

EXAM, Web Scripting  
BWI Abendform 2022

## Praktischer Zwischentest

---

Bei diesem praktischen Zwischentest gibt es 28 Punkte zu erreichen.

---

### Allgemeine Hinweise:

- Der praktische Zwischentest muss auf Ihrem eigenen Rechner durchgeführt werden.
- Ziel ist es Ihre Umsetzungskompetenz mit JavaScript zu evaluieren

### Unterlagen:

- Sie dürfen Internet, Bücher und eigene Programmstücke verwenden.
- Definitiv **verboten** ist die Unterstützung durch **andere Personen** bzw. ein **Datenaustausch** in jeglicher Form (manuell, elektronisch, ...).
- Ihre Abgaben werden einer Plagiatsprüfung unterzogen.

### Abgabemodus:

- Packen Sie Ihr gesamtes Projekt (alle HTML/JS/etc. Files) in **eine** zip-**Datei** und laden Sie diese in Moodle hoch. Dabei soll die vorgegebene Ordnerstruktur erhalten bleiben. Kontrollieren Sie Ihren erfolgreichen Upload bzw. die Validität der hochgeladenen Datei.

Wenn ein oder mehrere Ergebnisse zur Gänze oder auch nur in Teilen gleich oder sehr ähnlich sind (sodass ersichtlich ist, dass Ergebnisse kopiert wurden), dann wird die Prüfung für alle Beteiligten mit 0 Punkten beurteilt.

Aufgabenstellungen in grüner Schrift sind optional und können ggf. Punkte anderer fehlender bzw. unvollständiger Aufgabenstellungen kompensieren.

---

Viel Erfolg!

## IMPLEMENTIERUNG WEBANWENDUNG

### Vorbereitungen:

Nach dem Öffnen Ihrer Entwicklungsumgebung und dem Import der vorgegebenen Dateien erscheint die Projektstruktur, wie nachfolgend sichtbar. Dabei gilt, dass rote Teile erweitert werden müssen, schwarze Teile bleiben unverändert. Achtung! Sie müssen die Dateien über den Server ausführen, da sonst ein CORS Problem auftritt!

- Projektordner: **EXAM\_WebScripting\_Products**
  - Datei: **index.html**
  - Datei: **getdata.js**
  - Datei: **myStyle.css**
  - Datei: **products.json**

### Überblick Beispiel-Lösung:

- 1) Beim Aufruf der Webseite wird die Liste der Produkte automatisch geladen und wie folgt angezeigt:

### Products

Arduino UNO
Arduino Nano
Raspberry Pi 4
Raspberry Pi 3 Model B+
Raspberry Pi Zero

- 2) Wenn der AJAX-Call nicht funktioniert, erscheint z.B. folgende Fehlermeldung:

### Products

an error occurred

- 3) Wenn eines der Produkte aus 1) angeklickt wird, verschwindet die Liste wieder und nur das angeklickte Produkt wird angezeigt.

### Raspberry Pi Zero

A small single-board computer from the UK with 512MB RAM

#### Tipp:

Arbeiten Sie am Styling erst nachdem Sie die Funktionalitäten implementiert haben.

## 1 Grundidee

Auf der Webseite (**index.html**) werden die Produkte angezeigt, welche über die Datei **products.json** via JavaScript geladen werden. Die Elemente dieser Liste können angeklickt werden. Daraufhin verschwindet die Liste und nur das angeklickte Produkt wird angezeigt.

In der HTML Datei finden Sie HTML Elemente mit der ID **output** (Ausgabefeld) und **selection** (Detailfeld)

## 2 Beschreibung von products.json

Bevor Sie mit der Entwicklung der Funktionalitäten beginnen, machen Sie sich bitte mit dem Inhalt der Datei **products.json** vertraut. Jedes Objekt in diesem Array enthält Information zu „name“, „detail“, „stock“ und „price“.

## 3 Umsetzung der Funktionalitäten in getdata.js

Setzen Sie folgende Funktionalitäten mit JavaScript um:

- 3a Implementieren Sie einen AJAX-Call der ausgeführt wird, wenn die Seite aufgerufen wird. (2P)
  - i) Erstellen Sie eine beliebige Fehlermeldungsangabe wenn die Daten nicht geladen werden können. (1P)
- 3b Laden Sie die Inhalte von products.json via AJAX und nehmen Sie diese entgegen. (2P)
  - i) Hinweis: via `console.log(response)` ; können Sie die Daten auch in der Konsole sichtbar machen.
- 3c Für die Webseite werden einige Daten aus dem JSON File benötigt. Zeigen Sie diese entsprechend der Beispiel-Lösung auf der Webseite an. Erstellen Sie via JavaScript in einer Variable ein DIV Element pro eingelesenem Item und konfigurieren Sie dieses folgendermaßen:
  - i) Speichern Sie die Informationen von „detail“ und „price“ in ein Dataset. (2P)
  - ii) Alternieren Sie die Hintergrundfarbe bei der Darstellung zeilenweise. Verwenden Sie dazu die Klasse „greyline“ der myStyle.css-Datei. (2P)
  - iii) Machen Sie ihr DIV Element anklickbar, wenn es angeklickt wird, soll die Funktion „showdetails“ aufgerufen werden (wird später erstellt) (2P)
  - iv) Prüfen Sie ob ausreichend Waren für das Produkt verfügbar sind, wenn nichts on stock ist, soll das Element nicht anklickbar sein (2P)
  - v) Stellen Sie den Namen des Produktes dar und fügen Sie es dem DIV Element hinzu (2P)
  - vi) Fügen Sie das neu erstellte DIV ihrem HTML Ausgabefeld zu (2P)
  - vii) Verwenden Sie ein slide-Down mit einer Verzögerung von 350 ms, um ihr Ausgabefeld auf der Webseite einzublenden. (2P)
  - viii) Erstellen Sie einen Rahmen für Ihr Ausgabefeld via JavaScript. (2P)
- 3d Erstellen Sie eine neue Funktion mit dem Namen „showdetails“ welche aufgerufen wird, sobald ein Element in der Produktliste angeklickt wird. (2P)
- 3e Nehmen Sie den Inhalt des angeklickten Listeneintrags in dieser Funktion entgegen (2P)
- 3f Setzen Sie für die Produktliste ein slide-Up von 300 ms und fügen Sie den angeklickten Listeneintrag im Detailfeld mit einem fade-in von 200 ms ein. (2P)
- 3g Ersetzen Sie die Überschrift wo derzeit Products angezeigt wird durch den Produktnamen und stellen sie im Detailfeld die details, die Sie aus dem DataSet entnehmen, dar (3P)
- 3h Ergänzen Sie das Beispiel um die fehlenden Informationen aus dem JSON File und stellen Sie diese im Detailfeld optisch ansprechend dar (3P)