

Übungsblatt 1

0 Getting Started

Um Webapplikationen zu entwickeln benötigen Sie eigentlich nur einen Text-Editor und einen Browser zum Testen des Codes. Welchen Editor und Browser Sie verwenden ist Ihnen freigestellt, als Einstieg ist jedoch “Opera” als Browser und “Sublime Text” als Editor eine gute Wahl.

Da Opera nicht auf den CIP-Pool-Rechnern installiert ist, müssten Sie daher eine portable Version (z.B. von <http://portableapps.com>) herunterladen, installieren und den Ordner in ihr Home-Verzeichnis schieben (damit Sie die Prozedur nicht jeden Tag wiederholen müssen). Auch von Sublime Text gibt es eine portable Version auf deren Homepage.

1 Das Grundgerüst*

Erstellen Sie das Grundgerüst einer Webseite mit allen (benötigten) Elementen einschließlich eines Absatzes (der Tag dafür ist `<p>`), der ein wenig Text und ein Bild, das durch den ``-Tag angezeigt wird, enthält.

2 Es lebe die Struktur*

Verwenden Sie die Webseite aus Übungsaufgabe 1. Erstellen Sie anstelle des Textabsatzes auf der Seite einen Artikel, der jeweils einen Kopf- und einen Fußbereich besitzt. Fügen Sie dem Artikel zwei Textabsätze hinzu.

Im ersten Textabsatz sollten Sie noch ein wenig Text mit hervorgehobenen (``) Wörtern, kursiven (`<i>`) Ausdrücken und fett (``) dargestellten Zahlen geschrieben werden. Im zweiten Absatz sollten Informationen zum Autor (`<address>`) und zur letzten Erstellungszeit (`<time>`) der Seite angezeigt werden.

3 Weitere Funktionalitäten hinzufügen*

Ergänzen Sie nun die Webseite aus Aufgabe 2 um zwei eigenständige Sektionen. Diese sollen jeweils Überschriften der zweiten Ebene sowie (beliebige) Randinformationen besitzen.

Verwenden Sie für die Randinformationen der ersten Sektion eine nicht nummerierte Liste (``) mit entsprechenden Einträgen. Für die zweite Sektion sollten Sie eine

kleine Tabelle mit drei Spalten, vier Zeilen und entsprechenden Kopf- und Fußzeilen (`<thead>` und `<tfoot>`) einfügen.

4 Fakultätsfunktion*

Schreiben Sie eine Funktion namens `faculty()` die ein Argument `n` erhalten soll. Das Argument soll als (ganze) Zahl behandelt werden. Als Rückgabewert der Funktion soll $n!$ berechnet werden. Die genaue Formel hierzu lautet

$$n! = \prod_{i=1}^n i = n \cdot (n - 1) \cdot (n - 2) \cdots 1. \quad (1)$$

Die Ausgabe soll über die eingebaute Funktion `alert()` erfolgen. Die Eingabe kann über die eingebaute Methode `prompt()` geschehen.

5 Zahlen sortieren*

Über eine `while`-Schleife soll solange ein `prompt()` angezeigt werden, wie der Benutzer eine gültige Zahl eingibt (d.h. Abbruch der Schleife bei ungültiger Eingabe). Die Zahlen sollen in einem Array `zahlen` gespeichert werden. Nach dem Abbruch der `while`-Schleife soll das eingegebene Array `zahlen` aufsteigend sortiert werden. Abschließend soll das Array über `alert()` ausgegeben werden. Benutzen Sie hierzu `zahlen.toString()`.