

Programmieraufgaben für das Erasmus+ Projekt: Interdisziplinäre MINT Scenario Games

Das Holztechnikum Kuchl, das Ulrich-von-Hutten Gymnasium Berlin und die L&K development GmbH (Software-Entwicklungsunternehmen) kooperieren im Zuge des gemeinsamen Erasmus+ Projekts „Interdisziplinäre MINT Scenario Games“.

MINT Scenario Games sind an Online-Escape-Games angelehnt, wobei hier jedoch die Lehre und Praxisrelevanz im Vordergrund stehen. Das heißt, abstrakte Konzepte in MINT-Fächern (z. B. Mathematik, Naturwissenschaften, ...) sollen spielerisch und mit Aufgaben auf dem Handy, Tablet oder PC in einem übergeordneten Fall („Scenario“) erlernt werden.

Die Scenarios können in einer Web-App erstellt und nach Veröffentlichung in den Android und iOS Apps gespielt werden. Dabei ist der Inhalt des MINT Scenario Games weitgehend frei gestaltbar.

Die folgenden Programmieraufgaben sollen anhand des Beispielszenarios „Grundlagen der Europäischen Union“ die Basics von Webentwicklung (HTML, CSS, JavaScript, arbeiten mit JS Libraries und Datenbanken) vermitteln.

Die Aufgaben bauen teilweise aufeinander auf. Die letzte Aufgabe (Programmieraufgabe 6) ist die Anwendung der vorherigen Aufgaben und ist für Fortgeschrittene gedacht. Falls du hier eine gute Lösung entwickelst, kann diese teilweise oder ganz übernommen und als Teil des Projekts veröffentlicht werden.

Die L&K development GmbH wünscht euch viel Spaß bei den Aufgaben!

Programmieraufgabe 1: HTML

Falls du noch gar keine Erfahrung mit HTML hast, lies dir am besten [diese Anleitung](#) bis inkl. „Leere Elemente“ durch und verwende sie als Referenz beim Schreiben von HTML.

1. Gehe auf <https://jsbin.com/koqedaw/1/edit?html,output>. Das ist die Vorlage für diese Aufgabe. Zurzeit ist nur ein Beispieltext in einem `<p>` Tag gegeben. Verändere das HTML im „body“ so, dass zwei Absätze mit folgendem Text erscheinen:

Die Europäische Union, kurz EU, ist ein einzigartiger wirtschaftlicher und politischer Zusammenschluss aus 27 europäischen Ländern.

Dank der EU können wir in viele Länder ganz ohne Grenzkontrollen reisen, leben in Frieden und Stabilität und profitieren indirekt von vielen Förderprogrammen und Unterstützungen.

2. Bringen wir nun etwas Struktur in die Seite! Erstelle vor den zwei Absätzen eine Überschrift. Das geht mit dem Tag `<h1>`. Es gibt außer h1 auch h2, h3 bis h6. Probiere die verschiedenen Tags aus.
3. Einige Wörter sollten hervorgehoben werden. Mithilfe des Tags `` kannst du Wörter fett und mit `` kursiv erscheinen lassen. Nutze diese, um „Europäische Union“ fett und „EU“ kursiv zu machen.
4. Jetzt fehlt nur noch ein kleines Bild. Mit dem Tag `` kann man einfach ein beliebiges Bild einbinden. Suche dir im Internet ein passendes Bild und platziere es zwischen den Titel und dem ersten Absatz. Falls du kein passendes Bild findest, nutze diesen Link im „src“ Attribut:
https://upload.wikimedia.org/wikipedia/commons/thumb/b/b7/Flag_of_Europe.svg/150px-Flag_of_Europe.svg.png

Lösung der Programmieraufgabe 1

Das Ergebnis sollte jetzt so aussehen:

Die Europäische Union



Die **Europäische Union**, kurz *EU*, ist ein einzigartiger wirtschaftlicher und politischer Zusammenschluss aus 27 europäischen Ländern.

Dank der *EU* können wir in viele Länder ganz ohne Grenzkontrollen reisen, leben in Frieden und Stabilität und profitieren indirekt von vielen Förderprogrammen und Unterstützungen.

Wenn du das geschafft hast, hast du die Basics von HTML drauf. Schaffst du auch Aufgabe 2?

Programmieraufgabe 2: CSS

Machen wir unseren HTML Output ein bisschen schöner! Wir bauen dafür auf dem Code von Aufgabe 1 auf. Du kannst also deine Kreation direkt weiterverwenden. Fall nicht ist hier die Vorlage für diese Aufgabe

<https://jsbin.com/yorulux/1/edit?html,css,output>

1. Falls du noch gar keine Erfahrung mit CSS hast, lies dir am besten [diese Anleitung](#) durch und verwende sie als Referenz beim Schreiben von CSS.
2. Um CSS zu schreiben, klicke oben auf den Button „CSS“. (Hinweis: beim „echten“ Programmieren muss eine eigene .css Datei erstellt und im HTML eingebunden werden)
3. Die jetzige Schriftart gefällt uns nicht. Setze für den body die Schriftart Helvetica.
Tipp: Nutze dafür *font-family: 'Helvetica', sans-serif;*
4. Nun wollen wir den Text des Titels dunkelblau färben. Das geht mit der Eigenschaft **color** und dem Eigenschaftswert **darkblue**. Probiert auch andere Farben aus!
5. Jetzt wollen wir ein wenig mehr Platz mit „margin“ machen! Unter dem Bild sollten 30 Pixel Abstand zum ersten Absatz sein.
6. Einmal musst du noch deine HTML Skills einsetzen. Wir wollen einen gemeinsamen Rahmen um die beiden Absätze machen. Dazu musst du die zwei „p“ Elemente in einem „div“ Element verschachteln. Gib dem neuen div die Klasse „info“.
(Tipp: `<div class="info"> ... </div>`)
7. Um den Rahmen zu erstellen, müssen wir im CSS die Klasse info ansprechen.
Tipp: `.info { ... }`
Der Rahmen lässt sich mit der Eigenschaft **border** und dem Eigenschaftswert **1px black solid** erstellen. Schaffst du es auch, den Rahmen abzurunden? Tipp: Nutze border-radius.
8. Der Text ist etwas sehr eng am Rahmen. Wir möchten also noch 10px Platz links und rechts vom text haben. Tipp: Nutze padding-right und padding-left!

Lösung der Programmieraufgabe 2

Das Ergebnis sollte jetzt so aussehen:

Die Europäische Union



Die **Europäische Union**, kurz *EU*, ist ein einzigartiger wirtschaftlicher und politischer Zusammenschluss aus 27 europäischen Ländern.

Dank der *EU* können wir in viele Länder ganz ohne Grenzkontrollen reisen, leben in Frieden und Stabilität und profitieren indirekt von vielen Förderprogrammen und Unterstützungen.

Wenn du das geschafft hast, hast du die Basics von CSS drauf. In Aufgabe 3 programmieren wir richtig!

Programmieraufgabe 3: JavaScript

Erstellen wir nun ein kleines Quiz! Wir bauen dafür wieder auf Aufgabe 2 auf. Du kannst also deine Kreation direkt weiterverwenden. Falls nicht, ist hier die Vorlage für diese Aufgabe <https://jsbin.com/kituruk/2/edit?html,output>

1. Erstelle nach der Info-div einen Tag „form“ mit der id „quiz“.
2. Im Formular brauchen wir eine Frage.
Schreibe also: „Wie viele Mitgliedstaaten hat die Europäische Union?“ in einen h4 Tag.
3. Erstelle jetzt die Eingabemaske für die Antwort. Dafür brauchen wir einen input Tag und einen dazugehörigen label Tag. Die id im for Attribut muss gleich sein, die die id des input Elements. Das Input Element sollte dabei das Attribut type=“number“ haben, damit User nur Zahlen eingeben können.

Tipp:

```
<label for="answer">Antwort:</label>
```

5.

```
<input type="number" id="answer">
```

Bonusaufgabe: Wie kann sichergestellt werden, dass User keine negativen Werte eingeben können?

4. Damit der User überprüfen kann, ob die Antwort richtig ist, brauchen wir einen button.

Tipp:

```
<button>Prüfen</button>
```

5. Wenn der User auf den Prüfen Button klickt, soll also eine Funktion ausgeführt werden, die die gegebene Antwort mit der richtigen vergleicht. Falls die Antwort richtig war, soll der Input grün gefärbt werden. Falls nicht, rot. Wie macht man das am besten?

Wenn ein Button innerhalb eines Formulars geklickt wird, wird das Formular „submit“ et.

Wir brauchen also beim form Tag ein weiteres Attribut:

```
onsubmit="event.preventDefault(); checkAnswer();" 
```

6. Hinweis: wir brauchen event.preventDefault() um zu verhindern, dass wir beim submit weitergeleitet werden.

6. checkAnswer() ist also die JavaScript Funktion, die ausgeführt wird, wenn der button im Formular geklickt wird. Die Funktion müssen wir jetzt erstellen. Öffnet dazu den JavaScript Tab durch Klicken des JavaScript Buttons oben.

Erstellt jetzt die Funktion checkAnswer()

Tipp:

```
function checkAnswer() {
```

```
}
```

7. Jetzt geht es ans Programmieren! Wir wissen natürlich, dass die richtige Antwort 27 ist. Wir brauchen aber zuerst die gegebene Antwort. Diese bekommen wir so:

```
var answerElement = document.getElementById("answer")
```

```
var answer = answerElement.value
```

Links vom "=" definieren wir die Variablen und rechts vom „=" steht der Wert, der den Variablen zugewiesen wird.

8. Nun vergleichen wir die gegebene Antwort mit der richtigen. Dafür nutzen wir nicht wie in Mathe „=“, sondern das doppelte „==“. Das einfache Gleichheitszeichen ist für die Zuweisung von Werten an Variablen reserviert. Um aus dem Vergleich eine Fallunterscheidung zu machen brauchen wir ein „if“ Statement.

```
if (answer == 27) {
```

```
} else {
```

```
}
```

Falls die Antwort 27 ist, wird der Code im ersten Block ausgeführt. Andernfalls der Code im zweiten Block nach dem „else“

9. Zu guter letzt wollen wir die Antwort grün bzw rot färben. Das answerElement haben wir ja bereits. Das können wir jetzt wiederverwenden.

Tipp: `answer.style.backgroundColor = "green"`

Lösung der Programmieraufgabe 3

Das Ergebnis sollte jetzt so aussehen:

Die Europäische Union



Die **Europäische Union**, kurz *EU*, ist ein einzigartiger wirtschaftlicher und politischer Zusammenschluss aus 27 europäischen Ländern.

Dank der *EU* können wir in viele Länder ganz ohne Grenzkontrollen reisen, leben in Frieden und Stabilität und profitieren indirekt von vielen Förderprogrammen und Unterstützungen.

Wie viele Mitgliedstaaten hat die Europäische Union?

Antwort:

Programmieraufgabe 4: JavaScript mit chartjs

Diese Aufgabe baut nicht mehr auf den vorherigen auf. Nutze bitte diese Vorlage.

<https://jsbin.com/mudihaf/1/edit?html,js,output>

1. In der Vorlage wurde bereits chartjs eingebunden und eine Beispielgrafik generiert. Du musst für die folgenden Aufgaben nur den JavaScript Code verändern.
2. Das europäische Parlament besteht aus 705 Mitgliedern und 7 Fraktionen:

Fraktion	Sitze im EP	Mitglieder aus DE AUT
Die Europäische Volkspartei (EVP)	177	CDU, CSU ÖVP
Die Progressive Allianz der Sozialdemokraten (S&D)	145	SPD SPÖ
Renew Europe (Renew)	101	FDP, Freie Wähler NEOS
Die Grünen/Europäische Freie Allianz (Grüne/EFA)	73	Grüne, Piraten, VOLT Grüne
Identität und Demokratie (ID)	68	AfD FPÖ
Europäische Konservative und Reformer (EKR)	64	LKR
Die Linke	39	Die Linke
(Fraktionslos)	38	PARTEI

Stelle diese 8 Gruppen mit der Anzahl der Sitze im EP in einem „Doughnut“ Diagramm dar. Ändere dazu im JavaScript code bei config den type auf „doughnut“. Danach musst du die variablen labels und data anpassen. Verwende dabei passend die Farben der Fraktionen.

Tipp: Schau auf <https://www.chartjs.org/docs/latest/charts/doughnut.html> => Setup, um Beispielcode für ein doughnut Diagramm zu sehen.

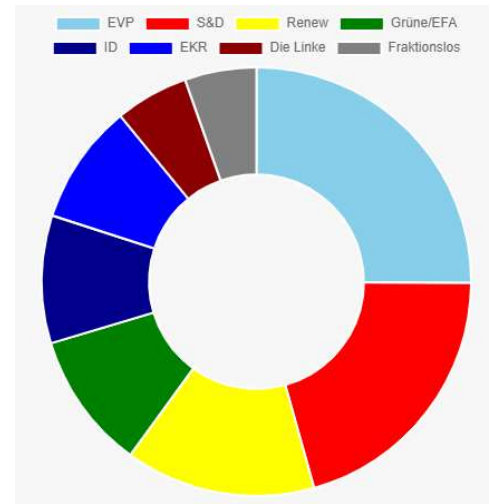
Tipp 2: Die Farben der Fraktionen:

EVP: skyblue, S&D: red, Renew Europe: yellow, Grüne: green, I&D: darkblue, EKR: blue, Linke: darkred, Fraktionslos: grey

Lösung der Programmieraufgabe 4

Das Ergebnis sollte jetzt so aussehen:

Wenn du bisher alle Aufgaben geschafft hast, hast du auf jeden Fall das Zeug zum Frontend Entwickler!



Programmieraufgabe 5: Datenbanken

Um eine solide Webanwendung zu bauen, braucht man auch eine gut entworfene Datenbank. Für die Szenarienerstellung gilt folgendes:

Ein User hat einen Benutzernamen, ein Passwort und eine E-Mail-Adresse und soll ein Szenario erstellen können. Ein Szenario hat einen Titel, eine Beschreibung, einen Autor (user) und besteht aus mehreren Aufgabengruppen.

Aufgabengruppen haben ebenfalls einen Titel, zusätzlich ein Gewicht (weight, bestimmt die Reihenfolge) und bestehen aus Aufgaben und Infotexten.

Aufgaben können vom Typ Multiple Choice, Freitext oder Zahlenantwort sein. Jede Aufgabe hat einen Titel, einen Fragetext, eine korrekte Antwort und innerhalb einer Aufgabengruppe ein Gewicht (weight s. o.). Multiple Choice Aufgaben haben zudem noch Antwortmöglichkeiten.

Infotexte haben einen Titel, einen Textkörper (body) und ebenfalls ein Gewicht.

User, Szenarios, Aufgabengruppen, Aufgaben und Infotexte haben jeweils eine eindeutige ID.

1. Erstelle mit den gegebenen Informationen ein ER-Diagramm (ein gutes online Tool dafür ist <https://erdplus.com/standalone>)
2. Erstelle einen Entwurf der Tabellen, die erzeugt werden müssen.

Programmieraufgabe 6: Komplettes Frontend

Die Aufgabe hat es in sich: falls dich die letzten 5 Aufgaben gelangweilt haben und du schon richtig gut mit HTML, JS und CSS umgehen kannst, hast du bei dieser Aufgabe freie Gestaltungsmöglichkeit.

Hier findest du die gesamten Daten unseres Beispielszenarios im JSON Format:

<https://scenario.laknet.de/scenario>

Derzeit gibt es noch keine Möglichkeit, die Szenarien auf der Webseite selbst durchzuspielen. Erstelle ein geeignetes Frontend inkl. Design, damit die Daten, die im JSON enthalten sind, tatsächlich spielbar werden.