war eine vielfältige Gestaltung des Unterrichts möglich. Nach Rücksprache mit der betreuenden Lehrkraft erfolgte die Einführung in neue Inhalte, beispielsweise die Syntax einer if-Verzweigung in der ersten Stunde, durch direkte Instruktion. Dies basierte auf früheren Erfahrungen mit der Klasse, welche nahelegten, dass diese Methode zu vergleichsweise wenigen Ablenkungen seitens der Schülerinnen und Schüler führte. Gleichzeitig wurde so ein schneller Einstieg in die neue Thematik geschaffen und damit ein zügiger Übergang in das selbstständige und schülerzentrierte Arbeiten.

Für die Arbeitsphasen wurde den Schülerinnen und Schülern folgendes Material zur Verfügung gestellt:

- Merkblätter, auf denen die in der Erarbeitungsphase kennengelernten Inhalte in kompakter und verständlicher Form erneut aufbereitet waren und somit als Arbeitshilfe genutzt werden konnten. Die Merkblätter finden sich in [Anhang B](#).
- Arbeitsblätter, auf welchen sich die Aufgabenstellungen für die aktuelle Doppelstunde finden ließen. Diese sind in [Anhang C](#) einsehbar.
- Zu den Aufgaben gehörige Codegerüste, welche neben dem von den Schülerinnen und Schülern zu bearbeitendem Javascript-Code auch auf HTML und CSS basierende Webseiten enthielten, um die praxisorientierten Aufgaben direkt testen zu können. Screenshots des Codes finden sich in [Anhang D](#).

Sämtliches Material lag grundsätzlich in digitaler Form vor und wurde über das Tauschlaufwerk der Schule kommuniziert. Zusätzlich bot sich den Schülerinnen und Schülern die Möglichkeit, auf Nachfrage Merk- und Arbeitsblätter auszudrucken, um diese in analoger Form als zusätzliche Hilfe bei der Arbeit vor sich liegen zu haben.

Weiterhin wurde bereits zu Beginn der Unterrichtseinheit kommuniziert, dass zur Information oder auch zur Weiterbildung für Interessierte zahlreiche Onlineresourcen zur Verfügung stehen. Insbesondere wurde die weit verbreitete Webseite [w3schools](#) genannt, welche neben Tutorials zu HTML, CSS, Javascript und weiteren Programmiersprachen auch Referenzdokumente zu oben genannten Komponenten der Webentwicklung zur Verfügung stellt [4].

Zur Entwicklung selbst wurden Desktop-Systeme mit Windows 10 verwendet. Als Code-Editoren bzw. IDEs kamen Notepad++ [3] und Visual Studio Code [5] zum Einsatz. Weiterhin wurde die integrierte Konsole sowie der Objekt-Inspektor des Firefox-Browsers verwendet.

Zusätzlich wurden auch Online-Quizzes verwendet, allgemeiner bekannt unter dem Namen “Kahoot”. Neben den positiven Erfahrungen, die Lehrkräfte bei ihren Klassen mit dem Format sammeln konnten [2], kann der Ansatz des game-based learning mithilfe von Kahoot positive Effekte auf Lernerfolge, Motivation und Ängste der Schülerinnen und Schüler haben sowie das Klassenklima verbessern [7]. Eingesetzt wurde das Tool sowohl bei Unterrichtseinstiegen zur Vorwissensaktivierung und Motivationsaufbau, als auch innerhalb von Übungsphasen zur Sammlung des Gruppenfokus.

3 Umsetzung

Bevor die genauen Durchführungen der einzelnen Doppelstunden betrachtet werden, sind zunächst einige Vorbemerkungen zu machen.

Erstens waren die folgenden selbst gehaltenen Stunden nicht das erste Kennenlernen der Klasse. Zuvor wurden bereits beide Hälften während Hospitationen bei Unterrichtsstunden zum Thema Python besucht und kennen gelernt. Währenddessen konnten auch erste Beobachtungen bezüglich der sozialen und persönlichen Verhältnisse der Schülerinnen und Schüler gemacht werden, welche in die in [Abschnitt 2](#) getroffene Konzeption des Unterrichts mit einfließen konnten.

Weiterhin war es nötig, dass die Klasse im eingeplanten Zeitraum eine Klassenarbeit schreiben musste, in welcher die in der vorhergehenden Unterrichtseinheit behandelten Inhalte (hauptsächlich Programmierung in Python) abgefragt werden würde. Ein großer Nachteil davon war, dass die folgende Unterrichtseinheit durch eine Doppelstunde unterbrochen werden würde, in welcher die Arbeit geschrieben wurde. Leider war es nicht möglich, den Termin anders zu legen, weswegen diese Unterbrechung mit in die Planung aufgenommen wurde.

3.1 Erste Doppelstunde

Der Einstieg in den Unterricht erfolgte problemorientiert anhand eines Beispiels. Gezeigt wurde eine Webseite, auf der die mit HTML und CSS gestaltete Eingabemaske eines Promillerechners zu sehen war. Ein Ausschnitt der genannten Webseite ist in [Abbildung 2](#) abgebildet.

Aufgabe 2: Promillerechner



Getrunkeene Flüssigkeitsmenge: Alkoholanteil:
Körpergewicht in kg: Mann? ☐ Promille berechnen 0 Promille

Abbildung 2: Webseite des Promillerechners

Bei Nutzung der Webseite wurde demonstriert, dass diese keinerlei Funktion hatte. Ein großes Ziel der ersten Doppelstunde war, den aktuellen Vorwissensstand

der Schülerinnen und Schüler zum Thema Webentwicklung herauszufinden. Die präsentierte Webseite wurde so einerseits dazu genutzt, um den Schülerinnen und Schüler einen Ausblick auf die kommenden Stunden zu geben und ihnen klar zu machen, dass wir uns nun (wieder) mit dem Thema Webentwicklung beschäftigen. Andererseits wurde damit auch eine Diskussion darüber angestoßen, aus was eine Webseite denn besteht und woran sie sich noch erinnern können. An dieser beteiligten sich die Schülerinnen und Schüler sehr engagiert und es fielen auch Fachbegriffe wie HTML. Auf Nachfrage wurde jedoch schnell ersichtlich, dass nur wenig Wissen über das Thema HTML vorhanden war.

Dies wurde als Überleitung zum ersten Arbeitsblatt genutzt, welches in [Abschnitt C](#) zu sehen ist. Das Übungsblatt sollte zunächst in Einzelarbeit bearbeitet werden, bei Schwierigkeiten konnten die Schülerinnen und Schüler jedoch selbstständig zur Partnerarbeit übergehen. In der Besprechung des Blatts wurde schnell klar, dass die meisten Fragen nur durch Internetrecherche beantwortet werden konnten und viele der Lerninhalte aus der Einheit zu HTML nicht mehr präsent waren. Mit CSS konnte nur ein Schüler etwas anfangen, die anderen hatten noch nie davon gehört. Während dies für die Schülerinnen und Schüler frustrierend gewirkt haben kann, war es dennoch ein wichtiges Vorgehen. Nun war klar, dass die kommenden Stunden nicht auf Dinge aufbauen konnten, die über ein Grundwissen von HTML hinausgingen und kein bzw. nur sehr grundlegendes CSS verwendet werden konnte.

Die anschließende Erklärung der Syntax von Javascript und ihren Aufbau als Programmiersprache verlief grundsätzlich produktiv. Von den Schülerinnen und Schülern kamen nur wenige Fragen, jedoch wurde anschließend das in [Abschnitt B](#) stehende Merkblatt als sehr hilfreich erachtet. Lediglich die für die kommenden Aufgaben notwendige Funktion `document.getElementById()` war schwierig zu verstehen. Hier konnte die Verwendung durch zahlreiche Beispiele anhand der Nutzung der Funktion in der Konsole auf einer Webseite verdeutlicht werden. Dennoch sollte dies ein Problem bleiben, welches dauerhaft bestehen blieb, was sich jedoch erst beim Feedback am Ende der Unterrichtseinheit herausstellte.

Die oben genannten Schwierigkeiten spiegelten sich auch in der Bearbeitung des folgenden Übungsblattes wider, welches sich in [Abschnitt C](#) finden lässt. Viele Schülerinnen und Schüler wussten nicht weiter. Resultierend daraus mussten zusätzliche Erklärungen und Beispiele eingeschoben sowie die Bearbeitungszeit für das Blatt nach oben angepasst werden. Schlussendlich hatte es niemand ge-

schaft, über Aufgabe 2 hinauszukommen. Die Konsequenz daraus war, dass die Planung für die nächste Stunde angepasst werden musste, um den Schülerinnen und Schülern dort genügend Zeit für die restlichen Aufgaben zu ermöglichen, da in diesen das essentielle Konzept der if-Verzweigung eingeübt wurde.

Die Durchführung der Stunde in der anderen Hälfte der Klasse war insofern anders, als dass bei der Planung dieser auf die Vorerfahrungen aus der ersten Durchführung in der anderen Klassenhälfte zurückgegriffen werden konnte. Beispielsweise war schon im Vorhinein klar, dass eine Wiederholung der Inhalte zu CSS nicht notwendig oder sogar verwirrend und frustrierend für die Schülerinnen und Schüler war. Entsprechend wurde auch das Übungsblatt zur Vorwissensreaktivierung angepasst; diese Version findet sich in [Abschnitt C](#). Außerdem konnte davon ausgegangen werden, dass die Schülerinnen und Schüler für die Bearbeitung der Aufgaben mehr Zeit als geplant benötigten und vertiefende Erklärungen zur Funktion `document.getElementById()` notwendig waren. Auch um den Teil der Klasse auf einem ähnlichen Stand zu halten wurde also weniger Stoff in der Doppelstunde behandelt, dafür jedoch mit ausführlicheren Erklärungen.

Angenommen wurde dies von der Klasse gut. Auch hier traten Schwierigkeiten bei der Verwendung der oben genannten Funktion auf, jedoch konnten diese die vertieften Erklärungen sowie zahlreiche Beispiele weitestgehend ausgeglichen werden. Grundsätzlich wirkte dieser Teil der Klasse generell motivierter für das Fach und arbeitete konzentriert mit. Einen Einfluss darauf könnte die durch das Fehlen einiger Schülerinnen und Schüler bedingte geringere Gruppengröße gehabt haben, was eine individuellere Förderung in den Arbeitsphasen ermöglichte.

3.2 Zweite Doppelstunde

Leider wurde der Start des Unterrichts aufgrund technischer Schwierigkeiten verzögert. Die Verbindung des Computers mit dem Whiteboard schlug fehl, was den Unterricht unmöglich machte. Da die betreuende Lehrkraft wegen ihrer Tätigkeit in der Schulleitung leider von einem Kollegen benötigt wurde und sie daher nicht vor Ort war, um bei der Beseitigung des Problems zu helfen, dauerte es 10 Minuten, bis der Unterricht regulär starten konnte.

Der Einstieg in die zweite Stunde erfolgte mit einem Kahoot [\[7, 2\]](#), in dem es um die bereits erlernte Syntax von Javascript ging. Dabei wurden auch mögliche Fehl-

vorstellungen bzw. Verwechslungen mit der Syntax von Python-Code aufgegriffen. Das Quiz wurde von den Schülerinnen und Schülern sehr gut angenommen und schien die Motivation für die kommende Stunde deutlich zu steigern. Ebenfalls konnten so Verständnisschwierigkeiten, die vom letzten Mal noch bestanden, beseitigt werden, beispielsweise die Verwendung von geschweiften Klammern in Javascript statt einem Doppelpunkt in Python.

Aufgrund der Schwierigkeiten, die bereits in der ersten Doppelstunde bei den Schülerinnen und Schülern aufgetreten waren, war es nicht möglich, sämtliche für die erste Stunde vorgesehenen Aufgaben zu bearbeiten und zu besprechen. Wie bereits erwähnt kam niemand über Aufgabe 2 hinaus, während in der Planung von drei bearbeiteten Aufgaben ausgegangen war. Aufgabe 3 war dabei so gestaltet, dass eine Lösung sowohl durch if-Verzweigungen, als auch durch die Nutzung einer Liste möglich war. Da die Aufgabe jedoch noch nicht bearbeitet worden war, war auch die Nutzung dieser Aufgabe als Heranführung an das neue Thema nicht möglich. Infolgedessen musste die Planung der Stunde verändert werden, so dass erst nachgeholt werden konnte, was in der letzten Stunde nicht mehr geschafft worden war.

Während der zusätzlich eingeplanten Arbeitszeit stellte sich heraus, dass die Schülerinnen und Schüler noch deutlich mehr Zeit benötigten als ursprünglich eingeplant. In der zusätzlich eingeplanten Arbeitsphase von 25 Minuten schaffte es niemand, die noch offene Aufgabe 3 fertig zu bearbeiten, insbesondere deshalb, da auch einige Schülerinnen und Schüler noch mit Aufgabe 2 beschäftigt waren. Nach Rücksprache mit der Klasse wurde daraufhin entschieden, die Doppelstunde als weitere Übungsstunde zu nutzen und das Thema Listen erst in der kommenden Stunde anzufangen. Einfluss auf diese Entscheidung hatten auch der verzögerte Start sowie die Aussage der Klasse, mehr Zeit und eine zusätzliche Erklärung wären sehr hilfreich.

Da nun die gesamte Doppelstunde dazu genutzt werden konnte, die Übungsaufgaben 2 und 3 fertig zu bearbeiten, war es sämtlichen Schülerinnen und Schülern möglich, dieses Ziel am Ende der Stunde zu erreichen. Sehr positiv war dabei, dass sich die Schülerinnen und Schüler untereinander geholfen haben und die schnelleren die langsameren unterstützt haben. Auch in der abschließenden Besprechung der Aufgaben kamen insbesondere zur Bildergalerie zahlreiche verschiedene Lösungsansätze. Dies regte unter den Schülerinnen und Schülern weitere Diskussionen darüber an, welche der vorgestellten Lösungen die eleganteste

und beste sei.

Vor diesem Hintergrund stellte sich nun die Frage, wie die entsprechende Stunde in der zweiten Hälfte der Klasse durchzuführen wäre. Einerseits war es ungünstig, die beiden Hälften auf unterschiedlichem Stand zu unterrichten, allerdings war es auch nicht sinnvoll, eine Doppelstunde ungenutzt zu lassen und Übungen zur Verfügung zu stellen, obwohl eigentlich neuer Stoff hätte behandelt werden können. Schlussendlich fiel die Entscheidung darauf, ebenfalls eine weitere Übungsstunde in der anderen Klassenhälfte durchzuführen und das Thema Listen noch um eine Woche zu verschieben. Dies geschah aus mehreren Gründen:

- Die betreuende Lehrkraft erkrankte leider, was Unterstützungsmöglichkeiten in fachlicher, aber auch technischer Hinsicht einschränkte.
- Auch die zweite Hälfte der Klasse hatte Schwierigkeiten mit der Bearbeitung von Aufgabe 2 und 3 und war damit noch nicht fertig.
- Am Tag der Durchführung war ein Streik der Bahnen angekündigt und einige Schüler aus der zweiten Klassenhälfte hatten bereits erwähnt, nicht anwesend sein zu können.
- Es gab neue Schüler, welche bei der ersten Stunde nicht da gewesen waren, welche den Stoff nacharbeiten mussten.
- In der darauf folgenden Woche würde die Klassenarbeit zum Thema Python geschrieben werden, was voraussichtlich zu Ablenkungen bei den Schülerinnen und Schülern führen würde.

Tatsächlich waren nur drei der Schüler, welche auch in der ersten Doppelstunde da gewesen waren, wieder anwesend. Zusätzlich dazu kamen zwei neue Schüler dazu, welche den Stoff nacharbeiten mussten. Insgesamt hätte die Einführung neuen Stoffes mehr negative als positive Folgen gehabt.

3.3 Dritte Doppelstunde

In der dritten Doppelstunde waren endlich alle Schülerinnen und Schüler auf einem geeigneten Lernstand, um mit neuem Stoff weiterzumachen und Listen in Javascript zu behandeln. Leider war die betreuende Lehrkraft jedoch nicht anwesend, weswegen technische oder didaktische Fragen nicht abgesprochen werden konnten. Glücklicherweise konnte der Unterricht dennoch regulär stattfinden und die Lehrkraft quasi vertreten werden.

Nach einer kurzen Begrüßung wurden zunächst einige organisatorische Dinge besprochen, was einiges an Zeit in Anspruch nahm. Dies war jedoch bereits bei der Planung der Stunde berücksichtigt und genügend Pufferzeit einkalkuliert worden. Es ging dabei um die Abwesenheit des Fachlehrers sowie die Korrektur der Klassenarbeit, welche ja in der Doppelstunde zuvor geschrieben worden war und die krankheitsbedingte Verzögerung der Rückgabe. Die letzte Stunde, in der es um Javascript ging, lag also bereits zwei Wochen zurück, weswegen eine Wiederholung des Stoffes zu Beginn der Stunde noch wichtiger wurde.

Der Einstieg in die Stunde erfolgte problemorientiert anhand Aufgabe 3 vom letzten Arbeitsblatt. Nach einer kurzen Aktivierung des Vorwissens (Worum ging es überhaupt in der Aufgabe? Was haben wir letztes Mal gemacht?) wurde das Szenario angesprochen, hundert Bilder auf der Webseite anzuzeigen. Die bisherige Lösung der Schülerinnen und Schüler mithilfe von if-Verzweigungen war dafür nicht mehr ausreichend. Dabei entstand eine Diskussion darüber, wie man dieses Problem wohl beheben könnte und was die bestmögliche Herangehensweise dafür war. Da das Konzept der Liste bereits aus Python bekannt war, wurde es nach einiger Zeit von einem der stärkeren Schüler in der Klasse vorgeschlagen.

In der anschließenden gemeinsamen Erarbeitung konnten sich alle Schülerinnen und Schüler daran erinnern, wie sie Listen in Python verwendet haben und wie dabei die Syntax war. Eine Übertragung des Konzepts auf die neue Programmiersprache Javascript fiel vielen von ihnen leicht, auch wenn die veränderte Syntax zunächst kleinere Schwierigkeiten bereitete. In der anschließenden Übungsphase war das zugehörige Merkblatt sehr hilfreich, welches sich in [Abschnitt B](#) finden lässt. Auch die Funktion `indexOf()` war den Schülerinnen und Schülern überraschenderweise neu, obwohl es in Python eine äquivalente Funktion `index()` gibt. Deren Verwendung wurde jedoch durch einige Beispiele schnell klar und nach wenigen Rückfragen gab es keine Probleme bei den zugehörigen Programmieraufgaben.

Die anschließende Übungsphase verlief im Vergleich mit den beiden vorhergegangenen Stunden deutlich reibungsloser und es gab weniger Verständnisschwierigkeiten bei den Schülerinnen und Schülern. Ein Grund hierfür könnte gewesen sein, dass die Übungsaufgaben für diese Stunde erst nach den beiden vorhergegangenen Stunden konzipiert worden waren und ihr Schwierigkeitsgrad somit an das Leistungsniveau der Klasse angepasst werden konnte. Weiterhin kann es sein, dass die Klasse mittlerweile mit der neuen Programmiersprache "warm

geworden“ war und intuitiver damit umgehen konnte, was die Bearbeitung der Aufgaben erleichtert hat. Schlussendlich ist das Thema Listen in Javascript der entsprechenden Einheit in Python natürlich ähnlich und so kann es gut sein, dass die Schülerinnen und Schüler von den vielen Parallelen profitieren konnten.

Bei der abschließenden Sicherung und Besprechung der Ergebnisse der Aufgaben kamen sehr gute Lösungsvorschläge zum Vorschein, insbesondere auch von eher leistungsschwächeren Schülerinnen und Schülern. Auf Nachfrage meinte die Klasse, dass die Art der Aufgaben besser verständlich gewesen sei und sie aufgrund der Ähnlichkeit zu Python mit dem Konzept der Liste besser umgehen könnten als bei Aufgaben, die eine häufige Verwendung von `document.getElementById()` erfordern.

Die Durchführung der Unterrichtsstunde in der anderen Hälfte der Klasse erfolgte aufgrund des hohen Erfolgs bei der ersten Durchführung analog. Da es einige Schülerinnen und Schüler gab, welche in der letzten Unterrichtseinheit gefehlt hatten, konnten besonders leistungsstarke Personen, welche schon früher mit den Aufgaben fertig waren, diesen helfen und sich so gegenseitig unterstützen. Dies resultierte in einem sehr produktiven Arbeitsklima mit dem Resultat, dass am Ende der Stunde alle Schülerinnen und Schüler der zweiten Klassenhälfte auf demselben Lernstand waren und die verpassten Inhalte größtenteils aufarbeiten konnten. Aufgaben, welche besondere Schwierigkeiten verursacht hatten, wurden hierbei jedoch ausgelassen und als Nachbereitungsaufgaben bzw. Kompensation für die fehlenden Stunden nahegelegt.

3.4 Vierte Doppelstunde

4 Reflexion

Literatur

- [1] Karl-Heinz Arnold and Silke Bakenhus. *Die deutsche Didaktik-Tradition*. Verlag Julius Klinkhardt, Bad Heilbrunn, 2015.
- [2] Ryan Dellos. Kahoot! a digital game resource for learning. *International Journal of Instructional technology and distance learning*, 12(4):49–52, 2015.
- [3] Don Ho. Notepad++: A free source code editor and notepad replacement. <https://github.com/notepad-plus-plus/notepad-plus-plus>, 2021. (Accessed 20.05.2024).
- [4] Deach Kamken and Phaosan Jehwae. Simple web application programming skills training with the website www.w3schools.com. *Proceeding: International Conference on Learning and Education (ICLE)*, 2:251–260, 2023.
- [5] Microsoft. Visual Studio Code - Open Source (“Code - OSS”). <https://github.com/microsoft/vscode>, 2024. (Accessed: 20.05.2024).
- [6] Ministerium für Kultus, Jugend und Sport Baden-Württemberg. Bildungsplan für das Berufliche Gymnasium, Fachplan Informatik TG. https://www.bildungsplaene-bw.de/In_OS, 2020. (Accessed: 07.03.2024).
- [7] Alf Inge Wang and Rabail Tahir. The effect of using kahoot! for learning – a literature review. *Computers & Education*, 149:103818, 2020.

Anhang

A Unterrichtsverlaufspläne

Unterrichtsverlaufsplan

Phase	Unterrichtsstruktur (mit Zeitplanung)	Lehrerhandeln	Schülerhandeln	Lernziele (fachliche und überfachliche)
Unterrichtseinstieg	<ul style="list-style-type: none"> Motivation (5m) 	<ul style="list-style-type: none"> Präsentation der Übungsaufgabe „Promillerechner“ und Aufwerfen des Problems: Die Webseite tut nichts Frage an die SuS, wie wir das ändern können Frage an die SuS, aus was eine Webseite denn besteht 	<ul style="list-style-type: none"> Die SuS diskutieren über Möglichkeiten, um die Funktion der Webseite herzustellen Die SuS erinnern sich zurück, was sie bereits über die Webentwicklung wissen 	<ul style="list-style-type: none"> Motivationsaufbau Vorwissensaktivierung
	<ul style="list-style-type: none"> Arbeitsblatt „Wiederholung HTML/CSS“ (15m) 	<ul style="list-style-type: none"> Austeilen des Arbeitsblattes Erläutern von Verständnisschwierigkeiten technische Hilfestellung 	<ul style="list-style-type: none"> Die SuS bearbeiten ein Arbeitsblatt, welches die Grundlagen von HTML/CSS wiederholt. Die SuS stellen Fragen zu Inhalten, die sie nicht wissen oder verstehen. 	<ul style="list-style-type: none"> Die SuS beschreiben die Struktur von HTML- und CSS-Dokumenten. Die SuS nennen verschiedene HTML-Tags und ihre Bedeutungen. Die SuS nennen CSS-Attribute und erläutern ihre Auswirkungen auf die Darstellung der Webseite.
Übung und Anwendung	<ul style="list-style-type: none"> Ergebnissicherung zum Arbeitsblatt (10m) 	<ul style="list-style-type: none"> Moderation und Diskussionsleitung Stellen von vertiefenden Fragen an die SuS 	<ul style="list-style-type: none"> Die SuS präsentieren ihre Lösungen und erläutern diese. Die SuS beantworten nach Aufruf durch die Lehrkraft Fragen (z.B. von anderen SuS gestellte) 	

(Hinweise zur Ergebnissicherung werden in den Spalten Lehrer- bzw. Schülerhandeln eingetragen)

Unterrichtsverlaufsplan

Erarbeitung	<ul style="list-style-type: none"> • Erarbeitung der Syntax, Erklärung von wichtigen Funktionen (10m) 	<ul style="list-style-type: none"> • Erklärung der Syntax von Statements, Variablenzuweisungen, if-Verzweigungen und getElementById() 	<ul style="list-style-type: none"> • Die SuS stellen Fragen, wenn sie etwas nicht verstehen • Die SuS beantworten Fragen der Lehrkraft und arbeiten aktiv im Unterrichtsgespräch mit 	<ul style="list-style-type: none"> • Die SuS beurteilen, ob Quellcodebeispiele syntaktisch korrekt sind • Die SuS identifizieren Unterschiede und Gemeinsamkeiten der JS- und Python-Syntax • Die SuS erklären die Bedeutung der Funktion getElementById()
Übung und Anwendung	<ul style="list-style-type: none"> • Arbeitsblatt „JavaScript: Variablen und if-Verzweigung“ (40m) • Ergebnissicherung zum Arbeitsblatt und Ausblick (10m) 	<ul style="list-style-type: none"> • Beantworten von Fragen • Aufrechterhaltung der Konzentration • Technische Hilfestellung • Moderation der Präsentationen • Stellen von Verständnisfragen an SuS • Beteiligung an der Diskussion 	<ul style="list-style-type: none"> • Die SuS bearbeiten das Arbeitsblatt • Die SuS stellen Fragen, wenn sie Schwierigkeiten bei der Bearbeitung der Aufgaben haben • Die SuS präsentieren ihre Lösungsvorschläge • Die SuS schlagen mögliche Verbesserungen vor 	<ul style="list-style-type: none"> • Die SuS wenden die erlernten Konzepte auf Aufgaben an • Die SuS diskutieren über die Qualität der präsentierten Vorschläge • Die SuS beurteilen die Korrektheit der Vorschläge

(Hinweise zur Ergebnissicherung werden in den Spalten Lehrer- bzw. Schülerhandeln eingetragen)

Unterrichtsverlaufsplan

Phase	Unterrichtsstruktur (mit Zeitplanung)	Lehrerhandeln	Schülerhandeln	Lernziele (fachliche und überfachliche)
Unterrichtseinstieg	<ul style="list-style-type: none"> Begrüßung Wiederholung (Kahoot) https://play.kahoot.it/v2?quizId=f7650480-3a03-4d5f-90fc-cd09ae67ad80 (15m) 	<ul style="list-style-type: none"> Begrüßung der SuS, erkundigen nach der KA letzte Woche Moderation des Kahoot Stellen von Verständnisfragen währenddessen 	<ul style="list-style-type: none"> Beantwortung der Fragen im Kahoot Die SuS reaktivieren ihr Wissen über die Konzepte Variable und if 	<ul style="list-style-type: none"> Aufbau eines produktiven Arbeitsklimas Vorwissensaktivierung
Übung und Anwendung 1	<ul style="list-style-type: none"> Bearbeitung von Aufgabe 3 von letztem Mal (Bildergalerie) (25m) 	<ul style="list-style-type: none"> Fragen beantworten technische Hilfestellung Je nach Leistungsniveau der SuS Tipps oder Hinweise geben 	<ul style="list-style-type: none"> Bearbeitung der Aufgabe Verständnisfragen stellen „da muss es doch eine schönere Lösung geben“ 	<ul style="list-style-type: none"> Wiederholung und Übung der if-Verzweigung Motivation des Themas der Stunde
Erarbeitung	<ul style="list-style-type: none"> Sicherung Aufgabe 3 Gemeinsame Erarbeitung Arrays/Listen (15m) 	<ul style="list-style-type: none"> Sammeln von Lösungsvorschlägen „Welche Möglichkeiten gäbe es noch zur Lösung dieser Aufgabe?“ Erklärung der Syntax, Vorstellung <code>indexOf()</code> etc. 	<ul style="list-style-type: none"> Präsentation von Lösungsvorschlägen Präsentation von Ideen, wie man das noch lösen könnte (→ Listen) 	<ul style="list-style-type: none"> Die SuS identifizieren gute Programmiertechniken und wenden diese selbst an Die SuS überführen die Python bekannte Struktur der Liste in die neue Programmiersprache Javascript

(Hinweise zur Ergebnissicherung werden in den Spalten Lehrer- bzw. Schülerhandeln eingetragen)

Unterrichtsverlaufsplan

Übung und Anwendung 2	<ul style="list-style-type: none"> • Übungsblatt zu Arrays/Listen (25-30m) 	<ul style="list-style-type: none"> • Fragen beantworten • technische Hilfestellung • Je nach Leistungsniveau der SuS Tipps oder Hinweise geben • Sammeln von Lösungsvorschlägen • Moderation der Diskussion zwischen den SuS • Verbesserung der präsentierten Lösungsvorschläge 	<ul style="list-style-type: none"> • Bearbeitung der Aufgabe • Verständnisfragen stellen • Präsentation der eigenen Lösung • Bewertung und Verbesserung der präsentierten Lösungsvorschläge 	<ul style="list-style-type: none"> • Die SuS implementieren Arrays in Javascript innerhalb vorgegebener Codeblöcke • Die SuS fassen zusammen, welche Programme sie entwickelt haben • Die SuS erklären, wie der geschriebene Code funktioniert • Die SuS bewerten den vorgestellten Code in Hinsicht auf Qualität
	<ul style="list-style-type: none"> • Sicherung Übungsblatt (5-10m) 			

(Hinweise zur Ergebnissicherung werden in den Spalten Lehrer- bzw. Schülerhandeln eingetragen)

Unterrichtsverlaufsplan

Phase	Unterrichtsstruktur (mit Zeitplanung)	Lehrerhandeln	Schülerhandeln	Lernziele (fachliche und überfachliche)
Unterrichtseinstieg	<ul style="list-style-type: none"> Begrüßung Organisatorisches: Infos zur KA-Rückgabe, Krankheit der Lehrkraft (5m) Wiederholung: Schwierigkeiten bei Aufgabe 3 (Bildergalerie) Problemstellung: Was tun bei 100 Bildern? → Nutzung einer Liste (10m) 	<ul style="list-style-type: none"> Rückgabe frühestens nächste Woche, Lehrkraft krank! Aufrufen der SuS Sammeln von Lösungsvorschlägen, Moderation der Diskussion 	<ul style="list-style-type: none"> Erinnern: Worum ging es? Wo waren die Schwierigkeiten? Rückführung des Problems auf bekannte Strukturen 	<ul style="list-style-type: none"> Vorwissensaktivierung Identifikation des Problems, Motivationsaufbau für die Stunde
Erarbeitung	<ul style="list-style-type: none"> Gemeinsame Erarbeitung Arrays / Listen (15m) 	<ul style="list-style-type: none"> „Welche Möglichkeiten gäbe es noch zur Lösung dieser Aufgabe?“ Erklärung der Syntax, Vorstellung <code>indexOf()</code> etc. 	<ul style="list-style-type: none"> Sammeln von Ideen, wie das Problem zu Lösen wäre "Wie würde ich es in Python lösen?" 	<ul style="list-style-type: none"> Die SuS identifizieren gute Programmier-techniken und wenden diese selbst an Die SuS überführen die aus Python bekannte Struktur der Liste in die neue Programmiersprache Javascript
Übung und Anwendung	<ul style="list-style-type: none"> Übungsblatt zu Arrays/Listen (45m) Puffer (5m) 	<ul style="list-style-type: none"> Erklärung der Aufgaben Fragen beantworten technische Hilfestellung Je nach Leistungsniveau der SuS Tipps oder Hinweise geben 	<ul style="list-style-type: none"> Bearbeitung der Aufgaben Stellen von Verständnisfragen Bei schnelleren SuS: Hilfestellung für schwächere SuS 	<ul style="list-style-type: none"> Die SuS implementieren Arrays in Javascript innerhalb vorgegebener Codeblöcke

	<ul style="list-style-type: none"> • Sicherung Übungsblatt (10m) 	<ul style="list-style-type: none"> • Sammeln von Lösungsvorschlägen • Moderation der Diskussion zwischen den SuS • Verbesserung der präsentierten Lösungsvorschläge 	<ul style="list-style-type: none"> • Präsentation der eigenen Lösung • Bewertung und Verbesserung der präsentierten Lösungsvorschläge 	<ul style="list-style-type: none"> • Die SuS fassen zusammen, welche Programme sie entwickelt haben • Die SuS erklären, wie der geschriebene Code funktioniert • Die SuS bewerten den vorgestellten Code in Hinsicht auf Qualität
--	---	--	---	--

B Merkblätter

Javascript: Syntax Variablen und if-Verzweigung

Variablen

In Javascript endet jeder Befehl mit einem Semikolon.

Variablen lassen sich mit dem Keyword `let` definieren. Beispiele:

```
let x = 3;  
let y = "Hallo Welt";  
let pi = 3.1415;
```

Mit ihnen kann man wie in Python umgehen:

```
x + pi; //Das Ergebnis ist 6.1415  
y + x; //Hier kommt "Hallo Welt3" heraus  
3.0 * x; //Das Ergebnis ist 9.0  
x / x; //Das Ergebnis ist 1
```

if-Verzweigung

Die if-Verzweigung sieht in Javascript anders als in Python aus. Statt einem Doppelpunkt und der Einrückung in Python beginnt die Verzweigung mit einer geöffneten geschweiften Klammer und endet mit einer geschlossenen.

Die Schlüsselwörter heißen – ähnlich wie in Python – `if`, `else` und `else if`.

```
if (x < 10) {  
    x = 0;  
} else if (x == 10) {  
    x = 1;  
} else {  
    x = 2;  
}
```

Verknüpfung mit dem HTML-Dokument

Um Javascript mit dem HTML-Dokument zu verbinden, gibt es mehrere Möglichkeiten. Die häufigste Variante ist das Programmieren in einer Javascript-Datei (z.B. `main.js`). Sie kann mit dem HTML-Tag `<script src="main.js" defer></script>` eingebunden werden.

Alternativ kann auch direkt innerhalb des `script`-Tags programmiert werden – zur Trennung von Inhalt und Funktion ist dies jedoch meistens nicht empfehlenswert.

Um die HTML-Elemente mithilfe von Javascript zu verändern, wird häufig die Funktion `document.getElementById("name")` genutzt. Sie wählt das Element aus, welches die ID "name" hat.

Nun können Attribute dieses Elements verändert werden, zum Beispiel die CSS-Eigenschaften oder der Inhalt des Elements:

```
let text = document.getElementById("textbox");
text.style.color = "red";
text.innerHTML = "Das ist der neue Text.";
```

Javascript: Syntax if-Verzweigung und Listen

if-Verzweigung

Die if-Verzweigung sieht in Javascript anders als in Python aus. Statt einem Doppelpunkt und der Einrückung in Python beginnt die Verzweigung mit einer geöffneten geschweiften Klammer und endet mit einer geschlossenen.

Die Schlüsselwörter heißen – ähnlich wie in Python – `if`, `else` und `else if`.

```
if (x < 10) {  
    x = 0;  
} else if (x == 10) {  
    x = 1;  
} else {  
    x = 2;  
}
```

Listen

Listen werden in Javascript sehr ähnlich wie in Python deklariert und verwendet. Um eine neue Liste zu erstellen, werden die Elemente mit Komma getrennt zwischen eckigen Klammern geschrieben:

```
let liste1 = ["Apfel", "Birne", "Clementine", "Datteln"];  
let liste2 = [2, 4, 6, 8, 10];
```

Listen können auch gemischte Datentypen enthalten:

```
let liste3 = ["Apfel", 2, "Birne", 3.1415];
```

Möchte man nun auf ein Element der Liste zugreifen, schreibt man den Index in eckigen Klammern hinter den Variablennamen:

```
let c = liste1[2]; //c hat den Wert "Clementine"  
let zahl = liste2[0]; //zahl hat den Wert 2
```

Man kann auch den Index eines Elements herausfinden, indem man die Funktion `indexOf()` verwendet:

```
let birnenIndex = liste1.indexOf("Birne"); //birnenIndex ist 1  
let piIndex = liste3.indexOf(3.1415); //piIndex hat den Wert 3
```

Die Länge einer Liste können wir z.B. mit `liste1.length` herausfinden.

Merkblatt: Funktionen in Javascript

Auch in Javascript ist es möglich, Funktionen zu definieren und diese aufzurufen. Eine Funktionsdefinition sieht dabei folgendermaßen aus:

```
function name(parameter1, parameter2, ...) {  
    // Dieser Code wird beim Aufruf ausgeführt  
}
```

Sie besteht also aus dem Schlüsselwort `function`, einem frei wählbaren Funktionsnamen und einer beliebigen Anzahl von Parametern. Der Code, den die Funktion beinhaltet, muss zwischen geschweiften Klammern stehen.

Das Schlüsselwort `return` kann innerhalb der Funktion genutzt werden, um einen Wert zurückzuliefern:

```
function summe(x, y) {  
    return x+y;  
}
```

Der Aufruf dieser Funktion könnte folgendermaßen aussehen:

```
let acht = summe(3, 5);
```

Wir haben Funktionen bereits verwendet, als wir die Aktionen eines Buttons festgelegt haben. Dafür setzt man das Attribut `onclick` eines Button-Elements auf den Namen einer Funktion:

```
let button = document.getElementById("knopf");  
  
function tuWas() {  
    // Hier steht der Code, der beim Anklicken ausgeführt wird  
}  
  
button.onclick = tuWas;
```

Das funktioniert übrigens nicht nur mit Knöpfen, sondern mit beliebigen HTML-Elementen (zum Beispiel auch Überschriften o.Ä.)!

C Übungsblätter

Wiederholung HTML/CSS

Aufgabe 1: Aufbau von HTML-Dokumenten

- a) Ein HTML-Dokument beschreibt was?
- ☐ Das Aussehen einer Webseite
 - ☐ Das Verhalten einer Webseite
 - ☐ Den Inhalt einer Webseite
- b) HTML-Dokumente bestehen aus sogenannten Tags. Welche Aussagen über Tags sind richtig?
- ☐ Tags beginnen und enden mit runden Klammern.
 - ☐ Der Inhalt des Dokuments steht im body-Tag des HTML-Dokuments.
 - ☐ Ein p-Tag muss durch einen schließenden Tag beendet werden.
 - ☐ Ein schließender Tag ist eine Wiederholung des öffnenden Tags.
 - ☐ In den Tags können bestimmte Eigenschaften festgelegt werden, z.B. mithilfe der Schlüsselwörter `id` oder `class`.
- c) Schreiben Sie auf, welche HTML-Tags Ihnen bekannt sind. Was sind ihre Bedeutungen? Benötigen sie jeweils einen schließenden Tag?

Aufgabe 2: Aufbau von CSS-Dokumenten

- a) Ein CSS-Dokument beschreibt was?
- ☐ Das Aussehen einer Webseite
 - ☐ Das Verhalten einer Webseite
 - ☐ Den Inhalt einer Webseite

- b) Welche Aussagen sind in Bezug auf CSS-Dokumente richtig?
- ☐ Ein CSS-Dokument kann mithilfe des `link`-Tags in ein HTML-Dokument eingebunden werden.
 - ☐ Ein CSS-Dokument besteht aus Blöcken, welche durch spitze Klammern gekennzeichnet sind.
 - ☐ Vor jedem Block steht ein Selektor. Selektoren sind z.B. HTML-Tags, Klassen oder IDs.
 - ☐ Jede Anweisung in einem CSS-Block endet mit einem Semikolon.
- c) Schreiben Sie auf, welche CSS-Attribute Sie kennen. Was sind ihre Bedeutungen?

Aufgabe 3: Anwendung

- a) Schreiben Sie den HTML-Code für einen Link mit dem Text „Hier klicken“, welcher auf die Webseite „<https://www.rfgs.de>“ verweist und die ID „schul-link“ hat.
- b) Schreiben Sie CSS-Code, der den Text des obigen Links rot färbt.
- c) Wo liegt der Fehler beim folgenden HTML-Code?
`<p class="beschreibung">Dieser Text beschreibt etwas.<p/>`
- d) Wo liegt der Fehler beim folgenden CSS-Code?
`p [color: red; font-size: 12px;]`
- e) Schreiben Sie HTML-Code für eine Webseite mit mehreren Textparagrafen, Überschriften und Links. Schreiben Sie dann CSS-Code, um die Textfarbe der Links auf Rot zu setzen, die Textgröße der Überschriften auf 20 Pixel zu setzen und die Hintergrundfarbe der Textparagrafen und Überschriften auf Gelb zu setzen.

Wiederholung HTML

Aufgabe 1: HTML im Browser

- a) Nutzen Sie den HTML-Inspektor Ihres Browsers, um sich die HTML-Struktur der Beispiel-Webseite anzusehen. Erinnern Sie sich noch an alle Elemente? Welches Element hat welche Bedeutung?
- b) Durch Doppelklick auf ein Element im HTML-Inspektor können Sie dieses bearbeiten. Ändern Sie die Überschrift der Webseite zu „HTML ist einfach!“
- c) Sie können auch die HTML-Tags selbst durch einen Doppelklick bearbeiten. Ändern Sie den Tag `<h1>` zu `<h2>`. Was müssen Sie beachten? Was ist der Unterschied zwischen den beiden Tags?
- d) Einige Elemente haben innerhalb ihrer Tags noch zusätzliche Angaben, wie beispielsweise `href` oder `id`. Wofür könnten diese Angaben da sein?

Aufgabe 2: Aufbau von HTML-Dokumenten

- a) HTML-Dokumente bestehen aus sogenannten Tags. Welche Aussagen über Tags sind richtig?
 - ☐ Tags beginnen und enden mit runden Klammern.
 - ☐ Der Inhalt des Dokuments steht im `body`-Tag des HTML-Dokuments.
 - ☐ Ein `p`-Tag muss durch einen schließenden Tag beendet werden.
 - ☐ Ein schließender Tag ist eine Wiederholung des öffnenden Tags.
 - ☐ In den Tags können bestimmte Eigenschaften festgelegt werden, z.B. mithilfe der Schlüsselwörter `id` oder `class`.
- b) Schreiben Sie auf, welche HTML-Tags Ihnen bekannt sind. Was sind ihre Bedeutungen? Benötigen sie jeweils einen schließenden Tag?
- c) Schreiben Sie HTML-Code für eine Webseite, welche Sie als Person vorstellt und speichern Sie diesen in einer Datei `steckbrief.html`. Verwenden Sie dazu mehrere verschiedene HTML-Elemente wie Textparagrafen, Überschriften, Links usw.

Javascript: Variablen und if-Verzweigung

Die Dateien zu den Aufgaben finden Sie im jeweiligen Ordner. Öffnen Sie die HTML-Datei im Browser und die Javascript-Datei in einem Editor.

Aufgabe 1: Personenbeschreibung

Legen Sie jeweils eine Variable für Ihren Vor- und Nachnamen, Ihr Alter und Ihre Größe in Metern an. Bauen Sie aus den verschiedenen Variablen einen Satz, welcher Sie als Person vorstellt. Geben Sie diesen Satz auf der Webseite aus, indem Sie das Attribut `innerHTML` des Elements mit der ID `ausgabe` überschreiben.

Aufgabe 2: Promillerechner

- a) Die vorgefertigte Webseite zu dieser Aufgabe enthält bereits einige Elemente. Schreiben Sie nun Javascript-Code in die bereits vorbereitete JS-Datei, um den korrekten Promillewert zu berechnen. Die Formel für eine Schätzung des Blutalkohols lautet $\text{Blutalkohol} = \frac{\text{Masse Alkohol}}{\text{Masse Person} \cdot 0,6}$, wobei sich die Masse des Alkohols errechnen lässt mit $\text{Masse Alkohol} = \text{Flüssigkeitsmenge} \cdot \text{Alkoholanteil} \cdot 0,8$. Die Werte in den Eingabefeldern erhalten Sie mit dem Attribut `value`.
- b) Die obige Formel stimmt nur, wenn es sich bei der Person um eine Frau handelt. Handelt es sich um einen Mann, muss der Wert 0,6 im Nenner auf 0,7 geändert werden. Verändern Sie also Ihren Code folgendermaßen: Wenn die Checkbox auf der Webseite aktiv ist, muss der Wert 0,7 lauten, ansonsten setzen Sie ihn auf 0,6. Den Status der Checkbox können Sie mit dem Attribut `checked` abfragen.

Aufgabe 3: Bildergalerie

Im Ordner für diese Aufgabe finden Sie neben der HTML- und der Javascript-Datei auch einige Bilder, welche von 1-5 durchnummeriert sind. Die Webseite zeigt momentan `bild1.jpg` an. Bei Klick auf den Button „Weiter“ soll das nächste Bild in der Reihe angezeigt werden, bei Klick auf den Button „Zurück“ das vorherige. Schreiben Sie Javascript-Code, der diese Funktionen der Webseite hinzufügt.

D Codegerüste zu den Übungsaufgaben

D.1 Aufgabe 0: Wiederholung HTML

Listing 1: Code zu Aufgabe 0: Wiederholung HTML

```
1  <!doctype html>
2  <html>
3      <head>
4          <title>Wiederholung HTML</title>
5      </head>
6      <body>
7          <h1>Wiederholung HTML</h1>
8          <h2>Inhalt dieser Webseite</h2>
9          <p>Diese Webseite ist ein Beispiel dafür, wie ein
              HTML-Dokument aussehen kann. Sie besteht aus mehreren
              verschiedenen Elementen, wie</p>
10         <ul>
11             <li>Überschriften</li>
12             <li>Textabschnitten</li>
13             <li>Aufzählungslisten</li>
14             <li>Bildern</li>
15             <li>Links</li>
16         </ul>
17         <h2>Wofür ist sie da?</h2>
18         <p>Das Übungsblatt enthält mehrere Aufgaben, die zur groben
              Wiederholung des Themas HTML dienen. Dabei müssen unter
              Anderem HTML-Elemente verändert werden oder neue
              Elemente hinzugefügt werden.</p>
19         <p>Bei Unsicherheiten, was die verschiedenen Tags bedeuten
              und wie man sie verwendet, kann man auf der <a href="
              https://www.w3schools.com/html/default.asp">Webseite von
              W3Schools zum Thema HTML</a> viele hilfreiche
              Informationen und Beispiele finden.</p>
20         <h2>Aus was besteht HTML-Code?</h2>
21         <p>HTML-Code kann folgendermaßen aussehen:</p>
22         
23         <p>Neben Text können wir aber auch viele andere Elemente
              erzeugen, zum Beispiel Buttons oder Eingabefelder:</p>
24         <table id="tabelle">
25             <tr>
26                 <th>Element</th>
27                 <th>Beispiel</th>
28             </tr>
```

```

29         <tr>
30             <td>Button</td>
31             <td><button id="knopf">Dies ist ein Button!</button>
                </td>
32         </tr>
33         <tr>
34             <td>Eingabefeld</td>
35             <td><input id="textfeld"></td>
36         </tr>
37         <tr>
38             <td>Beschriftung für ein anderes Element</td>
39             <td><label id="beschriftung">Dies ist ein Label!
                </label></td>
40         </tr>
41         <tr>
42             <td>Checkbox</td>
43             <td><input type="checkbox" id="input-checkbox"> Das
                ist eine Checkbox</td>
44     </table>
45 </body>
46 </html>

```

D.2 Aufgabe 1: Personenbeschreibung

Listing 2: HTML-Code zu Aufgabe 1: Personenbeschreibung

```

1 <!doctype html>
2 <html>
3     <head>
4         <title>Aufgabe 1: Personenbeschreibung</title>
5         <script src="aufgabe1.js" defer></script>
6     </head>
7     <body>
8         <h1>Aufgabe 1: Personenbeschreibung</h1>
9         <p id="ausgabe">Hier steht später deine Ausgabe.</p>
10    </body>
11 </html>

```

Listing 3: Javascript-Codegerüst zu Aufgabe 1: Personenbeschreibung

```

1
2 // Schreiben Sie hier Ihren Code für Aufgabe 1
3
4 let ausgabe = document.getElementById("ausgabe");

```

```
5  ausgabe.innerHTML = "Hello World"; // Mit diesem Code wird die
    Ausgabe erzeugt - ersetzen Sie den String "Hello World" durch
    Ihren Satz.
```

D.3 Aufgabe 2: Promillerechner

Listing 4: HTML-Code zu Aufgabe 2: Promillerechner

```
1  <!doctype html>
2  <html>
3    <head>
4      <title>Aufgabe 2: Promillerechner</title>
5      <script src="aufgabe2.js" defer></script>
6    </head>
7    <body>
8      <h1>Aufgabe 2: Promillerechner</h1>
9      <label>Getrunkene Flüssigkeitsmenge:</label> <input id="
        fluessigkeitsmenge" type="number"></input>
10     <label>Alkoholanteil:</label> <input id="alkoholanteil"
        type="number"></input>
11     <label>Körpergewicht in kg:</label> <input id="gewicht"
        type="number"></input>
12     <label>Mann?</label> <input id="mann" type="checkbox">
        </input>
13     <button id="berechnen">Promille berechnen</button>
14     <span id="ausgabe">0</span> <label>Promille</label>
15   </body>
16 </html>
```

Listing 5: Javascript-Codegerüst zu Aufgabe 2: Promillerechner

```
1  document.getElementById("berechnen").onclick = function() {
2    let fluessigkeitsmenge = document.getElementById("
        fluessigkeitsmenge").value;
3    let alkoholanteil = document.getElementById("alkoholanteil").
        value;
4    let gewicht = document.getElementById("gewicht").value;
5    let istMann = document.getElementById("mann").checked;
6
7    // Schreiben Sie hier Ihren Code für Aufgabe 2
8
9    let ausgabe = document.getElementById("ausgabe");
10   ausgabe.innerHTML = 0; // Mit diesem Code wird die Ausgabe
        erzeugt - ersetzen Sie den Wert 0 durch das Ergebnis der
        Berechnung.
```

```
11 };
```

D.4 Aufgabe 3: Bildergalerie

Listing 6: HTML-Code zu Aufgabe 3: Bildergalerie

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>Aufgabe 3: Bildergalerie</title>
5     <script src="aufgabe3.js" defer></script>
6   </head>
7   <body>
8     <h1>Aufgabe 3: Bildergalerie</h1>
9     <br>
10    <button id="zurueck">Zurück</button> <button id="weiter">
      Weiter</button>
11  </body>
12 </html>
```

Listing 7: Javascript-Codegerüst zu Aufgabe 3: Bildergalerie

```
1 document.getElementById("zurueck").onclick = function() {
2   // Code für den Zurück-Button
3 };
4
5 document.getElementById("weiter").onclick = function() {
6   // Code für den Weiter-Button
7 };
```

D.5 Aufgabe 4: Song Shuffle

Listing 8: HTML-Code zu Aufgabe 4: Song Shuffle

```
1 <!doctype html>
2 <html>
3   <head>
4     <title>Aufgabe 4: Song Shuffle</title>
5     <script src="aufgabe4.js" defer></script>
6   </head>
7   <body>
8     <h1>Aufgabe 4: Song Shuffle</h1>
9     <button id="shuffle">Shuffle!</button> <span id="ausgabe">
      Hier steht der Songvorschlag.</span>
```

```
10     </body>
11 </html>
```

Listing 9: Javascript-Codegerüst zu Aufgabe 4: Song Shuffle

```
1 document.getElementById("shuffle").onclick = function() {
2     let songs = []; //Schreiben Sie hier Ihre Lieblingssongs
        hinein.
3     let index = Math.floor(Math.random() * len); //Diese Zeile
        generiert eine Zufallszahl. Ersetzen Sie len durch die Länge
        ihrer Liste.
4
5     document.getElementById("ausgabe").innerHTML = "Hier steht der
        Songvorschlag."; //Ersetzen Sie den String, um die Ausgabe
        zu schreiben.
6 };
```

D.6 Aufgabe 5: To-Do-Liste

Listing 10: HTML-Code zu Aufgabe 5: To-Do-Liste

```
1 <!doctype html>
2 <html>
3     <head>
4         <title>Aufgabe 5: To-Do-Liste</title>
5         <script src="aufgabe5.js" defer></script>
6     </head>
7     <body>
8         <h1>Aufgabe 5: To-Do-Liste</h1>
9         <input id="eingabe"/> <button id="add">Hinzufügen</button>
10        <ul id="liste">
11        </ul>
12    </body>
13 </html>
```

Listing 11: Javascript-Codegerüst zu Aufgabe 5: To-Do-Liste

```
1 let toDoListe = [];
2
3 document.getElementById("add").onclick = function() {
4     //Schreiben Sie hier Ihren Code
5     updateList(); //Aktualisiert die angezeigte To-Do-Liste
6 };
7
8 function updateList() {
```

```

9     let liste = document.getElementById("liste");
10    liste.innerHTML = "";
11    for (let item of toDoListe) {
12        let li = document.createElement('li');
13        li.innerHTML = item;
14        liste.appendChild(li);
15    }
16 };

```

D.7 Aufgabe 6: Taschenrechner 1

Listing 12: HTML-Code zu Aufgabe 6: Taschenrechner 1

```

1  <!doctype html>
2  <html>
3      <head>
4          <title>Aufgabe 6: Taschenrechner 1</title>
5          <script src="aufgabe6.js" defer></script>
6      </head>
7      <body>
8          <h1>Aufgabe 6: Taschenrechner 1</h1>
9          <p>Für diese Aufgabe müssen Sie ausnahmsweise nicht mit dem
              HTML interagieren.
10         Schreiben Sie Ihren Javascript-Code in die Datei <code>
              aufgabe6.js</code> und überprüfen Sie
11         mithilfe der Konsole des Webbrowsers, ob Ihre Funktionen
              korrekt sind.</p>
12      </body>
13 </html>

```

D.8 Aufgabe 7: Klickzähler

Listing 13: HTML-Code zu Aufgabe 7: Klickzähler

```

1  <!doctype html>
2  <html>
3      <head>
4          <title>Aufgabe 7: Klickzähler</title>
5          <script src="aufgabe7.js" defer></script>
6      </head>
7      <body>
8          <h1>Aufgabe 7: Klickzähler</h1>
9          <button id="knopf">Klick mich!</button> <input id="zaehler"
              type="number" disabled value="0">

```

```
10     </body>
11 </html>
```

D.9 Aufgabe 8: Taschenrechner 2

Listing 14: HTML-Code zu Aufgabe 8: Taschenrechner 2

```
1  <!doctype html>
2  <html>
3      <head>
4          <title>Aufgabe 8: Taschenrechner 2</title>
5          <script src="aufgabe8.js" defer></script>
6      </head>
7      <body>
8          <h1>Aufgabe 8: Taschenrechner 2</h1>
9          <input id="feld1" type="number"> <input id="feld2" type="
              number"> <button id="rechnungStarten">Rechnung starten!
              </button>
10         <table>
11             <tr>
12                 <th>Rechenart</th>
13                 <th>Ergebnis</th>
14             </tr>
15             <tr>
16                 <td>Summe</td>
17                 <td><input id="ergebnisSumme" disabled></td>
18             </tr>
19             <tr>
20                 <td>Produkt</td>
21                 <td><input id="ergebnisProdukt" disabled></td>
22             </tr>
23             <tr>
24                 <td>Differenz</td>
25                 <td><input id="ergebnisDifferenz" disabled></td>
26             </tr>
27             <tr>
28                 <td>Quotient</td>
29                 <td><input id="ergebnisQuotient" disabled></td>
30             </tr>
31         </table>
32     </body>
33 </html>
```

E Ausdrücke der ersten Seiten der Onlineliteratur

[6]

