

## Praktische Arbeit Webtechniken / Web-Gestaltung 1 im WS 22/23

Prüfung zur Lehrveranstaltung Nr. 6984 im LSF.

**Ausgabe:** 27.01.2023, 08:00 Uhr

**Abgabe:** bis spätestens **10.02.2023** um **12:00 Uhr**, sowohl als ZIP-Datei auf Moodle als auch auf dem Webserver, siehe Hinweise zur Abgabe

**Bewertung:** Es können bis zu 100 Punkte erreicht werden. Zum **Bestehen** sind 50 Punkte erforderlich.

Ihre Aufgabe ist es, mit der Realisierung einer Website die unten folgenden Aufgaben umzusetzen. Das Thema für den Inhalt Ihrer Website dürfen Sie ansonsten frei bestimmen, es kann beispielsweise das Thema Ihres Semesterprojekts sein.

Lesen Sie zuerst alle Aufgaben und Hinweise vollständig, bevor Sie mit der Umsetzung beginnen.

### 1. Aufgabe: Aufsetzen der Website

Setzen Sie eine Website mit Webseiten auf. Folgende Grundregeln gelten für die Website:

- Sie besteht mindestens aus einer HTML-Startseite mit Namen `index.html` sowie 3 weiteren HTML-Dateien.
- Alle HTML-Dateien verwenden aktuelles HTML (mindestens HTML 5) und das entsprechende Grundgerüst.
- Die Zeichenkodierung für alle HTML-Dateien ist UTF-8.
- Die Dateien der Website sind in einem Projektverzeichnis, bei dem alle HTML-Dateien mit Ausnahme der Datei `index.html` in einem Unterordner „sites“, alle CSS-Dateien in einem Unterordner „styles“ und alle Bilder in einem Unterordner „images“ liegen. Die Datei `index.html` liegt direkt im Projektverzeichnis.

### 2. Aufgabe: Inhaltliche Grundstruktur der Seiten

Alle HTML-Seiten sollen eine inhaltliche Grundstruktur nach den folgenden Regeln besitzen:

- Jede HTML-Seite enthält mindestens folgende Elemente:
  - `article`, `nav`, `aside`, `footer` jeweils mindestens einmal,
  - `header` mindestens zweimal,
  - `section` mindestens dreimal.
- Das `body`-Element hat als direkte Kinder genau folgende Elemente in beliebiger Reihenfolge: `article`, `nav`, `header`, `footer`. Dabei dürfen `article`, `nav` und `footer` nur als direkte Kinder von `body` auftreten, und nur eines der mindestens zwei `header` darf als direktes Kind von `body` auftreten. Zusätzlich zu den geschilderten Kindern darf `body` als direkte Kinder nur `script`-Elemente enthalten (vgl. Aufgabe 14).
- `article` darf als direkte Kinder nur folgende Elemente besitzen: `aside`, `section`, `img`. Dabei dürfen `aside` und `section` nur als direkte Kinder von `article` auftreten.
- `aside` soll immer ein `header`-Element enthalten.
- `nav` darf als direkte Kinder nur folgende Elemente besitzen: Eine nicht nummerierte Liste. Als indirekte Kinder („Enkel“) von `nav` dürfen nur Listen, Listen-Elemente und Anchor-Elemente vorkommen.

### 3. Aufgabe: Links und Navigation

Setzen Sie Links innerhalb der Website und von der Website auf externe Quellen nach den folgenden Regeln:

- Alle HTML-Dateien der Website lassen sich von jeder Seite der Website aus per Link direkt erreichen.

- Setzen Sie einen Link von einer der Seiten auf ein beliebiges *Element* einer beliebigen anderen Seite von Ihnen.
- Setzen Sie zwei Links auf verschiedene externe Adressen.
- Verwenden Sie für *alle* Links innerhalb Ihrer eigenen Website nur relative Pfade, keine absoluten.

#### 4. Aufgabe: Bilder einbinden

Binden Sie Bilder in Ihre Website ein nach folgenden Kriterien:

- Jede HTML-Seite enthält mindestens ein Bild (insgesamt für die Website also mindestens 4 verschiedene Bilder aus dieser Aufgabe), welches kein über CSS eingebundenes Hintergrundbild ist. Ein Bild ist Kind eines `aside`-Elements, zwei Bilder jeweils Kinder eines `section`-Elements (Kinder können direkte und indirekte Kinder sein).
- Jedes dieser Bilder enthält einen Alternativtext.
- Mindestens drei der Bilder sind ein klickbarer Link.
- Jede Bilddatei in der Website muss in eine HTML-Datei eingebunden sein.

#### 5. Aufgabe: Allgemeine HTML-Elemente

Wählen Sie aus der Gruppe der in der Lehrveranstaltung demonstrierten HTML-Elemente zur logischen Auszeichnung von Text ein Element aus. Verwenden Sie dieses Element mindestens zweimal. Verwenden Sie außerdem mindestens ein Textabsatzelement und verwenden Sie Überschriften auf Stufe 1 und 2 jeweils mindestens zweimal.

Kennzeichnen Sie außerdem eine Textstelle so, dass der Browser erkennt: Der Text wurde zuerst eingefügt und dann gelöscht.

Setzen Sie außerdem eine HTML-Tabelle ein, die 2 Spalten und 4 Zeilen umfasst. Die erste Zeile enthält die Tabellenüberschrift. Verbinden Sie außerdem die Zellen der ersten Zeile. Die Tabelle soll mit einem Namen versehen (also benannt) sein.

#### 6. Aufgabe: Listen

Setzen Sie eine Beschreibungsliste ein, welche die folgenden Kriterien erfüllt:

- Die Liste enthält mindestens drei zu definierende Begriffe mit jeweils einer zugehörigen Beschreibung.
- Innerhalb der Beschreibungsliste setzen Sie eine ungeordnete Liste ein, die drei Listeneinträge umfasst.
- Innerhalb der ungeordneten Liste setzen Sie eine geordnete Liste ein, dessen Listentyp die (unsere üblichen lateinischen) Kleinbuchstaben sind.

#### 7. Aufgabe: Basislayout

Definieren Sie Basiseinstellungen für das Layout Ihrer Website. Berücksichtigen Sie dabei für diese und alle anderen Aufgaben, die CSS verwenden:

- Für alle CSS-Regeln soll die Website nur eine einzige, zentrale CSS-Datei verwenden. Es darf keine weitere CSS-Datei geben, egal ob eingebunden oder nicht.
- Die Verwendung des HTML-Elements `style` ist nicht zugelassen.
- Die Verwendung eines `style`-Attributes in HTML-Elementen ist nicht zugelassen.
- Selektoren in den CSS-Regeln dürfen nicht nur aus dem Namen von HTML-Elementen bestehen. Sie müssen neben dem Namen von HTML-Elementen auch weitere Selektionskriterien einsetzen. Das gilt auch für Strukturelektionen: Die einzelnen Strukturelemente dürfen also nicht nur über den Namen von HTML-Elementen selektiert werden.

Legen Sie dann folgendes Layout fest:

- Stellen Sie für alle Überschrift-Elemente auf Stufe 1 ein: Eigene Schriftarten mit mindestens einer Rückfallmöglichkeit, Schriftstil normal, sowie explizit eingestellt, aber frei gewählte: Schriftgröße, minimale Zeilenhöhe sowie Schriftgewicht.
- Stellen Sie für alle Absatz-Elemente eine eigene Schriftfarbe ein.
- Alle Kinder von `article`-Elementen, aber nicht das `article`-Element selber, soll eine eigene Hintergrundfarbe bekommen.
- Stellen Sie die horizontale Text-Ausrichtung für alle `aside`-Elemente auf zentriert.

## 8. Aufgabe: Layout für Bilder

Gestalten Sie das Layout für die Bilder aus Aufgabe 4:

- Alle Bilder sind ein `float`. Drei sollen auf der linken Seite, eines auf der rechten Seite angezeigt werden.
- Legen Sie die Breite oder die Höhe aller Bilder über CSS-Regeln (und nur über CSS-Regeln) auf Werte fest, die Sie selber aussuchen.

## 9. Aufgabe: Box-Modell

Setzen Sie folgendes um:

- Stellen Sie den Außenabstand für alle HTML-Absatz-Elemente an jeweils allen vier Seiten auf Werte Ihrer Wahl ein.
- Stellen Sie `padding` für `footer` an den Seiten oben und unten auf Werte Ihrer Wahl ein.
- Passen Sie alle HTML-Überschrift-Elemente `h2` so an, dass sie am unteren Rand einen Außenabstand von 5px haben.

## 10. Aufgabe: Positionierung mit CSS

Gestalten Sie das Layout aller Ihrer HTML-Seiten auf folgender Basis ohne Grid-Layout:

- Erreichen Sie, dass das direkte `header`-Kind von `body` oberhalb der Elemente `article`, `nav` und `footer` liegt und 100% der gesamten Breite der Webseite einnimmt.
- Die Navigation mit `nav` liegt am linken Seitenrand und hat eine Breite von 1/5 der Seitenbreite.
- Das `article`-Element liegt am rechten Seitenrand neben dem `nav`-Elementes und hat eine Breite von 4/5 der Seitenbreite.
- Erreichen Sie, dass das direkte `footer`-Kind von `body` unterhalb der Elemente `article`, `header` und `nav` liegt und 100% der Breite der Webseite einnimmt.
- Erreichen Sie, dass das `header`-Kind der `aside`-Elemente oberhalb aller anderen Kinder von `aside` liegt, unabhängig davon, welche oder wie viele andere Inhalte das `aside`-Elemente hat, und die gesamte Breite von `aside` einnimmt.

Setzen Sie nun das grid-Layout (nach W3C-Standard) ein, um folgendes zu erreichen:

- Die `section`-Elemente liegen am linken Rand ihres Eltern-Elements und haben eine Breite von 3/4 ihres Elternelements.
- Die `aside`-Elemente liegt am rechten Seitenrand und haben eine Breite von 1/4 ihres Elternelements. Die `aside`-Elemente schöpfen die volle Höhe ihres Eltern-Elementes aus.

## 11. Aufgabe: Pseudoelement

Es gibt mehr CSS-Pseudoelemente, als Sie in der Lehrveranstaltung kennengelernt haben. Recherchieren Sie das Ihnen aus der Lehrveranstaltung bisher unbekannte Pseudoelement, mit dem Sie folgendes Ziel erreichen: Es soll eine Überschrift der Stufe 1 geben, welches das Klassen-Attribut `ErstesZeichen` erhält. Über nur *eine* CSS-Regel soll immer der erste Buchstabe mit einer Schriftgröße von 150% angezeigt werden.

## 12. Aufgabe: Menü

Realisieren Sie in allen HTML-Seiten innerhalb des `nav`-Elements ein Navigationsmenü für Ihre Website, mit dem man innerhalb Ihrer Website alle Unterseiten aufrufen kann. Setzen Sie dafür im `nav`-Element als HTML-Elemente nur HTML-Listen und Anchor-Elemente ein. Die Menüoptionen sollen von oben nach unten angezeigt werden. Die Listeneinträge sollen eine einheitliche Größe von Breite 9,5% Viewport-Breite und Höhe 5% Viewport-Höhe haben und ohne Abstand am linken Rand des `nav`-Elements liegen. Die Hintergrundfarbe der Menüoption soll sich verändern, wenn der Mauszeiger über der Menüoption liegt. Die Links im Menü sollen nicht unterstrichen dargestellt werden. Es sollen keine Aufzählungssymbole oder Nummerierungen für die HTML-Listen erkennbar sein.

## 13. Aufgabe: Formulare

Integrieren Sie in Ihre Seite `index.html` ein HTML-Formular mit einem Button als einzigem Kind, der als Beschriftung „Anmeldung“ anzeigt.

## 14. Aufgabe: JavaScript

Setzen Sie folgende Schritte der Reihe nach unter Beachtung der Struktur-Vorgaben aus den vorherigen Aufgaben um:

1. Integrieren Sie ein JavaScript-Programm in Ihre Seite `index.html` (direkt und nicht ausgelagert in eine eigene Datei), dass es dem Benutzer erlaubt, einen Namen einzugeben und diesen in einer Variable zu speichern. Dieses Programm erweitern wir in den nächsten Schritten.
2. Prüfen Sie, ob der Name „Mustermann“ eingegeben wurde. Falls das so ist (und nur dann), soll das Programm die Eingabe noch einmal mit dem Hinweis „Bitte gib Deinen echten Namen ein“ durchführen und das Ergebnis der erneuten Eingabe in derselben Variable wie zuvor speichern.
3. Anschließend soll ein Fenster erscheinen, welches den Benutzer mit dem eingegebenen Namen begrüßt und durch einen Klick auf OK schließbar ist.
4. Erweitern Sie das Programm so, dass jetzt eine Funktion das bisherige JavaScript-Programm aus den Punkten 1-3 enthält.
5. Erweitern Sie das Formular aus Aufgabe 13 so, dass beim Klick auf den Button aus Aufgabe 13 die Funktion aus Punkt 4 aufgerufen wird.

## 15. Abgabe und abschließende Hinweise

Die Abgabe der erstellten Website erfolgt doppelt:

- Einmal packen Sie das Projektverzeichnis *inklusive aller Unterverzeichnisse* in ein ZIP-Archiv, dass Sie an der dafür vorgesehenen Stelle im **Moodle**-Kurs (neues Moodle-System) hochladen.
  - Außerdem übertragen Sie alle Dateien Ihrer Website auf den **Webserver**. Jeder Teilnehmer hat dazu ein Unterverzeichnis **praktarb** in seinem Benutzerverzeichnis erhalten, in das er den Inhalt des Projektverzeichnisses mit allen Unterverzeichnissen kopiert (ohne das Projektverzeichnis selbst, denn **praktarb** ist das Projektverzeichnis auf dem Webserver).
- Wichtiger Hinweis: Das Verzeichnis kann bis zum Abgabetermin nur vom Systemverwalter und dem jeweiligen