

### Exercise 3.1 Zeichen-Kodierung

- a) Erstellen Sie eine Webseite, die Text mit Umlauten enthält. Die Zeichen-Kodierung soll in einer Version der Webseite ISO 8859-1, in einer weiteren UTF-8 sein.
- b) Konfigurieren Sie Apache so, dass beide o.g. Kodierungen als Default-Einstellung gelten und versuch Sie auf die in a) erstellen Webseiten zuzugreifen. Welches Verhalten können Sie beobachten?

### Exercise 3.2 Tabellen

- a) Erstellen Sie folgende Tabellen in einer XHTML-Seite:
  - i. Mit Spaltenüberschriften
  - ii. Mit einer Beschriftung ("caption")
  - iii. Verwenden Sie eine Tabelle zur Aufteilung einer XHTML-Seite und zur Text-/Grafik-Positionierung.  
Welche Vor- und Nachteile gibt es hierbei?

### Exercise 3.3 Formulare

- a) Erstellen Sie ein Formular mit den folgenden Elementen: Texteingabefeld, Passworteingabefeld, Labels, Radio Buttons, Checkbox, Listbox, Hidden Field, Buttons, Image.
- b) Erstellen Sie ein Formular/mehrere Formulare für die von Ihnen geplante Webseite.

### Exercise 3.4 Content-Einbettung

- a) Erstellen Sie eine Webseite und betten Sie ein Youtube-Video in diese ein.
- b) Suchen Sie im Internet nach einer kostenlosen MIDI-Datei und betten Sie diese in eine Seite ein. Was bedeutet der Begriff MIDI und wie funktioniert dieses Konzept?
- c) Suchen Sie ein Java Applet im Internet und betten Sie dieses in eine XHTML-Seite Ihrer Wahl ein.

### Exercise 3.5 Frames/IFrames

Verwenden Sie SelfHTML zur Bearbeitung dieser Aufgabe.

*Web Engineering I (TIIT2101) – Prof. Dr. Holger D. Hofmann – Praktikum*

- a) Was versteht man unter einem Frameset und einem Frame?
- b) Was versteht man unter einem IFrame?
- c) Erstellen Sie ein Frameset mit drei Frames. Platzieren Sie in einem Frame Hyperlinks, die den Inhalt eines anderen Frames ändern, also z.B. ein Frame dient als Inhaltsverzeichnis und ein anderer Frame als Inhaltsseite.
- d) Erstellen Sie eine HTML-Seite und betten Sie einen IFrame in diese ein. Für welche Anwendung könnten Sie dies benutzen?

### Exercise 3.6 CSS

- a) Erstellen Sie für ein vor Ihnen erstellte XHTML-Seite eine CSS-Datei screen.css (Darstellung für Bildschirmausgabe) und eine print.css (Druckausgabe). Experimentieren Sie mir verschiedenen Layouts bzw. der Colorierung von Seiten-Elementen.  
Wenden Sie die Formatierungen per **Tag-Namen, Wildcard, Klasse, Sub-Element einer Klasse, Pseudo-Klasse und ID** an.  
Tipp: Verwenden Sie im Head-Bereich der XHTML-Seite das Attribut "media" des link-Tags.

### Exercise 3.7 CSS-Positionierung

- a) Erstellen Sie eine CSS-Datei position.css für eine XHTML-Datei Ihrer Wahl. Diese soll verschiedene Elemente (paragraphs, images, div-Elemente) auf einer Seite positionieren.
- b) Erstellen Sie eine CSS-Datei boxlayout.css und positionieren Sie auf einer XHTML-Seite div-Element so, dass Sie ein mehrspaltiges Layout (ähnliche Frameset) erhalten.  
Schauen Sie sich die Lösung hierfür unter [3].  
Optional: Realisieren Sie die einzelnen Elemente als Boxen mit abgerundeten Ecken (Tipp: Verwenden Sie [2] zur Erstellung der rounded boxes)
- c) Realisieren Sie ein Stapel von Elementen Ihrer XHTML-Seite durch Verwendung des Z-Index.

### Online-Ressourcen

- [1] Selft-HTML (<http://de.selfhtml.org>)
- [2] RoundedCornr ([www.roundedcornr.com](http://www.roundedcornr.com))
- [3] [www.yaml.de](http://www.yaml.de)