Klausur Web-Anwendungen, WS18

B_CGT, B_Inf, B_MInf, B_WInf, EComI

- Gestellt von: Marcus Riemer (**mri**) und Florian Schatz (**fls)**
- Erlaubte Hilfsmittel: Zeichengeräte, Taschenrechner.
- Dauer: 60 Minuten.
- Diese Klausur besteht inklusive dieses Deckblattes und den Anhängen aus **7 Fragen** auf **15 Seiten** (die Fragen enden auf **Seite 10**). Es können maximal **51** **Punkte** erreicht werden. Punktzahlen sind in Kästen notiert und einmal als Summe (Σ) je Aufgabe und dann einzeln für jede Teilaufgabe angegeben.
- **Wahlaufgabe**: Es wird **entweder** Aufgabe 6 **oder** Aufgabe 7 gewertet. Sie legen unten auf dieser Seite fest, welche Lösung in die Benotung einfließen soll.
- Sie sollten zu keinem Zeitpunkt zwischen mehreren Seiten blättern müssen, um sich vorgegebene Quelltexte anzusehen. Daher werden alle **Anhänge mit den Quelltexten** zur Klausur **separat ausgeteilt**. Die Heftung der Anhänge sowie der Klausur dürfen **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.

Ich bearbeite folgende Wahlaufgabe:

☐ Aufgabe 6 (Themenkomplex "Cloud", Vorlesung "Web-Technologien")

☐ Aufgabe 7 (Themenkomplex "Javascript", Vorlesung "Web-Anwendungen")

1) Semantik und Syntax von `HTML` $\boxed{\sum 6}$

Das `HTML`-Dokument in Anhang 1a enthält einige Fehler und ist teilweise semantisch unsinnig. Geben Sie einen strukturellen oder syntaktischen Fehler an und benennen Sie zwei semantisch unsinnige Sachverhalte. Beschreiben Sie dabei knapp die Fehlerursache unter Angabe einer Zeilennummer.

(a) Syntaktisch fehlerhaftes `HTML` in Zeile _____ 2

(b) Semantisch unsinniges `HTML` in Zeile _____ 2

(c) Semantisch unsinniges `HTML` in Zeile _____ 2

2) CSS

$\sum 9$

Im Falle von Mehrdeutigkeiten werden die anzuwendenden CSS-Deklarationen anhand der so genannten "Spezifität" sortiert.

(a) Geben Sie für die folgenden Selektortypen an, in welcher "Spezifiätskategorie" sich dieser befindet. [2]

_____ Typ-Selektoren (`p`)
_____ Pseudo-Klassen (`:hover`)
_____ ID-Selektoren (`#beispiel`)
_____ Attribut-Selektoren (`[href="/"]`)

(b) Berechnen Sie die Spezifität des folgenden Ausdrucks: [2]

```
p > * + * #important.type
```

(c) Betrachten Sie die HTML-Struktur in Anhang 2a. Schreiben Sie einen CSS-Selektor, der die `li`-Elemente in den Zeilen 8 und 17 selektiert **ohne** dabei die `li`-Elemente in den Zeilen 23 und 24 zu selektieren. [2]

Der in Anhang 2a zu sehende Ausschnitt wird mit den CSS-Regeln aus Anhang 2b dargestellt. Welche Eigenschaften werden dadurch auf den folgenden Elementen gesetzt sein?

(d) Anhang 2a: Zeile 1 (`article`) [1]

(e) Anhang 2a: Zeile 1 (`article`), allerdings befindet sich nun die Maus des Nutzers über diesem Element. [1]

(f) Anhang 2a: Zeile 23 & 24 (`li`) [1]

3) JavaScript

$\sum 8$

Betrachten Sie das JavaScript-Programm in Anhang 3a.

(a) Die Funktion `iterDouble` implementiert gleich zwei der in der Vorlesung vorgestellten Iterationsfunktionen und wendet diese nacheinander an. Um welche Funktionen handelt es sich? 2

Zuerst angewandte Funktion:

Zuletzt angewandte Funktion:

(b) Was ist die Ausgabe des `console.log`-Aufrufs mit dem Kommentar "Ausgabe 1"? 2

(c) Was ist die Ausgabe des `console.log`-Aufrufs mit dem Kommentar "Ausgabe 2"? 2

(d) Was ist die Ausgabe des `console.log`-Aufrufs mit dem Kommentar "Ausgabe 3"? 2

4) Gemischte Themengebiete $\boxed{\sum 9}$

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) Welche der folgenden `JavaScript`-Ausdrücke ist standardmäßig nur in einem Browser, nicht aber in einer anderen `JavaScript`-Umgebung (wie z.B. `fhw-web` mit `node`) ausführbar (1 Kreuz)? [1]

- ☐ `[1,2,3].filter(v => v >= 2);`
- ☐ `[4,5,6].some(v => v >= 4);`
- ☐ `document.querySelector(`#server`);`
- ☐ `fs.readFile('./data.json', console.log);`

(b) Wofür steht die Abkürzung `MVC` (im Kontext der Vorlesung, 1 Kreuz)? [1]

- ☐ `Method, View, Checkbox`
- ☐ `Model, Value, Controller`
- ☐ `Method, Value, Constraint`
- ☐ `Message, Value, Constraint`
- ☐ `Method, View, Controller`
- ☐ `Model, View, Controller`
- ☐ `Message, View, Controller`
- ☐ `Model, Value, Constraint`

(c) Welche dieser Angaben können in einer standardkonformen `package.json`-Datei (zur Definition von Projekten oder Paketen) gemacht werden (2 Kreuze)? [2]

- ☐ Abmessungen, Transportdienstleister und Gewicht des Pakets
- ☐ Inhaltsangaben für Zollbefreiung
- ☐ Lizenzangabe um zu spezifizieren, unter welchen Bedingungen der Quelltext verwendet werden darf
- ☐ Notfallkontaktinformationen, falls das Paket aus dem Serverraum entwendet werden sollte
- ☐ Tracking-Nummer um bei verzögerten Zustellungen den Status abfragen zu können
- ☐ Versionsangabe zur Kommunikation von Kompatabilitäten
- ☐ Boolescher Wert “Bitte nicht auf den Kopf stellen”
- ☐ Portokosten

(d) Welche dieser Begriffe bezeichnen Bestandteile einer `URL` (2 Kreuze)? 2

- ☐ Query
- ☐ Select
- ☐ From
- ☐ Where
- ☐ `AJAX`
- ☐ `DOM`
- ☐ `CSS`
- ☐ Callback
- ☐ Closure
- ☐ Host

(e) Welche dieser Ausdrücke sind gültiges `JSON`? Gültige `JSON`-Strings lassen sich mit `JSON.parse` fehlerfrei in `JavaScript`-Objekte umwandeln (2 Kreuze). 2

- ☐ `[ ["valid"], true ]`
- ☐ `{ valid: true }`
- ☐ `{ [ valid: true ] }`
- ☐ `{ valid: true || false }`
- ☐ `{ "valid": `true` // Actually true! }`
- ☐ `{ "valid": null }`
- ☐ `{ 1: "valid" }`
- ☐ `{ 0: "valid" }`
- ☐ `{ "valid": false, } }`
- ☐ `{ "valid": FALSE }`
- ☐ `{ "valid": true, { "sub": "valid!" } }`

(f) Wie stellt ein standardkonformer Browser ihm unbekannte `HTML`-Elemente dar? Beispielelemente: `<sarcasm>Oh, wie nützlich</sarcams>` oder `<irony>Das sah sehr gut aus<irony>` (1 Kreuz). 1

- ☐ Als `block`-Element, sie nehmen also so viel horizontalen Platz wie möglich ein.
- ☐ Als `inline`-Element, sie betten sich also in den Textfluss ein.
- ☐ Als `flex`-Element, ihre Kinder lassen sich also flexibel anordnen.
- ☐ Als `table`-Element, ihre Kinder müssen also in Zeilen und Spalten organisiert werden.
- ☐ Überhaupt nicht

5) Vertiefung JavaScript

$\sum 10$

(a) Was unterscheidet den Operator `===` vom Operator `==`? 1

(b) Berechnen sie beispielhaft anhand zweier von ihnen gewählten, konkreten Werten einmal mit dem `===`-Operator und einmal mit dem `==`-Operator zwei **unterschiedliche** Ergebnisse. 2

Betrachten Sie das JavaScript-Programm in Anhang 5a.

(c) Was ist die Ausgabe des `console.log`-Aufrufs mit dem Kommentar "Ausgabe 1"? 1

(d) Was ist die Ausgabe des `console.log`-Aufrufs mit dem Kommentar "Ausgabe 2"? 1

(e) Was ist die Ausgabe des `console.log`-Aufrufs mit dem Kommentar "Ausgabe 3"? 1

(f) Die gezeigte Implementierung von `some` mit `filter` ist für bestimmte Eingaben sehr viel langsamer, als es eine optimierte Fassung sein könnte. Geben Sie ein konkretes Beispiel für eine Eingabe, bei welcher die Funktion unnötig viel rechnet. Was charakterisiert diese "optimierbaren" Eingaben allgemein? Und wie sähe die Optimierung aus (kurze Beschreibung genügt, kein Quelltext notwendig)? 4

6) Cloud-Infrastruktur (Wahl) $\boxed{\sum 10}$

(a) Welche Deployment Modelle ausser Public Cloud gibt es nach der NIST Definition? Beschreiben Sie in einem Satz einen Anwendungsfall, in dem es sinnvoll ist. [6]

(b) Beschreiben Sie in einem Satz was dynamische und statische Inhalte einer Internetseite sind. Nennen Sie dann mit Bezug auf einen Online-Shop jeweils zwei Beispiele mit einem Satz Begründung [4]

7) Fortgeschrittenes JavaScript (Wahl) $\sum 10$

(a) Betrachten Sie das JavaScript-Programm in Anhang 7a. Inwiefern verhalten sich mit `function` deklarierte Funktionen beim Aufruf anders als Arrow-Funktionen (deklariert mit `=>`)? Welche Art der Funktionsdefinition ist in diesem Fall sinnvoller und warum? [4]

Betrachten Sie das JavaScript-Programm in Anhang 7b.

(b) Was ist die Ausgabe des `console.log`-Aufrufs mit dem Kommentar "Ausgabe 1"? [1]

(c) Was ist die Ausgabe des `console.log`-Aufrufs mit dem Kommentar "Ausgabe 2"? [1]

(d) Was ist die Ausgabe des `console.log`-Aufrufs mit dem Kommentar "Ausgabe 3"? [2]

(e) Die Ausgabe des `console.log`-Aufrufs mit dem Kommentar "Ausgabe 4" wird niemals erfolgen, warum? [2]

Anhang: Aufgabe 1

```
1  <!DOCTYPE html>
2  <html lang="de">
3    <body>
4      <header>
5        <img src="/header-logo.svg">
6          Kontoübersicht
7        </img>
8      </header>
9      <div class="main">
10       <h1>Sie haben die folgenden Konten bei uns registriert:</h1>
11       <ul>
12         <li>Battle.net | UserID: MikeTheBike | Registriert: 1.1.2017</li>
13         <li>GOG.com | UserID: ChadTheQuad | Registriert: 1.1.2017</li>
14       </ul>
15     </div>
16     <footer>
17       <table>
18         <tr>
19           <td class="pull-left"><button id="logout">Logout</button></td>
20           <td class="pull-right">(C) 2019 Blockoli Inc.</td>
21         </tr>
22       </table>
23     </footer>
24   </body>
25 </html>
```

(a) Fragwürdiges HTML

Anhang: Aufgabe 2

```html
<article id="history-wedel">
  <section>
    <h1>Von der PTL zur FH Wedel</h1>
    <p>
      Die Geschichte von <time datetime="1945">1945</time>
      bis <time datetime="1974">1974</time>.
      <ul>
        <li>
          <details>
            <summary>1945</summary>
            Die naturwissenschaftlichen Berufsfachlehrgänge
            von Prof. Dr. Helmut Harms in Lübeck sind die
            Grundsteine der Physikalisch-Technischen Lehranstalt
            (PTL) und der Fachhochschule Wedel (FH Wedel).
          </details>
        </li>
        <li>
          <details>
            <summary>1955</summary>
            Ausgebildet werden mittlerweile die folgenden
            Studiengänge:
            <ol>
              <li>Physikalisch-Technische Assistenten</li>
              <li>Physik-Ingenieure</li>
            </ol>
          </details>
        </li>
      </ul>
    </p>
  </section>
</article>
```

(a) HTML-Fragment

```css
article:hover { background-color: red; }
li { font-style: italic; }
ul li { font-style: bold; }
#history-wedel { background-color: yellow; }
```

(b) CSS-Dokument

Anhang: Aufgabe 3

Hinweis: Die Funktion `Array.prototype.concat` gibt ein neues Array zurück, bei welchem das übergebene Element an das referenzierte Array hinten angehangen wurde. Das referenzierte Array wird dabei nicht verändert.
`[1].concat([2])` berechnet also ein neues Array `[1,2]`.

```
1  const iterDouble = function(a, b, c, d) {
2    let toReturn = b;
3    for (let i = 0; i < a.length; i++) {
4      const curr = c(a[i], i, a);
5      toReturn = d(toReturn, curr, i, a);
6    }
7    return (toReturn);
8  }
9
10 const data = [
11   { name: "Clara", job: "Companion" },
12   { name: "Adele", job: "Author" }
13 ]
14
15 // Ausgabe 1
16 console.log(
17   iterDouble(
18     data,
19     "Amilia",
20     (v) => v.name,
21     (a, v) => `${a},  ${v}`
22   )
23 )
24
25 // Ausgabe 2
26 console.log(
27   iterDouble(
28     data,
29     2000,
30     (v) => 9,
31     (a, v) => a + v
32   )
33 )
34
35 // Ausgabe 3
36 console.log(
37   iterDouble(
38     data,
39     [],
40     (v) => [v.job],
41     (a, v) => a.concat(v)
42   )
43 )
```

(a) Programm

Anhang: Aufgabe 4 (Leer)

(a) Clara Adele drückt Ihnen die Daumen!

(Dieser Anhang existiert nur aus ärgerlichen technischen Gründen und hat keine Bedeutung für die Klausur als solche.)

Anhang: Aufgabe 5

```
1  const some = function(arr, func) {
2    return (arr.filter(func).length >= 1);
3  };
4
5  const onlyThree = v => v == 3;
6
7  // Ausgabe 1
8  console.log(some([1,2,3], onlyThree));
9
10 // Ausgabe 2
11 console.log(some([33, 333], onlyThree));
12
13 // Ausgabe 3
14 console.log(some([], onlyThree));
```

(a) Programm

Anhang: Aufgabe 7 (Web-Anwendungen)

```
1  function mkPerson(name) {
2    return ({
3      "name": name,
4      "greetFun": function() {
5        console.log(`I am "${this.name}"`);
6      },
7      "greetArr": () => {
8        console.log(`Arr, I am "${this.name}"!`);
9      }
10   });
11 }
12
13 const p = mkPerson("Guybrush");
14 p.greetFun();
15 p.greetArr();
```

(a) Programm 1

```
1  const take = (it, pred) => {
2    if (typeof(pred) === "undefined") {
3      pred = () => true;
4    }
5    const toReturn = [];
6    for (let val of it) {
7      if (pred(val)) {
8        toReturn.push(val);
9      } else {
10       break;
11     }
12   }
13   return (toReturn);
14 }
15 function* gen() {
16   yield true;
17   yield false;
18   yield true;
19 }
20 function* fibonacci() {
21   let [prev, curr] = [0, 1];
22   while (true) {
23     [prev, curr] = [curr, prev + curr];
24     yield curr;
25   }
26 }
27 // Aufruf 1
28 console.log(take(gen()));
29 // Aufruf 2
30 console.log(take(gen(), v => v));
31 // Aufruf 3
32 console.log(take(fibonacci(), v => v <= 10));
33 // Aufruf 4
34 console.log(take(fibonacci()));
```

(b) Programm 2