

## UE1 – HTML5, CSS3, WAI und JavaScript (25 Punkte)

Ziel dieses Übungsbeispiels ist die Erstellung von statischen, validen und barrierefreien Webseiten, sowie einfacher Scripts für client-seitige Funktionalität. Serverseitige Funktionalität muss in dieser Übung nicht berücksichtigt werden.

Deadline der Abgabe via TUWEL<sup>1</sup>: **Sonntag, 5. April 2015 23:55 Uhr**

**Nur ein Gruppenmitglied muss die Lösung auf TUWEL abgeben.**

### Jeopardy-Spiel

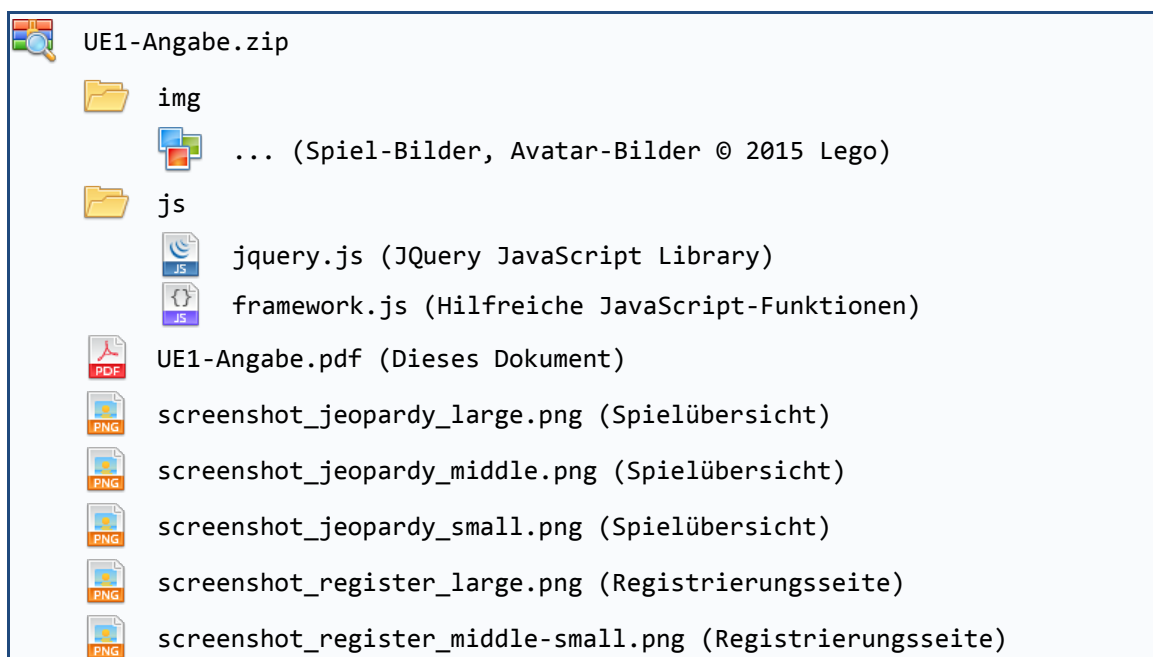
Bei „Jeopardy“ handelt es sich um ein Fragespiel, bei dem die KandidatInnen Fragen beantworten müssen, um Punkte zu sammeln. Jede Frage ist dabei einer Themenkategorie zugeordnet und hat einen bestimmten Wert, der den Schwierigkeitsgrad der Frage widerspiegelt. Eine Frage kann beliebig viele, aber mindestens zwei, Antwortmöglichkeiten haben, wobei mindestens eine dieser Antwortmöglichkeiten die korrekte Lösung auf die gestellte Frage ist („Multiple Choice“).

Ein Spiel besteht aus 10 Runden, wobei in jeder Runde eine Frage pro SpielerIn beantwortet wird. Der/Die SpielerIn (Mensch oder Computer) mit weniger Punkten darf zuerst eine Frage auswählen, bei Gleichstand wird der Mensch bevorzugt. Eine Frage gilt nur dann als richtig beantwortet, wenn alle korrekten Antwortmöglichkeiten ausgewählt und als Antwort an den Server übermittelt wurden. Zusätzlich gibt es bei jeder Frage ein Zeitlimit, bis zu dem diese Frage beantwortet werden muss. Ist dieses Zeitlimit abgelaufen, wird die aktuell ausgewählte Antwort gewertet.

Wurden von beiden SpielerInnen alle Fragen beantwortet, wird der Punktestand beider SpielerInnen ausgewertet und der/die GewinnerIn angezeigt. Herrscht Gleichstand, gewinnt der Computer.

### Angabe

Diese Angabe umfasst folgende Dateien:



Gegeben sind die Screenshots von den zwei Seiten, die unter Berücksichtigung von *Responsive Design* umzusetzen sind. Der Screenshot `screenshot_register_*.png` zeigt eine Seite mit einem Registrierungsformular („Registrierungsseite“) und `screenshot_jeopardy_*.png` zeigt das Aussehen des Jeopardy-Spiels vor der Fragenauswahl („Spielübersicht“).

Die Registrierungsseite ermöglicht es BenutzerInnen ihre persönlichen Daten sowie ihre Login-Daten einzugeben und an den Server zu übermitteln. Der Typ der Eingabefelder entspricht dabei den erwarteten Eingabewerten (z.B. Text für Vorname, Geschlecht als Option, Avatar als Auswahlmenü).

Die Spielübersicht zeigt den aktuellen Spielstand beider SpielerInnen, die bereits gewählten Fragen und die noch offenen Fragen an. Zum Spielstand gehören der bei der Registrierung gewählte Avatar, der Benutzername und das aktuelle Punktekonto. Zusätzlich wird angezeigt, wie viele Fragen bereits gestellt wurden (z.B. „2/10“). Mit Ausnahme der ersten Runde werden außerdem die letzten Punkteänderungen und ggf. Fragenauswahlen angezeigt.

In der Fragenübersicht werden alle Fragen geordnet nach ihrem Wert und zu ihrer Kategorie zugeordnet angezeigt. Pro Runde kann ein/e SpielerIn genau eine Frage auswählen. Die aktuelle Auswahl wird dabei farblich hervorgehoben (vgl. `screenshot_jeopardy_large.png`). Fragen, die bereits gewählt wurden, werden ausgegraut dargestellt und können nicht erneut ausgewählt werden. Mit einem Klick auf „wählen“ wird die ausgewählte Frage an den Server übermittelt werden.

Auf beiden Seiten befindet sich oben links ein Header mit dem entsprechenden Logo und dem Namen des Spiels („Jeopardy!“) und oben rechts die Navigation mit dem Link zum Anmelden bzw. Abmelden. Das Layout beider Seiten soll eine passende Mindest- und Maximalbreite haben und sich wie in den Screenshots angezeigt für verschiedene Bildschirmgrößen anpassen.

Implementieren Sie die beiden Seiten mit Hilfe von **XHTML5** und **CSS3** unter der Befolgung des **WAI Conformance Level Double-A** (oder höher). Beachten Sie daher, dass Sie die richtigen, semantischen Elemente in Ihrem XHTML5-Code auswählen. Achten Sie außerdem darauf, dass die Darstellung der Inhalte **ohne** Stylesheets (z.B. für Screen-Reader) im Sinne der WAI ebenso gewährleistet ist, um eine komfortable Verwendung für alle BenutzerInnen zu ermöglichen.

Das Aussehen aller Seiten ist so exakt wie möglich einzuhalten. Insbesondere gilt das auch für Texte, die in deutscher Sprache ausgewiesen sind. Verwenden Sie ausschließlich die Bilder, die in den Angaberessourcen (Ordner `img`) zur Verfügung gestellt wurden. Achten Sie auch vor allem darauf, dass die Ausgabe in den gängigsten Browsern (zumindest Mozilla Firefox 26, MS Internet Explorer 10 und Google Chrome 33) keine Unstimmigkeiten oder Probleme verursacht.

Entwickeln Sie darüber hinaus folgende Funktionalität mit JavaScript:

Die **folgenden Felder des Registrierungsformulars sollen mittels JavaScript während der Eingabe validiert werden**. Tritt ein Validierungsfehlers auf, so ist eine entsprechende, aussagekräftige Meldung bei dem entsprechenden Feld anzuzeigen. Beim ersten Aufruf der Seite sollen derartige Meldungen jedoch noch nicht angezeigt werden.

- *Vorname* (optionales Feld): Keine Einschränkungen.
- *Nachname* (optionales Feld): Keine Einschränkungen.

- *Geburtstag* (optionales Feld): Überprüfen Sie, ob es sich bei der Eingabe um ein gültiges Geburtsdatum handelt. Um Sie bei der Überprüfung zu unterstützen, stellt `framework.js` die Funktion `getNormalizedDateString` zur Verfügung. Dieser Funktion können Sie den CSS-Selektor auf das entsprechende Datumsfeld (bspw. `#birthdate`) übergeben und erhalten ein Datum im Format „dd.mm.yyyy“ zurück.
- *Benutzername* (verpflichtendes Feld): Mindestlänge von 4 Zeichen und Maximallänge von 8 Zeichen.
- *Passwort* (verpflichtendes Feld): Mindestlänge von 4 Zeichen und Maximallänge von 8 Zeichen.

Achten Sie darauf, dass der Benutzer die Daten nur dann an den Server übermitteln kann, wenn im Formular keinerlei Validierungsfehler enthalten sind. Das bedeutet, der „Registrieren“-Button darf nur aktiviert sein (Aussehen wie „wählen“-Button), wenn alle Pflichtfelder ausgefüllt (Benutzername und Passwort) ausgefüllt sind und das Formular keine Validierungsfehler enthält. Ansonsten ist der Button inaktiv und kann vom Benutzer nicht gedrückt werden (graue Darstellung, siehe Screenshot).

Verwenden Sie zur Validierung einerseits die Validierungsmöglichkeit von HTML5 bzw. eine eigene JavaScript-Implementierung falls diese Validierungsmöglichkeit nicht zur Verfügung steht. Um zu überprüfen, ob HTML5-Formularvalidierung möglich ist, können Sie die Funktion `hasFormValidation` aus `framework.js` verwenden. Bei der Überprüfung des Geburtsdatums müssen Sie zusätzlich zu `hasFormValidation` noch `hasNativeDateInput` aufrufen, da es Browser gibt, die zwar HTML5-Formularvalidierung ermöglichen, jedoch das Datumsfeld nicht unterstützen (zB Mozilla Firefox).

## Hinweise

### Validierung

Verwenden Sie zur Validierung Ihrer XHTML-Dateien den Validator <http://validator.nu/> und für Ihre CSS-Dateien den vom W3C zur Verfügung gestellten Validation-Service <http://jigsaw.w3.org/css-validator/>. Beachten Sie, dass der Typ „date“ für Eingabefelder derzeit nicht in allen Browsern unterstützt wird. Eine entsprechende Warnung bei der Validierung dürfen Sie in diesem Fall ignorieren. Stellen Sie sicher, dass Ihre XHTML-Dateien die Endung `.xhtml` haben, sonst kommt es zu Problemen mit dem Validator.

Zur Überprüfung der WAI-Tauglichkeit stehen Ihnen eine Vielzahl von Services im Internet zur Verfügung (z.B. AChecker, <http://achecker.ca/checker/index.php>). Nähere Infos dazu finden Sie in den Folien bzw. im TUWEL.

### Wahl der Elemente

Überlegen Sie sich eine geeignete Repräsentation der Fragen und Antworten, die der Bedeutung der verwendeten Elemente aus HTML gerecht werden. Beachten Sie dabei insbesondere die von HTML5 neu eingeführten Elemente.

Die Antworten sind keine Datentabelle (Element `table`), da die Spalten und Zeilen, abgesehen vom grafischen Layout, keinen wirklichen Zusammenhang haben. Antworten können entweder ausgewählt werden oder nicht und stehen in keiner geordneten Beziehung zu einander.

Tabellen zu Layout-Zwecken sind unter keinen Umständen erlaubt.

## Cascading Style Sheets (CSS)

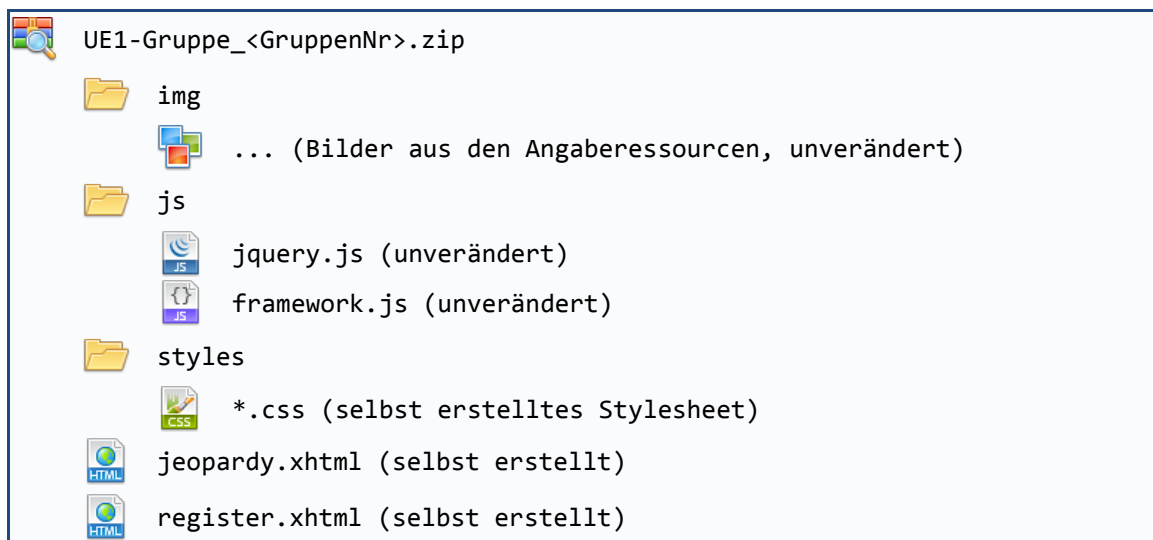
Der Einsatz von fremdem CSS-Code (z.B. `reset.css`) ist gestattet. Es muss jedoch klar ersichtlich sein, woher dieser Code kommt und dass dieser Code nicht von Ihnen stammt. CSS-Code, der die Übung vollständig löst (z.B. von Ihren Kollegen und Kolleginnen) darf selbstverständlich nicht verwendet werden.

## JavaScript

Zur Realisierung der Scripts darf nur jQuery<sup>1</sup> und die von uns bereitgestellte `framework.js` verwendet werden. Der Einsatz anderer JavaScript-Frameworks ist nicht gestattet. Eine Berücksichtigung der Barrierefreiheit der Scripts selbst wird für diese Übung nicht unbedingt gefordert, aber Sie sind eingeladen sich über diesen Aspekt selbstständig Gedanken zu machen und Lösungen zu realisieren.

## Abgabemodalität

Beachten Sie die allgemeinen Abgabemodalitäten des TUWEL-Kurses<sup>2</sup>. Zippen Sie Ihre Abgabe, sodass sie die folgende Struktur aufweist:



**Alle Dateien müssen UTF-8 codiert sein!**

**ACHTUNG:** Wird das Abgabeschema nicht eingehalten, so kann es zu Punkteabzügen kommen!

<sup>1</sup> <http://jquery.com/>

<sup>2</sup> <https://tuwel.tuwien.ac.at/course/view.php?id=5399>