

## Übungsblatt 2 WT:II+III

Bis zum 11.05.2022, 23:59, sind Lösungen zu folgenden Aufgaben abzugeben: 1, 2, 3a,c, 4a,b, 5, 6.

### Aufgabe 1 : TLS (1+2 Punkte)

- (a) Besuchen Sie <https://webtec.webis.de> und lassen Sie sich in Ihrem Browser das zur Website gehörige TLS-Zertifikat anzeigen. Finden Sie folgende Informationen heraus:
  - (a1) Welche Certificate-Authorities (CAs) signieren welche Zertifikate in der Zertifikats-Kette?
  - (a2) Welche Domains werden durch das ausgelieferte Server-Zertifikat authentisiert?
- (b) Wenn Sie <https://badtls.webtec.webis.de> besuchen, zeigt Ihr Browser Ihnen eine Sicherheits-Warnung an. Erklären Sie die Bedeutung dieser Warnung und finden Sie zwei Gründe, warum für diese Domain Probleme auftreten.

### Aufgabe 2 : HTML: Validierung (2 Punkte)

Forken sie das [Repository](#) dieser Abgabe und öffnen sie die Datei non-valid-html5.html.

Korrigieren Sie mit möglichst wenigen Änderungen diese Datei bedeutungserhaltend, so dass ein valides HTML5-Dokument entsteht.

Hinweis: Sie können die Validität des Dokumentes auf <https://validator.w3.org/> überprüfen lassen.

### Aufgabe 3 : HTML: Inline-/Block-Elemente, physische/logische Auszeichnung (1+0+1+0 Punkte)

- (a) Wie unterscheiden sich HTML-Block-Elementen von HTML-Inline-Elementen in der Browser-Darstellung bezüglich des Textflusses?
- (b) Was bewirken die HTML-Elemente `<div>` und `<span>`?
- (c) Was ist der Unterschied zwischen physischen und logischen Auszeichnungen?
- (d) Handelt es sich bei folgendem HTML-Code um eine physische oder eine logische Auszeichnung? Begründen Sie Ihre Antwort.

```
<span class="review-content">Tolles Produkt, fünf Sterne!</span>
```

### Aufgabe 4 : CSS: Regeln erläutern (1+3+0 Punkte)

- (a) Skizzieren Sie die Struktur des folgendem HTML-Ausschnitts als Baum (siehe DOM):

```
<main><h1>Über<em>schrift</em></h1><p>Text im Abschnitt <em>mit Betonung</em> und ohne.</p><p>Ein <cite><em>weiterer</em> <span class="keyword"> <em>wichtiger</em> Abschnitt</span></cite> mit Text.</p></main>
```
- (b) Erläutern Sie knapp und präzise die Funktion der folgenden Stylesheet-Regeln und ihre konkrete Wirkung auf den obigen HTML-Ausschnitt.

```

(b1) h1 {font-weight: bold}
(b2) .keyword {color: green}
(b3) h1:hover {color: blue}
(b4) p em {color: red}
(b5) p * em {font-style: normal}
(b6) p > em {font-style: italic}

```

- (c) Welche Konflikte zwischen verschiedenen Regeln können auftreten und wie löst man sie?

#### Aufgabe 5 : CSS: Cascading Sizing (1+1 Punkte)

Gegeben sei folgendes HTML-Dokument (`css-cascading-sizing.html` im Repository).

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Verschachtelte Listen</title>
  </head>
  <body>
    <h1>Verschachtelte Listen</h1>
    <ul>
      <li>Pflanzen</li>
      <li>Tiere
        <ul>
          <li>Säugetiere
            <ul>
              <li>Hunde</li>
              <li>Katzen</li>
            </ul>
          </li>
          <li>Fische</li>
        </ul>
      </li>
    </ul>
  </body>
</html>

```

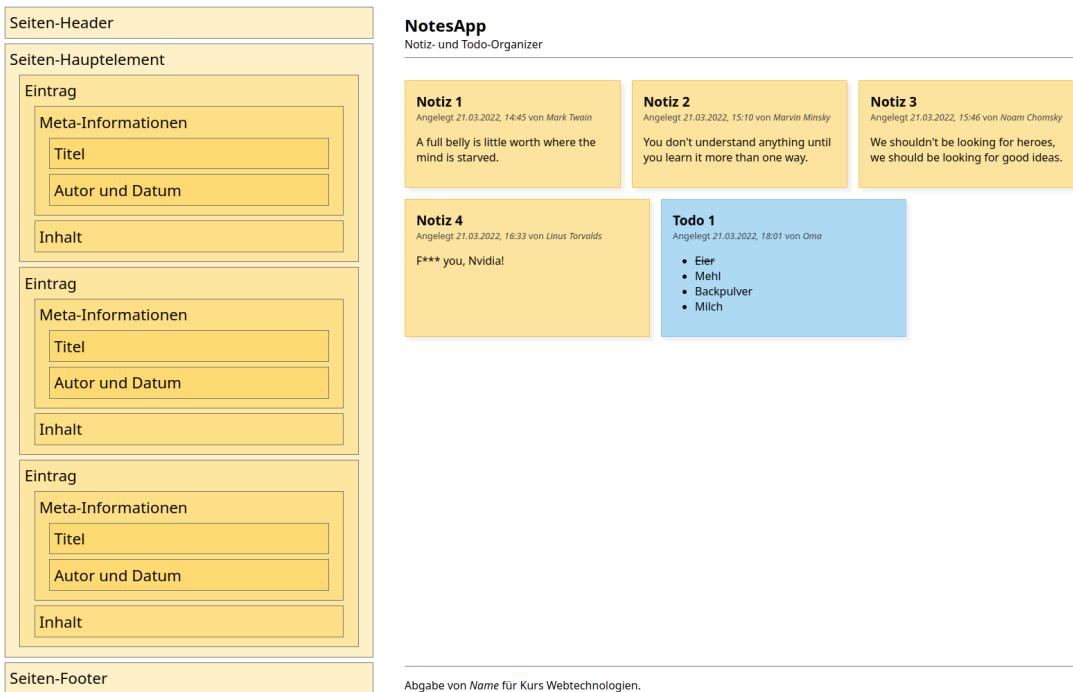
- (a) Ergänzen Sie eine CSS-Regel im `<head>` des Dokumentes, die der Schriftgröße (Property `font-size`) von Listenelementen `li` den Wert `1.2em` zuweist. Vergleichen Sie die Darstellungen mit und ohne diese Regel in Ihrem Browser und beschreiben Sie: Welches ggf. unerwünschte Verhalten tritt zu Tage und warum? Sie können [W3C: CSS Values and Units \(Relative lengths\)](#) für Informationen zu relativen CSS-Maßeinheiten konsultieren.
- (b) Ändern Sie lediglich die Maßeinheit der Schriftgröße, um das unerwünschte Verhalten zu unterbinden. Welche andere relative Maßeinheit bietet sich hierfür an? Begründen Sie!

#### Aufgabe 6 : Programmiermodul: NotesApp HTML+CSS P (4+5+1 Punkte)

Forken Sie das [Repository](#) dieser Abgabe und erstellen Sie ein HTML-Gerüst und CSS-Layout für „NotesApp“, eine simple browserbasierte Notiz- und Todo-Verwaltungsapp. Die App soll zwei verschiedene Typen von Einträgen darstellen können: Notizen („Post-its“) und Todo-Listen.

- (a) Erstellen Sie ein [HTML5](#)-Dokument `index.html`, welches das Seiten-Grundgerüst der App enthält und mindestens vier Einträge (dabei mindestens einen jedes Typs) beispielhaft abbildet. Folgende Bedingungen müssen erfüllt sein:

- Die Struktur muss der schematischen Übersicht der folgenden Grafik (linke Hälfte) folgen. Jeder Kasten der Grafik muss durch mindestens ein eigenes HTML-Element repräsentiert sein. Die rechte Hälfte der Grafik veranschaulicht das finale Aussehen des Layouts beispielhaft.



- Mindestens vier der semantischen Elemente aus HTML5 müssen benutzt werden:  
`<article>, <aside>, <details>, <figcaption>, <figure>, <footer>, <header>, <main>, <mark>, <nav>, <section>, <summary>, <time>.`  
 Die Elemente müssen getreu ihrer Semantik benutzt werden.

(b) Erstellen Sie eine CSS-Datei für das Layout der App und referenzieren Sie diese im HTML-Dokument. Folgende Bedingungen an das Layout müssen erfüllt sein:

- Einträge, besitzen eine Maximalbreite, sind nebeneinander angeordnet und brechen zeilenweise um (ähnlich wie in der Grafik). Sowohl für das generelle Seitenlayout als auch für die zeilenweise Anordnung der Einträge, ist CSS-Flexbox zu verwenden.
- Die einzelnen Einträge müssen klar voneinander getrennt sein.
- Einträge vom Typ „Notiz“ und vom Typ „Todo“ sind farblich klar unterscheidbar.
- Einträge vom Typ „Todo“ besitzen als Inhalt eine unsortierte HTML-Liste von Elementen. Erledigte Listenelemente sind also solche gekennzeichnet (z.B. durchgestrichen).
- Überschriften, Inhaltstext und Meta-Informationen müssen klar unterscheidbar sein.
- Seiten-Header, Hauptinhaltsbereich und Seiten-Footer müssen klar unterscheidbar sein.

Jenseits dieser Vorgaben können Sie selbstverständlich beliebig kreativ werden.

(c) Erweitern Sie die CSS-Datei, sodass für Browserfenster schmäler als 650px – und nur für diese – Einträge mittels CSS-Media-Queries linear untereinander statt, wie in der obigen Grafik, reihenweise nebeneinander dargestellt werden.

## Hinweise zur Abgabe

- Erstellen Sie *eine* PDF-Datei, in der Texte und Grafiken zu den Aufgaben enthalten sind.
- Quellcode (Python, Java, JavaScript, PHP, HTML, CSS, XML, XSL, XSD, etc.) innerhalb des PDF-Dokuments wird nicht korrigiert. Quellcode jeder Aufgabe soll, wie im Tutorium vorgestellt, über GitLab verwaltet werden.
- Stellen Sie sicher, dass Sie keine in der `.gitignore`-Datei gelisteten Dateien und Verzeichnisse versehentlich mit committen.
- Referenzieren Sie Ihre Quellcode-Dateien in dem PDF-Dokument, so dass sie einer Aufgabe eindeutig zuzuordnen sind. Verlinken sie sie entsprechenden Repositories.
- Gruppenabgaben mit bis zu drei Personen sind erlaubt; pro Gruppe genügt dann eine Abgabe, in der alle Gruppenmitglieder mit Namen, Matrikel, und E-Mail-Adresse verzeichnet sind.
- Abzugeben ist *eine* PDF-Datei über Moodle.