

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. Server-seitige Funktionalität muss in dieser Übung nicht berücksichtigt werden.

Deadline der Abgabe via TUWEL¹: **Sonntag, 30. März 2014 23:55 Uhr**

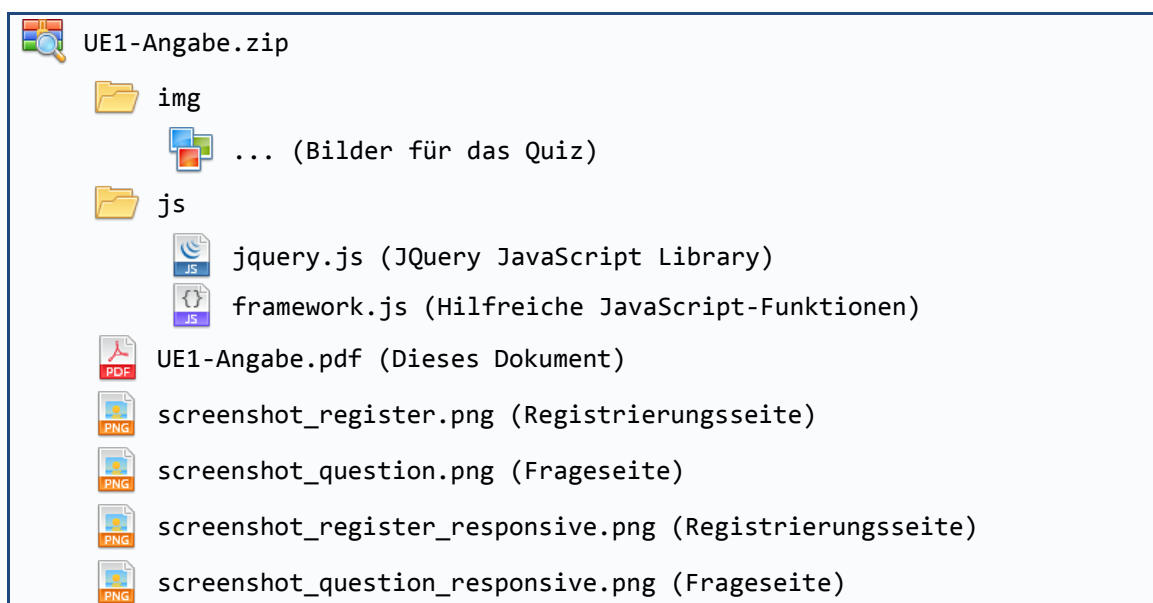
Quiz-Spiel

Beim "Quiz" handelt es sich um ein klassisches Fragespiel mit Mehrfachauswahl ("Multiple Choice"). Registrierte BenutzerInnen müssen während eines Spiels Fragen aus verschiedenen Kategorien beantworten und dabei gegen den Computergegner gewinnen.

Ein Spiel besteht aus fünf Runden und jede Runde aus drei Fragen, die alle einer bestimmten Kategorie zugewiesen sind. Eine Frage kann beliebig viele, aber mindestens zwei, Antwortmöglichkeiten haben, wobei mindestens eine dieser Antwortmöglichkeiten die korrekte Lösung auf die gestellte Frage ist. 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, gilt die Frage als falsch beantwortet und es wird ggf. die nächste Frage gestellt. Am Ende jeder Runde (= nach drei gestellten Fragen aus einer Kategorie), wird der/die RundensiegerIn ermittelt. Der/Die RundensiegerIn ist jene/jener SpielerIn (Mensch oder Computer), welche die meisten Fragen in dieser Runde richtig beantwortet hat. Bei Gleichstand entscheidet die Zeit, die benötigt wurde, um die Fragen zu beantworten (je weniger desto besser). Handelt es sich bei der Runde nicht um die letzte Runde, wird eine neue Runde mit einer zufällig gewählten Kategorie gestartet. Ist die letzte Runde abgeschlossen, wird der/die SpielsiegerIn durch die Anzahl der gewonnenen Runden ermittelt. Herrscht Gleichstand geht das gesamte Spiel unentschieden aus, ansonsten gewinnt der/die SpielerIn mit den meisten Rundensiegen.

Angabe

Diese Angabe umfasst folgende Dateien:



Gegeben sind u.a. zwei Screenshots, namens `screenshot_register.png` und `screenshot_question.png`. Der Screenshot `screenshot_register.png` zeigt eine Seite mit einem Registrierungsformular ("Registrierungsseite") und `screenshot_question.png` zeigt das Aussehen des Quiz-Spiels während eine Frage gestellt wird ("Frageseite").

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 (zB Text für Vorname, Geschlecht als Auswahllisten).

Die Frageseite zeigt den aktuellen Rundenstand beider SpielerInnen, die aktuelle Kategorie, die aktuelle Frage sowie die dazugehörigen Antwortmöglichkeiten. Ein ausgefüllter Kreis im Rundenstand deutet an, dass diese Frage richtig beantwortet wurde, ein Kreuz zeigt, dass diese Frage falsch beantwortet wurde und ein nicht ausgefüllter Kreis zeigt noch offene Fragen an. Antwortmöglichkeiten werden mit ausreichend Innenabstand und einer hellblauen Hintergrundfarbe dargestellt. Wird eine Antwortmöglichkeit ausgewählt, erhält sie eine dunkelblaue Hintergrundfarbe. Mit einem Klick auf "weiter" kann die Antwort (= Summe aller ausgewählten Antworten) an den Server übermittelt werden. Unterhalb der Antwortmöglichkeiten befinden sich die ablaufende Zeit sowie die Information über das zuletzt gespielte Spiel. Für diese statische Seite können Sie von einem Zeitlimit von 30 Sekunden ausgehen sowie, dass dies das erste Spiel ist ("Letztes Spiel: Nie").

Auf beiden Seiten befindet sich oben links ein Header mit dem entsprechenden Logo und dem Namen des Spiels ("Quiz"), und oben rechts die Navigation mit dem Link zum Anmelden. Der tatsächliche Inhalt beider Seiten soll eine passende Mindest- und Maximalbreite haben. Der Inhalt der Frageseite soll sich außerdem der Bildschirmbreite anpassen (etwa 80%). Erstellen Sie die Seiten außerdem mit Rücksicht auf "Responsive Design", in dem sie ab einer Bildschirmbreite von 32em das Design der beiden Seiten entsprechend den Screenshots `screenshot_register_responsive.png` und `screenshot_question_responsive.png` anpassen. Hierbei ist unter anderem darauf zu achten, dass auf der Frageseite alle Antworten einzeln untereinander dargestellt werden und dass die Formularfelder der Registrierungsseite nun die ganze Fensterbreite einnehmen.

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 auf der Frageseite dargestellte, ablaufende Zeit soll im Format "mm:ss" angezeigt und einmal pro Sekunde aktualisiert werden. Ist die Zeit abgelaufen, wird wieder von vorne begonnen (30 Sekunden).

Zu Ihrer Unterstützung stellt die Datei `framework.js` die Funktion `secToMMSS` zur Verfügung, welche Sekunden in das entsprechende Format konvertiert.

Außerdem sollen folgende Felder des Registrierungsformulars mittels JavaScript während der Eingabe validiert werden. Tritt ein Validierungsfehler 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 "weiter"-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.

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.

Für die ablaufende Zeit kommen zwei in HTML5 eingeführte Elemente in Frage (`meter` und `progress`). Überprüfen Sie, welches Element hier semantisch besser passen würde. Das `meter`-Element wird derzeit in Internet Explorer nicht unterstützt, was in diesem Fall jedoch ignoriert werden dürfte.

Cascading Style Sheets (CSS)

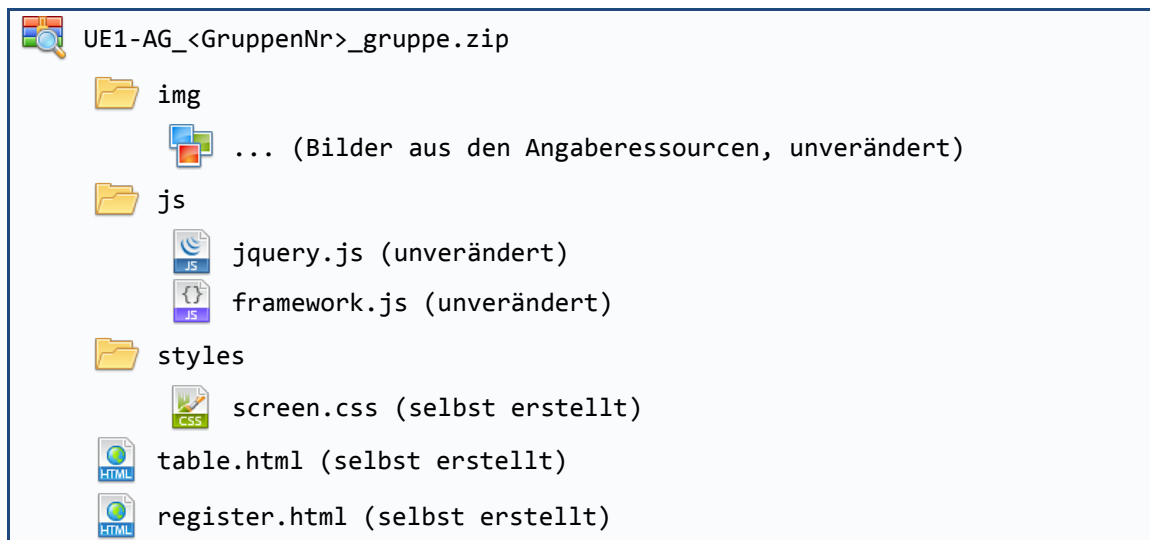
Der Einsatz von fremdem CSS-Code (zB `reset.css`) ist gestattet. Es muss jedoch klar ersichtlich sein, woher dieser Code kommt und dass dieser Code nicht von Ihnen stammt. CSS-Code, der die Übung vollständig löst (zB von Ihren Kollegen und Kolleginnen) darf selbstverständlich nicht verwendet werden.

JavaScript

Zur Realisierung der Scripts darf nur jQuery¹ 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-Kurses². 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!

¹ <http://jquery.com/>

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