

## JavaScript Aufbau

Bausteinprüfung: IEjQuery Javascript - Entwickeln mit jQuery

Trainer: Rech Steffen

Bearbeitungszeit: bis 16:00 inklusive Pausenzeit frei einteilbarer

Gesamtpunktzahl: 100

Hilfsmittel: Internet, Unterlagen

Lesen Sie bitte zuerst die Bearbeitungshinweise und alle Arbeitsanweisungen in Ruhe durch.

Bearbeitungshinweise:

- Nutzen Sie für jede Aufgabe eine Datei aufgabe-n.html
- Ändern Sie den Ordernamen nach folgendem Schema: vorname-nachname
- Kommentieren Sie Ihren Code aus. Nicht funktionaler aber auskommentierter Code wird bewertet.
- Syntaxfehlermeldungen in der Browserkonsole führen dazu das die Aufgabe mit 0% bewertet wird.
- Die Prüfung ist bestanden, wenn mindestens 50% erreicht wurden.
- Achten Sie auf Hinweise im Lernraum (Hinweispods)
- Nicht hochgeladene Prüfungen werden mit 0% bewertet.

Abgabe der Prüfung:

Rechtsklick auf den Ordner -> Senden an zipkomprimierter Ordner.

Laden Sie diesen Ordner im E-Campus unter Bausteinprüfungen hoch.



ihr-name.zip

Viel Erfolg

## Aufgabe 1 - Zufällige Bildanzeige beim laden der Seite

### Teilpunktzahl: 25 Punkte

Nutzen Sie für jede Aufgabe die Vorlagendatei aufgabe1.html

Wenn die Seite geladen wird muss eines der folgenden Bilder zufällig in der #key-visuals ausgegeben werden. Der Bereich #key-visual ist im Dokument mit einem roten Rahmen versehen. Die Klasse hier soll ihnen den Bereich im Dokument nur zeigen und hat keine Bedeutung.

Erzeugen Sie ein Objekt in dem Sie die Daten für die Bilder ablegen.

- haus-im-himmel.jpg, alt: „Holzhütte vor wolkeigem Himmel“
- ziemlich-beste-freunde.jpg, alt: Ziemlich beste Freunde“
- briefkaesten.jpg, alt: Amerikanische Briefkästen auf dem Land“

Denken Sie daran eine Fehlermeldung (alert) auszugeben wenn der Ajax-Request einen Fehler verursacht.

## Aufgabe 2 - Aktuelle Nachrichten zum gewählten Thema

### Teilpunktzahl: 25 Punkte

Nutzen Sie für jede Aufgabe die Vorlagendatei aufgabe2.html und artikel.json

Der Bereich #artikel ist im Dokument mit einem roten Rahmen versehen. Immer wenn ein Hyperlink angeklickt wird soll eine AJAX-Anfrage die Datei artikel.json anfordern. Die angeforderte Datei steht in jetzt in Ihrem Skript als gesamtes Objekt zur Verfügung. Fügen Sie den gewünschten Nachrichtenartikel im div#artikel ein. Denken Sie daran eine Fehlermeldung (alert) auszugeben wenn der Ajax-Request einen Fehler verursacht.

```
<div id="artikel">
  <h2>Wichtiger Nachrichtenartikel 1</h2>
  <p>
    Nachrichtenartikel 1 Lorem ipsum dolor sit amet, consectetur adipisicing elit. Reprehenderit, m
    ab doloremque dolores perferendis vitae amet temporibus ut labore sint ex facilis harum quaerat
    possimus voluptates est debitis doloribus eius voluptatibus quo nostrum deserunt voluptatem tem
    eligendi itaque nihil sunt in exercitationem tempore quisquam eum consequatur non repellat....
  </p>
  
</div>
```



## Aufgabe 3 - Zufällige News & Themen

**Teilpunktzahl: 25 Punkte**

Nutzen Sie für jede Aufgabe die Vorlagendatei aufgabe3.html und newsthemen.json

Der Bereich #hier ist im Dokument mit einem roten Rahmen versehen.

Für den folgenden Arbeitsauftrag können Sie die bestehende HTML-Struktur im Dokument nutzen. Inhalte aus der externen Datei sind in die bestehenden Tags einzufügen.

Alle Ausgaben erfolgen in den rot markierten Bereich der Webseite.

```
<div id="inhalt" class="startseite">
<div class="teaser">
<section id="hier">
<!-- hier -->
```

Wenn das DOM der Webseite geladen ist, soll die Datei newsthemen.json über einen AJAX-Request angefordert werden. Die Datei liefert Ihnen ein Array mit Objekten auf den einzelnen Indizes. Bin Sie die Datei lib.js ein um auf die Funktion rand(min, max) zugreifen zu können. Immer wenn die Seite geladen wird ermitteln Sie eine zufällige Zahl mit der Sie auf einen zufällige Index des Arrays zuzugreifen. Die Zufallszahl muss modular sein, Sie soll vom ersten Index bis zum letzten Index erzeugt werden. Würde im Array ein weiteres Objekt mit Einträgen hinzugefügt soll sich die Zufallszahl automatisch anpassen. Nutzen Sie die Daten aus... ..der Eigenschaft h1 aus dem Objekt um die h1-Überschrift zu füllen. Nutzen Sie die Daten aus der Eigenschaft p0 aus dem Objekt um den ersten Absatz zu füllen. Nutzen Sie die Daten aus der Eigenschaft p1 aus dem Objekt um den zweiten Absatz zu füllen. Denken Sie daran auch den Hyperlink zu bearbeiten und dem passenden title zu füllen. Denken Sie daran eine Fehlermeldung (alert) auszugeben wenn der Ajax-Requeust einen Fehler verursacht.

## Aufgabe 4 - Formularvalidierung mit jQuery

**Teilpunktzahl: 25 Punkte**

Nutzen Sie für jede Aufgabe die Vorlagendatei aufgabe4.html-

Der Bereich #haupt-inhalt .hier ist im Dokument mit einem roten Rahmen versehen.

Vorname und Nachname: Muss ausgefüllt werden (required)

E-Mail: -Muss ausgefüllt werden (required)

Anzahl: -Muss ausgefüllt werden (required), Gültige Zahl

AGB: -Muss ausgefüllt werden (required)

Wenn das Formular korrekt ausgefüllt wurde, simulieren Sie den Versand mit einem alert.

Das Bild auf der folgenden Seite ist nur ein Beispiel, ihr HTML muss nicht vollkommen identisch sein.

## Formular im Inspektor des Browsers wenn noch nicht validiert wurde:

```
▼ <form method="get" action="">
  ▼ <p class="standard-zeile">
    <label for="first">* Vorname:</label>
    <input id="first" name="vorname">
  </p>
  ▼ <p class="standard-zeile">
    <label for="last">* Nachname:</label>
    <input id="last" name="nachname">
  </p>
  ▼ <p class="standard-zeile">
    <label for="mail">* E-Mail:</label>
    <input id="mail" name="mail">
  </p>
  ▼ <p class="standard-zeile">
    <label for="anzahl">* Anzahl Karten:</label>
    <input id="anzahl" name="anzahl">
  </p>
  ▼ <p class="box-zeile">
    *
    <input id="agb" name="agb" value="ja" type="checkbox">
    0
    <label for="agb">AGB gelesen?</label>
  </p>
  ▼ <p class="standard-zeile">
    <label for="memo">Anmerkung:</label>
    <textarea id="memo" name="kommentar"></textarea>
  </p>
  ▶ <p class="submit-zeile"></p>
```

## Formular im Inspektorfenster des Browsers wenn validiert wurde und alle Pflichtfelder nicht ausgefüllt wurden:

```
▼ <form method="get">
  ▼ <p class="standard-zeile">
    <label for="first">* Vorname:</label>
    <input id="first" class="errorInput" name="vorname" aria-required="true" aria-describedby="first-error" aria-invalid="true">
    <div id="first-error" class="error">Ist ein Pflichtfeld</div>
  </p>
  ▼ <p class="standard-zeile">
    <label for="last">* Nachname:</label>
    <input id="last" class="errorInput" name="nachname" aria-required="true" aria-describedby="last-error">
    <div id="last-error" class="error">Ist ein Pflichtfeld</div>
  </p>
  ▼ <p class="standard-zeile">
    <label for="mail">* E-Mail:</label>
    <input id="mail" class="errorInput" name="mail" aria-required="true" aria-describedby="mail-error">
    <div id="mail-error" class="error">Ist ein Pflichtfeld</div>
  </p>
  ▼ <p class="standard-zeile">
    <label for="anzahl">* Anzahl Karten:</label>
    <input id="anzahl" class="errorInput" name="anzahl" aria-required="true" aria-describedby="anzahl-error">
    <div id="anzahl-error" class="error">Ist ein Pflichtfeld</div>
  </p>
  ▼ <p class="box-zeile">
    *
    <input id="agb" class="errorInput" name="agb" value="ja" aria-required="true" aria-describedby="agb-error" type="checkbox">
    0
    <label for="agb">AGB gelesen?</label>
    <div id="agb-error" class="error">Ist ein Pflichtfeld</div>
  </p>
```

## Formular im Browser

### Ticketreservierung

Mit \* gekennzeichnete Felder sind Pflichtfelder

\* VORNAME:

Ist ein Pflichtfeld

\* NACHNAME:

Ist ein Pflichtfeld

\* E-MAIL:

Ist ein Pflichtfeld

\* ANZAHL KARTEN:

Ist ein Pflichtfeld

\* ☐ AGB GELESEN?

Ist ein Pflichtfeld

ANMERKUNG:

Jetzt absenden!