

## **Hausaufgabe 5 – HTML-Formulare, Javascript und DOM**

### ***Aufgabenstellung:***

Es soll mittels HTML, Javascript und DOM ein vereinfachter Terminkalender erstellt werden, der das Anlegen von neuen Terminen ermöglicht. Termine werden angelegt, indem die Uhrzeit des Termins eingegeben wird und weitere Teilnehmer zu dem Termin aus dem Adressbuch ausgewählt werden. Anschließend wird auf ein Datum im Kalender (eine Tabelle) geklickt, wodurch der neue Termin angelegt wird. Alle angelegten Termine werden in eine weitere Tabelle geschrieben. Alle Tage im Kalender, denen ein Termin zugeordnet wurde, werden farblich hervorgehoben.

### **Hilfestellung:**

1. Erstellen Sie eine Webseite HTML-Seite *calendar.html*, und darin ein Formular zur Eingabe der Termine. Nach Absenden des Formulars soll eine weitere HTML-Seite *success.html* aufgerufen werden. *Success.html* soll lediglich einen ganz kurzen Text beinhalten, dass die Termine erfolgreich angelegt wurden. Die folgenden Schritte beziehen sich auf *calendar.html*.
2. Erstellen Sie ein einfaches Eingabefeld für die Uhrzeit und eine Javascript-Funktion `getTimeInput()`, die den eingegebenen Wert ausliest. Die Funktion `getTimeInput()` liefert die Uhrzeit als Rückgabewert.
3. Erstellen Sie für das Adressbuch eine Auswahlliste mit 10 Namen. Es soll möglich sein, gleichzeitig mehrere Namen auszuwählen. Definieren Sie eine Javascript-Funktion `getSelectedNames()`, die einen String zurückgibt, der aus den selektierten Namen zusammengesetzt ist.
4. Erstellen Sie eine Tabelle mit Rahmen und Hintergrundfarbe für den Kalender. Es genügt, wenn er die Tage einer Woche enthält. Geben Sie jeder Tabellenzelle eine ID, die sich aus einem Bezeichner und dem Datum zusammensetzt (z.B. *calendar5* für den 5. Tag). Dann können Sie später aus Javascript mittels `getElementById("calendar" + date)` auf die Tabellenzelle zugreifen. Fügen Sie weiterhin jeder Tabellenzelle das Attribut *onClick* hinzu und rufen Sie dort die Funktion `createAppointment()` auf, wobei Sie als Parameter *x* das Datum, auf das geklickt wurde, übergeben.
5. Erstellen Sie die Javascript-Funktion `createAppointment(date)`, die als Parameter *date* das Datum erhält, auf das geklickt wurde. In dieser Funktion soll der Tag, auf den geklickt wurde, markiert werden (d.h. die entsprechende Tabellenzelle soll eine andere Hintergrundfarbe erhalten). Verwenden Sie dazu die Möglichkeiten von DOM.
6. Erstellen Sie eine zunächst leere Tabelle, in die per Javascript die Termine eingetragen werden sollen. Damit die Webseite auch in allen Browsern funktioniert, benötigt die Tabelle auch einen Tabellenkörper (table body). Geben Sie dem Tabellenkörper eine ID, damit Sie ihm mittels Javascript Tabellenzeilen hinzufügen können. Ergänzen Sie nun die Funktion `createAppointment(date)`, so dass mittels DOM die Termine in die Tabelle eingetragen werden. Jede Tabellenzeile soll einen Termin enthalten, wobei es jeweils eine Tabellenzelle für das Datum, die Uhrzeit und für die Teilnehmer des Termins geben soll. D.h. die Javascript-Funktion muss die entsprechenden Knoten erstellen (für Zeile, Zellen und den darin enthaltenen Text) und jeweils dem Elternknoten hinzufügen.

Für den einzutragenden Text werden die oben definierten Funktionen `getTimeInput()` und `getSelectedNames()` verwendet.

### **Optionale Zusatzaufgaben:**

- Überprüfen Sie der Funktion `getTimeInput()` das eingegebene Datum auf Gültigkeit (Format *hh:mm*). Falls die Eingabe leer oder ungültig ist, soll ein Eingabedialog angezeigt werden, in den der Benutzer eine gültige Uhrzeit eintragen muss.
- Fügen Sie die Termine in der Liste nach Datum sortiert ein.
- Fügen Sie die Termine zusätzlich nach Uhrzeit sortiert ein.
- Ermöglichen Sie, dass die Teilnehmer eines Termins von einem anderen Termin übernommen werden können, indem man sie in der Tabelle markiert. D.h. immer wenn der Benutzer in der Tabelle mit den Terminen Text markiert, liefert die Javascript-Funktion `getSelectedNames()` (anstatt der Auswahl im Adressbuch) diesen markierten Text zurück. (Hinweis: verwenden Sie dazu eine Methode des Objektes *document*).

### **Weitere Hinweise:**

Sie finden alle benötigten Javascript-Funktionen sowie Code-Beispiele unter SelfHTML:

- Formulare unter HTML
- Vordefinierte Javascript-Funktionen zum Zugriff auf DOM finden Sie unter der Objektreferenz im Kapitel Javascript. Sie benötigen hauptsächlich die Funktionen der Objekte *document* und *node*.
- Die Auswirkung von eingefügtem Javascript-Code lassen sich oft am einfachsten testen, indem man sich das Ergebnis einer Anweisung oder einer Funktion einfach testweise in einem Dialogfenster ausgeben lässt.

### ***Abgabe:***

- Die Lösung ist als ein **Zip-Archiv** einzusenden das die beiden HTML-Dateien *calendar.html* und *success.html* enthält.
- Das Archiv muss mit **vorname.nachname.zip** benannt werden.
- Es werden nur individuelle Lösungen angenommen.
- Senden Sie eine Email mit dem Zip-Archiv als Anhang an **andreas.pleuss@ifi.lmu.de** bis **8. Februar 2006**.