

Übungsblatt 09

Thomas Samy Dafir

Alle Lösungen sind unter *buntmeise.cosy.sbg.ac.at* verfügbar

Aufgabe 1

Aufgabe 2

Lösung

Das Script wird einfach über ein script-Tag im HTML-head eingebunden. Die reine JS Lösung verwendet das 'onclick' Attribut. Darin wird ein JS Befehl angegeben, der aufgerufen wird, wenn das Element (hier Button) angeklickt wird.

In unserem Fall wird eine Funktion angegeben, die das zu verbergende Element über 'getElementById' findet und dann dessen 'display' Attribut auf 'none' setzt. Um das Element wieder einzublenden wird das 'display' Attribut auf 'block' zurückgesetzt.

Die jQuery Lösung verwendet einen JQuery selector nach id um einen Event-Listener auf die Buttons zu setzen. Werden diese dann angeklickt, wird das zu verbergende Element wieder über die id selektiert und die jQuery Funktion 'hide' bzw. 'show' aufgerufen. Diese tun nichts anderes als die display Eigenschaft zu setzen.

Lizenzen

jQuery unterliegt einer MIT Lizenz. Diese erlaubt unbeschränkte Benutzung, Veränderung und Verkauf. Die Lizenz muss jedoch in den Projekten immer angegeben werden. Außerdem gibt es Gewährleistung und es wird keine Haftung übernommen.

Bildlizenzen:

- Rights Managed: Gebühr pro Verwendung, abhängig vom Verwendungszweck
- Royalty Free: Gebühr abhängig von der Größe, einmalig. Mehrfache Verwendung
- Exclusive: Bild wird nur an einmal vergeben über eine gewisse Zeit / in einer gewissen geographischen Gegend.
- Non-Exclusive: Rechte mehrfach vergeben/verkauft.

- Creative Commons: 6 verschiedene Lizenzmodelle. Von: sämtliche Verwendung erlaubt (mit Angabe des Urhebers), bis: keine Veränderung, keine kommerzielle Verwendung.
Creative Commons basieren auf dem Prinzip, dass Fortschritt auf schon vorhandenem aufbaut.

Quelle: <http://shutha.org/node/600>

Aufgabe 3

Lösung:

Es wurde ein einfaches Formular erstellt und mit CSS gestyled. Validierung des Email Feldes erfolgt mit pure Javascript.

- Das Fformular verwendet Standard HTML Form Elemente, die teilweise in Block-Elemente verschachtelt wurden, um den Aufbau an die angabe anzupassen und einen Zeilenumbruch zu erreichen.
- Für sämtliche Bezeichnungen der wurden 'label' Tags verwendet. Diese enthalten ein 'for' Attribut, das angibt für welches Element (id) das Label gedacht ist. Label stellt eindeutig eine Zugehörigkeit her. Außerdem wird ins Feld gesprungen, wenn auf das label geklickt wird.
- ':focus' wurde verwendet, um das aktuell selektierte Eingabefeld zu formatieren (grüner Hintergrund).
- Zur Validierung wurde das Absenden des Formulars zu einer JS Funktion umgeleitet. Der 'submit' Button erhält ein 'onclick' Event, das eine Funktion aufruft, die die Email-Eingabe validiert (auf @ prüft). dazu wird zuerst die 'default-action' (Formular absenden) verhindert (preventDefault), dann der Wert des Email Feldes abgefragt, dieser auf '@' geprüft und dann entweder das Formular abgesendet oder nicht. Kommt kein '@' vor, wird das Feld mit einem roten Rahmen umgeben.

Aufgabe 4

Lösung wurde mit jQuery erstellt, Layout mit CSS.

Lösung

- Um Dropdown-Menüs aus- und einzublenden wurden die radio Buttons mit onclick events versehen. Beim Klick wird eine Funktion aufgerufen und die Ids des zu verbergenden und anzuzeigenden Elements übergeben. Diese werden dann mittels *style.display* ein bzw. ausgeblendet.
- Um Bilder zu wechseln, wurde ein 'change' eventListener auf 'select' gesetzt. Die aufgerufene Funktion liest den selektierten Wert aus dem 'option' Tag und setzt mit jQuery das 'src' Attribut des 'img' Tags.
- Um hier Injection zu verhindern werden die ausgelesenen 'option' Werte immer escaped.
- Wechselt man das angezeigte Dropdown Menü, wird immer das zuletzt ausgewählte Item des einzublendenden Menüs ausgelesen und das passende Bild angezeigt.

Web Applikation Projekt

Es soll eine Quiz-Web-App mit folgenden Features zu erstellen:

- Responsive Web App
- Umsetzung mit HTML5, CSS, python, JS, jQuery.
- Keine statischen Fragen. Fragen sollen per API von OpenTrivia bezogen werden.
- Die App soll Optionen, die OpenTrivia erlaubt verwenden: Schwierigkeit, Thema.

Ablauf (User Sicht):

1. User besucht die Seite.
2. Wählt Schwierigkeit, Thema, Anzahl an Fragen.
3. User beantwortet Fragen.
4. Nach dem ersten Einloggen einer Antwort hat der User nur mehr x Sekunden, um sich anders zu entscheiden.
5. Antwort wird gespeichert.

6. Am Ende erhält der User seine Ergebnisse.

Ablauf (Programm Sicht):

1. Start View wird an User gesendet.
2. Eingaben (Schwierigkeit, Thema) von User empfangen.
3. Anfrage an OpenTrivia. Fragen und Antworten serverseitig abspeichern.
4. Fragen und Antwortmöglichkeiten generieren und an User senden.
5. Antworten lokal (beim Benutzer) sammeln und abschließend an den Server senden.
6. Auswertung der Antworten.
7. Senden der Ergebnisse an den User (evtl. mit Lösungen).
8. Löschen der Session Daten.