Klausur Web-Anwendungen, SS17

B_CGT 14.0, B_ECom I14.0, B_Inf 14.0, B_MInf 14.0, B_WInf 14.0,
ITAM 2.0, ITAS 2.0, ITAW 2.0, KAI 2.0

- Gestellt von: Marcus Riemer (**mri**)
- Erlaubte Hilfsmittel: Zeichengeräte, Taschenrechner.
- Dauer: 60 Minuten.
- Diese Klausur besteht inklusive dieses Deckblattes und den Anhängen aus **6 Fragen** auf **13 Seiten**. Es können maximal **45** **Punkte** erreicht werden. Punktzahlen werden in Kästen notiert und einmal als Summe (Σ) je Aufgabe und dann einzeln für jede Teilaufgabe angegeben.
- Sie sollten zu keinem Zeitpunkt zwischen mehreren Seiten blättern müssen, um sich vorgegebene Quelltexte anzusehen. Daher werden alle **Anhänge mit den Quelltexten** zur Klausur **zusätzlich separat ausgeteilt**. Die Heftung der Anhänge darf **nicht** gelöst werden, Notizen auf dem separat ausgeteilten Anhang werden keinesfalls bewertet. Sie müssen die Anhänge daher auch nicht wieder abgeben.
- Verwenden Sie zur **Lösung** die **freigelassenen Räume auf den Aufgabenblättern** (diese können, müssen aber nicht komplett gefüllt werden). Sollte der Platz nicht ausreichen, verwenden Sie die Rückseite der Aufgabenblätter.
- Der Inhalt der **Rückseiten** wird nur **gewertet**, wenn er eindeutig als Lösung gekennzeichnet ist (Stichwort "Lösung" mit Angabe der Aufgabennummer) und vom vorgesehenen Lösungsfeld auf die zu wertende Rückseite verwiesen wird.

1) Semantik und Syntax von HTML

$\sum 6$

Das in Anhang 1a zu sehende HTML-Dokument enthält einige Fehler und ist teilweise semantisch unsinnig. Geben sie zwei strukturelle oder syntaktische Fehler an und benennen Sie einen semantisch unsinnigen Sachverhalt. Beschreiben Sie dabei knapp die Fehlerursache unter Angabe einer Zeilennummer.

(a) Syntaktisch fehlerhaftes HTML in Zeile _____ 2

(b) Syntaktisch fehlerhaftes HTML in Zeile _____ 2

(c) Semantisch unsinniges HTML in Zeile _____ 2

2) CSS

$\sum 7$

(a) Spezifität von Selektoren [5]

Berechnen Sie die Spezifitäten für die in Anhang 2a angegebenen CSS-Selektoren. Geben Sie die einzelnen Werte in einer Tabelle an und verwenden Sie eine Tabellenzeile für jeden der fünf zu bewertenden Selektoren.

(b) Welches Kriterium wird genutzt, wenn die Spezifität zweier Selektoren exakt identisch ist? [1]

(c) Wie können für bestimmte Elemente im DOM-Baum abweichende CSS-Eigenschaften gesetzt werden, ohne dabei die Regeln für Spezifitäten zu berücksichtigen? [1]

3) CSS

$\sum 5$

Schreiben Sie jeweils einen CSS-Selektor, der exakt die Elemente der angegebenen Zeilen in den entsprechenden Listings selektiert.

Hinweis: Es ist nach genau **einem** Selektor je Aufgabenstellung gefragt, auch wenn mehrere HTML-Dokumente referenziert werden. Diese Anforderung schließt die Verwendung des Komma-Operators explizit aus.

(a) Anhang 3a: Zeile 4 [1]
Anhang 3b: Zeile 5

(b) Anhang 3a: Zeilen 9 und 13 [1]
Anhang 3b: Zeilen 10 und 14

(c) Für diesen Aufgabenteil sollen die Elemente nur selektiert werden, wenn die Maus sich über den entsprechenden Elementen befindet. [2]

Anhang 3a: Zeilen 6 & 7, 10 & 11, 14 & 15
Anhang 3b: Zeilen 7 & 8, 11 & 12, 16 & 16

(d) Für diesen Aufgabenteil sollen die Elemente selektiert werden, wenn in der URL des aktuell angezeigten Dokumentes das Fragment `'my-heroes'` vorkommt. [1]

Anhang 3a: Zeile 4
Anhang 3b: Zeile 5

4) JavaScript

$\sum 8$

Betrachten Sie das JavaScript-Programm in Anhang 4a.

(a) Was ist der gängige (und in der Vorlesung verwendete Name) für die Funktion `iter1`? 1

(b) Was ist der gängige (und in der Vorlesung verwendete Name) für die Funktion `iter2`? 1

(c) Welche Ausgabe erzeugt das Programm? 3

(d) Betrachten Sie lediglich die Definitionen von `iter3` und `iter4` sowie den Aufruf `iter3(iter4())`. Wird dieser Aufruf zur Laufzeit zu einem Absturz führen? 3

☐ Ja ☐ Nein

Wenn "Ja": In welcher Zeile findet der Absturz statt und was ist die Ursache?

Wenn "Nein": Was ist die Ausgabe des Programms?

5) XML

$\sum 7$

Die XML-Dokumente für diese Aufgabe (zu finden in den Listings 5b, 5c, 5d und 5e) sind syntaktisch einwandfrei.

(a) Welches dieser Dokumente ist im Sinne der Schema-Definition in Anhang 5a valide? [1]

☐ Dokument 1 ☐ Dokument 2 ☐ Dokument 3 ☐ Dokument 4

(b) Benennen Sie je einen Fehler in den übrigen Dokumenten. Nehmen Sie dabei Bezug auf die Zeile, in welcher der Fehler auftritt, die Zeile(n) im XML, in denen die verletzte Restriktion genannt wird und begründen Sie stichwortartig die Art der Verletzung.

i. Dokument _____ in XML-Zeile _____, Schema-Zeile: _____ [2]

ii. Dokument _____ in XML-Zeile _____, Schema-Zeile: _____ [2]

iii. Dokument _____ in XML-Zeile _____, Schema-Zeile: _____ [2]

6) Gemischte Themengebiete $\sum 12$

Setzen Sie exakt die Anzahl der geforderten Kreuze! Sofern Sie mehr Kästchen ankreuzen als die Aufgabenstellung vorsieht, wird die entsprechende Teilaufgabe mit 0 Punkten bewertet.

(a) `HTTP`

i. Welche der folgenden `HTTP`-Verben nehmen typischerweise keine Veränderungen an den referenzierten Ressourcen vor (2 Kreuze)? [1]

- ☐ `GET`
- ☐ `HEAD`
- ☐ `POST`
- ☐ `PUT`
- ☐ `PATCH`
- ☐ `DELETE`

ii. Was unterscheidet `GET` von `HEAD`? [1]

(b) Welche Aussagen über `CSS`-Selektoren und `CSS`-Deklarationen sind wahr? (2 Kreuze) [4]

- ☐ Die Struktur eines `HTML`-Dokumentes hat Einfluss auf die Spezifität eines Selektors.
- ☐ Die anzuwendenden Deklarationen haben Einfluss auf die Spezifität eines Selektors.
- ☐ Kinder von Elementen mit `display: flex;` werden in ihrer Spezifität anders behandelt als Elemente mit `float: left;` oder `float: right;`.
- ☐ Pseudo-Elemente (`::before`) und Pseudo-Klassen(`:hover`) sind aus Sicht der Spezifität identisch relevant.
- ☐ Der universelle Selektor (`*`) sowie die Kombinatoren (`+`, `>`, `~` und Leerzeichen) haben keinen Einfluss auf die Spezifität.
- ☐ Das Pseudo-Element `first-child` ist aus Sicht der Spezifität relevanter als `last-child`.
- ☐ Das Pseudo-Element `last-child` ist aus Sicht der Spezifität relevanter als `first-child`.
- ☐ Klassenselektoren (`.class`) und Attributselektoren (`[name='foo']`) sind aus Sicht der Spezifität identisch relevant.

(c) Serverseitiges JavaScript mit node.js

i. Das Ausführungsmodell von node.js bezeichnet man als ... (1 Kreuz) [1]

☐ Single-Threaded, non blocking `I/O`

☐ Single-Threaded, blocking `I/O`

☐ Multi-Threaded, non blocking `I/O`

☐ Multi-Threaded, blocking `I/O`

ii. Beschreiben Sie anhand der Auswirkung auf Endbenutzer: Welches Problem entsteht, wenn im Rahmen einer Anfrage an eine node.js-Server-Instanz innerhalb einer häufig durchlaufenen Schleife komplexe Berechnungen (wie zum Beispiel Primzahltests) `ohne` Nutzung von Callback-Funktionen vorgenommen werden? [2]

(d) "Media Queries" bilden die technische Grundlage für so genanntes "Responsive Design". Wie funktioniert diese Technik? (1 Kreuz) [2]

☐ Ein JavaScript-Programm fragt die Dimensionen des Viewports (`media.dimensions`) ab und lädt dann dazu passende weitere JavaScript-Programme.

☐ Ein JavaScript-Programm fragt die fest hinterlegte Geräteklassen (`screen`, `print` und `speech`, Zugriff über `media.class`) ab und lädt dann dazu passende weitere JavaScript-Programme.

☐ In `CSS`-Dokumenten werden mit `SQL`-Abfragen Selektoren und Deklarationen weiter eingeschränkt.

☐ In `CSS`-Dokumenten werden mit JavaScript-Anweisungen Selektoren und Deklarationen weiter eingeschränkt.

☐ In `CSS`-Dokumenten werden mit `@media`-Regeln Selektoren und Deklarationen weiter eingeschränkt.

(e) "Promises" sind eine der Möglichkeiten in JavaScript auf das Ergebnis von andauernden Berechnungen zu warten. Welche der folgenden Aussagen ist wahr? (1 Kreuz) [1]

☐ Die `Promise`-Schnittstelle ist Bestandteil von `ECMA`-Script 6 und steht deswegen in jeder konformen Laufzeitumgebung zur Verfügung.

☐ Die `Promise`-Schnittstelle ist Bestandteil von node.js und ist deswegen nicht in Browsern verfügbar.

☐ Die `Promise`-Schnittstelle ist Bestandteil der `DOM`-Schnittstelle und ist deswegen ausschließlich in Browsern verfügbar.

Anhang: Aufgabe 1

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Meine Seite</title>
5    </head>
6    <body>
7      <header id="der-"kopf"">
8        Willkommen auf meiner Seite
9      </header>
10     <nav>
11       <table>
12         <tr>
13           <td><a href="/Du-und-Ich">Du  &amp; Ich</a></td>
14           <td><a href="/Er-und-Sie">Ich &amp; Sie</a></td>
15           <td><a href="/Sie-gt-Er">Sie > Er</a></td>
16         </tr>
17       </table>
18     </nav>
19     <article>
20       Ich verkaufe diese sch&ouml;nen Lederjacken.
21     </article>
22     <footer>
23       Copyright 2012: Ein dreik&ouml;pfiger Affe
24     </footer>
25   </body>
26 </html>
```

(a) Fragwürdiges HTML

Anhang: Aufgabe 2

```
1  ul > li + ul ~ li    { color: red; }
2  #friedrich .heinrich { color: green; }
3  input[name=klausur]  { color: blue; }
4  #ustus p.eter b.ob   { color: white; }
5  p.eter b.ob:hover    { color: black; }
```

(a) Zu bewertendes CSS

```
1  <ul class="heinrich">
2    <li id="friedrich">
3      <ul class="ustus">
4        <li>
5          <p class="eter">
6            <b class="bob">Erster</b> Detektiv
7          </p>
8        </li>
9      </ul>
10   </li>
11 </ul>
```

(b) HTML-Fragment

Anhang: Aufgabe 3

```html
<!DOCTYPE html>
<html>
  <body>
    <table id="my-heroes">
      <tr class="heading">
        <th>Name</th>
        <th>Role</th>
      </tr>
      <tr class="row">
        <td class="name">Beastmaster</td>
        <td class="role">Jungler</td>
      </tr>
      <tr class="row">
        <td class="name">Beastmaster</td>
        <td class="role">Jungler</td>
      </tr>
    </table>
  </body>
</html>
```

(a) HTML

```html
<!DOCTYPE html>
<html>
  <body>
    <p class="name role"></p>
    <div id="my-heroes">
      <div class="heading">
        <div>Name</div>
        <div>Role</div>
      </div>
      <div class="row">
        <div>Beastmaster</div>
        <div>Jungler</div>
      </div>
      <div class="row">
        <div>Beastmaster</div>
        <div>Jungler</div>
      </div>
    </table>
  </body>
</html>
```

(b) HTML

Anhang: Aufgabe 4

Hinweis: Die Funktion `Array.prototype.push` hängt ein neues Element an das entsprechende Array an. Das Array auf dem die Funktion aufgerufen wird verändert sich dadurch.

```
1  const iter1 = function(a, b) {
2    const toReturn = [];
3    for (let i of a) {
4      toReturn.push(b(i));
5    }
6    return (toReturn);
7  }
8
9  const iter2 = function(a, b) {
10   for (let i = 0; i < a.length; i++) {
11     b(a[i]);
12   }
13 }
14
15 const iter3 = function(a) {
16   iter2(a, console.log);
17 }
18
19 const iter4 = function*() {
20   yield('a');
21   yield('bc');
22   yield('def');
23 }
24
25 const a = [];
26 const b = ['a', 'bc', 'def'];
27 const c = iter1(b, a => a.length);
28 const d = iter1(c, a => a + b.length);
29
30 iter3(c);
31 iter3(d);
```

(a) Programm

Anhang: Aufgabe 5

```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="address">
    <xs:complexType>
      <xs:choice>
        <xs:element name="generic" type="xs:string" />
        <xs:element name="germany">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="name"   type="xs:string" />
              <xs:element name="street" type="xs:string" />
              <xs:element name="zip"    type="xs:integer" />
              <xs:element name="city"   type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

(a) XML-Schema

```xml
<address><generic>Weihnachtsmann</generic></address>
```

(b) Dokument 1

```xml
<generic>Geller,425 Grove Street,10014 New York</generic>
```

(c) Dokument 2

```xml
<address>
  <germany>
    <name>Dirk Harms</name>
    <street>FS 143</street>
    <zip>Wedel</zip>
    <city>-1337</city>
  </germany>
</address>
```

(d) Dokument 3

```xml
<address>
  <germany>
    <name><street></street></name>
    <street>XML Boulevard</street>
    <zip>887776</zip>
    <city>XML City</city>
  </germany>
</address>
```

(e) Dokument 4