

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="Style.css">
</head>
<body>
  <div class="ueberschrift">Hezlich Willkommen bei der Quiz-App</div>
  <div class="knopfRand">
    <button class="knopf">
      <a href="Quiz.html">Start</a>
    </button>
    <button class="knopf-bedienung">
      <a href="Bestenliste.html">Bestenliste</a>
    </button>
  </div>

</body>
</html>
```

/* Farben basieren auf dem Google Material Design: <https://material.io/guidelines/style/color.html#color-color-palette> */

```
body {
  display: flex;
  justify-content: center;
  align-items: center;
  background-color: #BBDEFB; /* Blauer Hintergrund */
  font-family: 'Roboto', sans-serif;
  font-size: 14px;
  color: #212121;
  margin: 0;
  padding: 0;
  height: 100vh;
  flex-direction: column;
}

.knopfRand {
  display: flex;
  justify-content: center; /* Buttons horizontal mittig ausrichten */
  align-items: center; /* Buttons vertikal mittig ausrichten */
  background-color: #3F51B5;
  padding: 20px;
  margin: 20px auto; /* Auto-Margin für zentrierte Ausrichtung */
  border-radius: 2px;
  box-shadow: 0 2px 2px 0 rgba(0, 0, 0, 0.14), 0 3px 1px -2px rgba(0, 0, 0, 0.12), 0 1px 5px 0 rgba(0, 0, 0, 0.2);
  width: 50%;
  flex-direction: column;
}

.ueberschrift {
  font-size: 24px;
  font-weight: 500;
  margin-bottom: 10px;
  color: #3F51B5;
}

button a {
  color: #FFFFFF; /* Weißer Text auf den Buttons */
  text-decoration: none; /* Entfernt die Unterstreichung */
  flex-direction: column;
}

button {
  background-color: #3F51B5; /* Dunkelblaue Buttons */
  color: #FFFFFF; /* Weißer Text auf den Buttons */
  padding: 10px 20px;
  border: none;
  border-radius: 2px;
  cursor: pointer;
  transition: background-color 0.3s ease; /* Übergangseffekt */
}

button:hover {
  background-color: #303F9F; /* Dunklerer Blauton beim Überfahren mit der Maus */
}

.antworten {
  display: flex;
```

```

    flex-direction: column;
    align-items: center;
    width: 50%;
    margin: 20px 0;
}

.knopf {
    display: flex;
    flex-direction: column;
    align-items: center;
    margin: 20px 0; /* 20px Abstand nach oben und unten */
    width: 100%;
}

.frage{
    font-size: 30px;
    font-weight: 500;
    margin-bottom: 10px;
    margin-top: 1rem; /* 1rem = 16px */
    color: #3F51B5;
}

.punktezahl{
    font-size: 30px;
    font-weight: 500;
    margin-bottom: 10px;
    margin-top: 1rem;
    color: #3F51B5;
}

.fortschritt{
    background-color: #CDDC39; /* Die Farbe des Fortschrittsbalkens */
    height: 20px; /* Die Höhe des Fortschrittsbalkens */
    width: 0%; /* Die anfängliche Breite des Fortschrittsbalkens */
}

.timer{
    font-size: 30px;
    font-weight: 500;
    margin-bottom: 10px;
    margin-top: 1rem;
    color: #3F51B5;
}

```

```
let frage = document.getElementById("frage");
```

```
let knopf1 = document.getElementById("knopf1");  
let knopf2 = document.getElementById("knopf2");  
let knopf3 = document.getElementById("knopf3");  
let knopf4 = document.getElementById("knopf4");  
let pauseKnopf = document.getElementById("pauseKnopf");  
let beendenKnopf = document.getElementById("beendenKnopf");
```

```
let antwortAusgewaehlt = false;  
let punkteAnzeige = document.getElementById("punkteAnzeige");  
let ausgewaehlterKnopf;  
let timer;
```

```
let punkte = 0;
```

```
let aktuelleFrageIndex = 0;  
let ausgewaehlteAntwort;
```

```
let fragenListe = [  
  {  
    frage: "Was macht man mit einem Hund ohne Beine",  
    auswahl1: "Gassi gehen",  
    auswahl2: "Um die Häuser ziehen",  
    auswahl3: "Nichts, der kommt eh nicht",  
    auswahl4: "Ihn streicheln",  
    antwort: 2  
  },  
  {  
    frage: "Was ist der Unterschied zwischen einem Keks und einem Elefanten",  
    auswahl1: "Der Keks ist kleiner",  
    auswahl2: "Der Elefant ist größer",  
    auswahl3: "Der Keks ist zum Essen da",  
    auswahl4: "Der Elefant ist ein Tier",  
    antwort: 3  
  },  
  {  
    frage: "Wer ist ein ET?",  
    auswahl1: "Ein Außerirdischer",  
    auswahl2: "Ein Mensch",  
    auswahl3: "Ein Tier",  
    auswahl4: "Ein Roboter",  
    antwort: 1  
  }  
];
```

```
// Eventlistener
```

```
knopf1.addEventListener("click", function () {  
  if(antwortAusgewaehlt == false) {  
    ausgewaehlteAntwort = 1;  
    ausgewaehlterKnopf = 1;  
    ueberpruefen();  
    antwortAusgewaehlt = true;  
  }  
});
```

```

knopf2.addEventListener("click", function () {
    if(antwortAusgewaehlt == false) {
        ausgewaehlteAntwort = 2;
        ausgewaehlterKnopf = 2;
        ueberpruefen();
        antwortAusgewaehlt = true;
    }
});

```

```

knopf3.addEventListener("click", function () {
    if(antwortAusgewaehlt == false) {
        ausgewaehlteAntwort = 3;
        ausgewaehlterKnopf = 3;
        ueberpruefen();
        antwortAusgewaehlt = true;
    }
});

```

```

knopf4.addEventListener("click", function () {
    if(antwortAusgewaehlt == false) {
        ausgewaehlteAntwort = 4;
        ausgewaehlterKnopf = 4;
        ueberpruefen();
        antwortAusgewaehlt = true;
    }
});

```

```

beendenKnopf.addEventListener("click", function () {
    window.location.href = "Startseite.html";
    alert("Du hast " + punkte + " Punkte erreicht");
});

```

// Funktionen

```

function frageAnzeigen() {
    let aktuelleFrage = fragenListe[aktuelleFrageIndex];
    frage.innerText = aktuelleFrage.frage;
    knopf1.innerText = aktuelleFrage.auswahl1;
    knopf2.innerText = aktuelleFrage.auswahl2;
    knopf3.innerText = aktuelleFrage.auswahl3;
    knopf4.innerText = aktuelleFrage.auswahl4;

    /* clearInterval(timer);
    let sekunden = 10;
    timer = setInterval(function () {
        sekunden--;
        if (sekunden < 10) {
            sekunden = "0" + sekunden;
        }
        document.getElementById("sekunden").innerText = sekunden;
        if (sekunden == 0) {
            clearInterval(timer);
            aktuelleFrageIndex++;
            naechsteFrage();
        }
    });
    */
}

```

```

    },1000); */
}

function farbeAendern() // Funktion um die Farbe der Knöpfe zu ändern
{
    let aktuelleFrage = fragenListe[aktuelleFrageIndex];
    let knopf = null;

    switch (ausgewaehlterKnopf) { // Switch-Case um den ausgewählten Knopf zu ermitteln
        case 1:
            knopf = knopf1;
            break;
        case 2:
            knopf = knopf2;
            break;
        case 3:
            knopf = knopf3;
            break;
        case 4:
            knopf = knopf4;
            break;
        default:
            console.log("Fehler");
            return; // Verlässt die Funktion oder den Codeblock
    }

    if (ausgewaehlteAntwort == aktuelleFrage.antwort) {
        knopf.style.backgroundColor = "green";
    } else {
        knopf.style.backgroundColor = "red";
    }
}

setTimeout(function() { // Timeout um die Farbe wieder zu ändern
    knopf.style.backgroundColor = "#3F51B5";
}, 2000);
}

function ueberpruefen() {
    let aktuelleFrage = fragenListe[aktuelleFrageIndex];

    if (ausgewaehlteAntwort == aktuelleFrage.antwort) {
        farbeAendern()
        punkte++;
        punkteAnzeige.innerText = "Punkte: " + punkte;
    }
    else {
        console.log("Falsch");
    }
    farbeAendern()

    setTimeout(function() { // Timeout um die Farbe wieder zu ändern
        aktuelleFrageIndex++;
        naechsteFrage();
    }, 2000);
}

function naechsteFrage() {
    if (aktuelleFrageIndex < fragenListe.length) {
        frageAnzeigen();
        let fortschritt = document.getElementById("fortschritt");

```

```
    let fortschrittProzent = (aktuelleFrageIndex / fragenListe.length) * 100;
    fortschritt.style.width = fortschrittProzent + "%";
    fortschritt.innerText = parseInt(fortschrittProzent) + "%";
    antwortAusgewaehlt = false;
  }
  else {
    console.log("Ende");
    weiterleiten();
  }
}
```

```
function weiterleiten() {
  setTimeout(function () {
    window.location.href = "Ende.html";
  }, 1000);
}
```

```
frageAnzeigen();
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" href="Style.css">
</head>
<body>
  <div class="ueberschrift">Löse die folgenden Fragen</div>
  <div class="knopfRand">
    <div class="fortschritt" id="fortschritt"></div>
  </div>
  <div id="sekunden" class="timer"></div>

  <div class="punktezahl" id="punkteAnzeige">Punkte: 0</div>
  <div class="frage" id="frage"></div>
  <div class="antworten">
    <button id="knopf1" class="knopf">
      <div id="antwort1"></div>
    </button>
    <button id="knopf2" class="knopf">
      <div id="antwort2"></div>
    </button>
    <button id="knopf3" class="knopf">
      <div id="antwort3"></div>
    </button>
    <button id="knopf4" class="knopf">
      <div id="antwort4"></div>
    </button>
  </div>
  <div class="knopfRand">
    <button class="knopf-bedienung" id="beendenKnopf">Stop</button>
  </div>

</body>
<script src="Quiz.js"></script>
</html>
```


Beleg webbasiertes Lernprogramm Übersicht

Es soll ein webbasiertes Lernprogramm erstellt werden. Als Fundament nutzen wir die Technik der Progressive Web App (PWA). Der Beleg dient zur praktischen Anwendung der Kenntnisse zu HTML, CSS und Javascript. Die Umsetzung als PWA ermöglicht auch die einfache und komfortable Nutzung in mobilen Geräten.

Lernaspekte

- Nutzung von HTTP/HTTPS
- Einsatz von HTML zur Strukturierung
- Einsatz von CSS zur Formatierung
- Webprogrammierung mittels Javascript (ECMAScript)
- Wahl einer geeigneten Softwarearchitektur
- Nutzung einer JS-Bibliothek zur Darstellung von speziellen Inhalten
- Entwurf und Implementierung eines sinnvollen Nutzerinterfaces
- Implementierung eines responsive Designs
- Nutzung der Technik einer PWA
- Offline-Nutzung einer Webapp
- dynamisches Nachladen von Inhalten mittels Ajax-Technik
- Nutzung einer REST-Schnittstelle

Beschreibung

Das Lernprogramm soll mindestens folgende Funktionalität besitzen:

- Wahl zwischen mindestens 3 verschiedenen lokal gespeicherten Aufgabenkategorien (Mathematik, Internettechnologien und allgemeines Wissen sind Pflicht) sowie einer Aufgabenkategorie, bei welcher die einzelnen Aufgaben von einem bereitgestellten externen Server mittels Ajax und REST-API geholt werden.
- zufällige Auswahl und Darstellung einer Aufgabe mit 4 Auswahlmöglichkeiten
- Anzeige des Lernfortschritts nach jeder Aufgabe
- Anzeige einer Statistik am Ende eines Durchlaufs
- die Anzeige sollte sich an verschiedene Anzeigegeräte (PC-Browser, Tablet, Smartphone) sinnvoll anpassen (responsive Design)
- der Beleg soll auf dem Webserver der HTW-Dresden abrufbar sein, Pfad: ~sxxxxx/Lernprogramm
- Abgabe entsprechend Abgabeformat

Technische Umsetzung

- nutzen Sie für die Umsetzung HTML5/CSS3/JS
- nutzen Sie in JS den strikten Modus
- der Beleg sollte im aktuellen Firefox oder Google Chromium lauffähig sein, es wird keine Abwärtskompatibilität erwartet
- entsprechend einer PWA sollte sich die Anwendung auf einem Smartphone installieren lassen
- man benötigt in einer PWA ein Manifest und einen Service Worker zur Steuerung des Caches für den Offline-Betrieb und die Installation
- verwenden Sie keine weiteren Frameworks wie jquery, Bootstrap etc., sondern nutzen Sie die Funktionalität von ECMAScript und CSS3 in den aktuellen Browsern (TypeScript ist für Entwickler mit Vorkenntnissen erlaubt)
- Als Entwicklungsumgebung empfiehlt sich die Nutzung der Entwicklertools im Browser Chromium oder Firefox
- zum Testen der Funktionalität auf einem Smartphone kann die Device Toolbar in o.g. Entwicklertools genutzt werden
- für die grafische Formeldarstellung (Rendering) sollte die JS-Bibliothek KaTeX genutzt werden, siehe Beispiel
- für die grafische Notendarstellung sollte die JS-Bibliothek Vexflow mit der Notensprache EasyScore genutzt werden
- das Format der Fragen ist JSON entsprechend folgendem Fragment (a - Aufgabe, l - Antworten, die erste ist immer korrekt, bei der Anzeige sind die Antworten sinnvollerweise zu verwürfeln ;-)):

```
{
  "teil-mathe": [
    {"a": "x^2+x^2", "l": ["2x^2", "x^4", "x^8", "2x^4"]},
    {"a": "x^2*x^2", "l": ["x^4", "x^2", "2x^2", "4x"]}
  ]
  "teil-internettechnologien": [
```

```

    {"a": "Welche Authentifizierung bietet HTTP", "l": ["Digest Access Authentication", "OTP", "OAuth", "2-Faktor-Authentifizierung"]},
    {"a": "Welches Transportprotokoll eignet sich für zeitkritische Übertragungen", "l": ["UDP", "TCP", "HTTP", "Fast Retransmit"]},
    ...
  ]
  "teil-allgemein": [
    {"a": "Karl der Große, Geburtsjahr", "l": ["747", "828", "650", "1150"]},
    ...
  ]
  "teil-noten": [
    {"a": "C4", "l": ["C", "D", "E", "H"]},
    {"a": "(C4 E4 G4)", "l": ["C", "H", "F", "D"]},
    ...
  ]
}

```

REST-Schnittstelle des externen Aufgabenservers

Es soll im Beleg die Möglichkeit bestehen, weitere Aufgaben von einem externen Server mittels REST zu laden. genutzt wird das Projekt Web-Quiz mit der entsprechenden API für das Holen der Aufgabe und die Überprüfung der Lösung.

Das Web-Quiz-Projekt ist auf einem Server der Informatik gehostet. Die Eckdaten dieses Servers werden in der Lehrveranstaltung bekannt gegeben bzw. finden Sie im Chat

Es sind bereits Aufgaben vorhanden, welche Sie nutzen können (ID 2-33).

Sie sollten mit Ihrem eigenen Account auch einige Aufgaben hochladen.

per AJAX-Request muss lediglich die Aufgabe geholt werden und das Ergebnis überprüft werden, alle anderen notwendigen Aufgaben (Nutzer + Aufgaben anlegen) können außerhalb des Lernprogramms per CURL erledigt werden

Befüllen Sie die Datenbank am besten per Script, so können Sie Ihre Daten auch im Falle eines Problems schnell wieder auffüllen.

Vorschlag für Vorgehen bei der Bearbeitung

Erstellung des HTML-Gerüsts mit allen Elementen

Nutzung von CSS zur Gestaltung + Responsive Design

Erstellung der Javascript-Programmstruktur (Architektur Model-View-Presenter empfohlen)

Implementierung einer geeigneten Model-Schnittstelle zum Erhalt der Aufgabe und zur Übergabe der gewählten Lösung (zunächst mit einfacher Dummy-Frage)

Implementierung der Button-Handler, welche die Auswertefunktion des Presenters aktivieren

Erweiterung des Models auf verschiedene Aufgaben mit Zufallsfunktion

Implementierung der Statistikfunktionalität

Erweiterung der Anzeige auf andere Aufgabentypen (Mathe -> Katex, etc.)

Erweiterung des Models um die Nutzung der angebotenen REST-Schnittstelle

Offlinefunktionalität implementieren (minimaler Serviceworker in Beispiel)

Weitere Anforderungen

Dokumentation des Projektes, so dass eine andere Person ggf. am Projekt weiterarbeiten könnte

Erstellung eines Lernportfolios (Dokumentation Ihrer Entwicklungsschritte, des Lernfortschritts, der Misserfolge, etc.)
machen Sie Vorschläge zur Erweiterung/Verbesserung des Belegs

Mögliche Erweiterungen (optional, Zusatzpunkte möglich)

Wichtung der Aufgabenstellung anhand der bisherigen Ergebnisse

Erweiterung auf mögliche Mehrfachauswahl

zusätzliche Kategorie Notenlernen vorsehen (Einzelnoten / Akkorde / Umkehrungen ganz nach Belieben / Klaviatur).

Speicherung der erreichten Punkte im Browserspeicher oder per PHP-Script auf dem Server

Mehrnutzerbetrieb mit Nutzerauthentifizierung

Links

KaTeX
Web-Quiz
reines Javascript
Fehlersuche - Stackoverflow

Prinzipdarstellung

Die Darstellung unten zeigt prinzipiell, wie der Beleg auf einem Smartphone aussehen könnte. Sie sind nicht an die Darstellung gebunden. Die HTML-Elemente wurden für den kleinen Viewport mittels CSS-Mediaqueries untereinander dargestellt. Auf einem Desktopbrowser würde die Darstellung teilweise nebeneinander erfolgen. Die Darstellung dient nur zur Orientierung. Sie können eine abweichende Oberfläche erstellen. Der Screenshot wurde mit den Entwicklertools des Browsers erstellt.