

Klausur Web-Anwendungen, SS18

B_CGT 14.0, B_Inf 14.0, B_MInf 14.0, B_WInf 14.0, ECom I14.0

- 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 **17 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

 $\Sigma 6$

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

 $\sum 8$

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

Hinweis: Es ist nach genau **einem** Selektor je Aufgabenstellung gefragt. Diese Anforderung schließt die Verwendung des Komma-Operators explizit aus.

- (a) Anhang 2a: Zeile 3
Anhang 2a: Zeile 17

1

- (b) Anhang 2a: Zeile 3
Anhang 2a: **Nicht** Zeile 17

2

- (c) Anhang 2a: **Nicht** Zeile 4
Anhang 2a: Zeile 8
Anhang 2a: **Nicht** Zeile 18

2

Betrachten Sie die **HTML**-Struktur in Anhang 2a. Der dort 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 (**main**)

1

- (e) Anhang 2a: Zeile 3 (**h3**)

1

- (f) Anhang 2a: Zeile 8 (**p**)

1

3) JavaScript

 $\Sigma 7$

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"? 1

4) Gemischte Themengebiete

 $\sum 11$

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) Was versteht man unter "Routing" in einem HTTP-Framework wie zum Beispiel **fhw-web** (1 Kreuz)? Die Zuordnung von ... 1
- ... **JSON**-Objekten auf Controller-Funktionen.
 - ... JavaScript-Funktionen auf Controller-Funktionen.
 - ... Zahlen auf Controller-Funktionen.
 - ... URLs auf Controller-Funktionen.
- (b) Welche der folgenden Aussagen über HTML-Formulare sind wahr (2 Kreuze)? 2
- Der Inhalt von **form**-Elementen kann ausschließlich bei aktiviertem JavaScript versendet werden.
 - input**-Elemente ohne **name**-Attribut bekommen einen zufälligen Namen zugewiesen.
 - input**-Elemente mit **type='radio'**-Attribut erlauben eine "1 aus n"-Auswahl.
 - form**-Elemente mit **method='post'**-Attribut übertragen ihre Daten als Teil der aufgerufenen URL.
 - In JavaScript kann mittels **FormData**-Objekten auf Daten aus Formularen zugegriffen werden, um Sie zum Beispiel als **AJAX**-Anfrage zu verschicken.
- (c) Welche der folgenden Aussagen über die Sicherheit von Web-Anwendungen sind wahr (3 Kreuze)? Ein Anwender ... 3
- ... kann mittels einer **POST**-Anfrage beliebigen JavaScript-Code zum Server schicken um ihn dort als Controller auszuführen.
 - ... kann den JavaScript-Code, der in seinem Browser ausgeführt wird, komplett oder teilweise blockieren.
 - ... kann den JavaScript Code, der auf dem Server in Controller-Funktionen ausgeführt wird, komplett oder teilweise blockieren.
 - ... kann den in seinem Browser angezeigten **DOM**-Baum beliebig modifizieren.
 - ... kann die an den Server verschickten Formular-Daten nur im Rahmen der im **HTML**-Dokument vorgegebenen Validierungsgrenzen (z.B. **min** und **max** für **type='number'**) verschicken.
 - ... kann mittels der Route **/s3cr3t/nsa-backdoor** auf jeder Seite beliebigen JavaScript-Code auf dem Server ausführen.
 - ... kann mit jeder Anfrage vollkommen beliebige Cookies zum Server schicken.
 - ... kann das Anlegen von Cookies nicht verhindern.
 - ... kann mittels **PUT**-Anfragen (**CREATE** aus dem **CRUD**-Zyklus) beliebige **HTML**-Dokumente auf jedem Webserver erstellen.

(d) Welche der folgenden Programmier-Schnittstellen stehen auf einem Server in der node.js-Umgebung **standardmäßig** (also **ohne** die Nutzung des **fhw-web-Frameworks**) zur Verfügung (3 Kreuze)? 3

- `document.querySelector(selector)` zum Zugriff auf den DOM-Baum.
- `new FormData(form)` zum Zugriff auf die Daten in einem `<form>`-Element.
- `require(path)` zum Laden von anderen JavaScript-Dateien.
- `fhwWeb.loadJson(path)` zum Laden von JSON-Dokumenten.
- `JSON.stringify(obj)` zur Umwandlung eines Objekts in eine String-Darstellung.
- `forEach(cb)` auf Array-Objekten (z.B. `[].forEach(...)`).
- `groupBy(cb)` auf Array-Objekten (z.B. `[].groupBy(...)`).
- `having(cb)` auf Array-Objekten (z.B. `[].having(...)`).

(e) Welche der folgenden Aussagen über **AJAX** sind wahr (2 Kreuze)? 2

- Es handelt sich um eine rein serverseitige Technik.
- Es handelt sich um eine rein serverseitige Technik, der Browser benötigt dafür aktiviertes JavaScript.
- Es handelt sich um eine rein clientseitige Technik, der Browser benötigt dafür **kein** aktiviertes JavaScript.
- Es handelt sich um eine rein clientseitige Technik, der Server muss **AJAX** nicht speziell unterstützen.
- Mit einem **AJAX**-Aufruf können ausschließlich **XML**-Daten verschickt werden.
- Mit einem **AJAX**-Aufruf können ausschließlich **JSON**-Daten empfangen werden.
- Mit einem **AJAX**-Aufruf können **HTML**-Daten empfangen und gesendet werden.
- Ein **AJAX**-Aufruf erfordert ein erneutes Laden der aktuellen Browserseite.

5) Fortgeschrittenes JavaScript

 $\Sigma 9$

Mit ECMA-Script 6 (besser bekannt als JavaScript 6) wurde eine neue Syntax für die objektorientierte Programmierung eingeführt. Anhang 5a sollte von einem unbedarften Programmierer von der `class`-Syntax in die explizite `prototype`-Syntax überführt werden (Anhang 5b). Dabei sind allerdings einige Fehler passiert: Eigentlich sollten sich alle Operationen auf identisch benannten Klassen exakt identisch verhalten, praktisch ist das nicht der Fall.

Zeigen Sie anhand der gegebenen Instanzen durch einen beispielhaften Aufruf Unterschiede in exakt der Klasse der jeweiligen Instanz. Begründen Sie bei Abweichungen kurz die Fehlerursache im Code von Anhang 5b.

- (a) Gegebene Instanz: `const p = new Person('Donald', 'Ente');`

3

Ihr Aufruf: _____

Ergebnis / Ausgabe / Fehler für Anhang 5a: _____

Ergebnis / Ausgabe / Fehler für Anhang 5b: _____

- (b) Gegebene Instanz: `const p = new Actor('Arnold', ['Terminator']);`

3

Ihr Aufruf: _____

Ergebnis / Ausgabe / Fehler für Anhang 5a: _____

Ergebnis / Ausgabe / Fehler für Anhang 5b: _____

- (c) Gegebene Instanz: `const p = new Writer('Terry', ['Hogfather']);`

3

Ihr Aufruf: _____

Ergebnis / Ausgabe / Fehler für Anhang 5a: _____

Ergebnis / Ausgabe / Fehler für Anhang 5b: _____

6) Cloud-Infrastruktur (Wahl)

 $\Sigma 10$

- (a) Welche Nachteile hat das Modell Public Cloud? Nennen Sie zwei Nachteile jeweils mit einem Satz Begründung. 2

- (b) Was kann ein Unternehmen machen, wenn es Daten hat, die es auf Systemen vor Ort speichern muss, aber die Vorteile einer Cloud einsetzen möchten. Denken Sie dabei an die von der NIST vorgegeben Modelle. 2

- (c) Welche Inhalte sind in der Regel im Beispiel in Anhang 6a statisch, welche dynamisch? Bitte listen Sie die Elemente auf. Denken Sie auch an ggf. AJAX-Inhalte.

4

- (d) Was würde ein Content Delivery Network bringen? Reduzieren Sie sich auf zwei Sätze.

2

7) Fortgeschrittenes JavaScript (Wahl)

 $\Sigma 10$

- (a) Was ist die Ausgabe des Programms in Anhang 7a?

3

- (b) Was ist, nach Ausführung des Programms, der Typ der Variable `a` in Zeile 5 von Anhang 7b?

1

- (c) Was ist die Ausgabe des Programms in Anhang 7b?

2

- (d) Gegeben sei die Funktionsdeklaration `function* fibonacci()`, welche dazu dient die Fibonacci-Sequenz zu erzeugen. Der Aufruf `fibonacci()` erzeugt ein Objekt mit zwei Eigenschaften. Wie heißen diese Eigenschaften und welchen Zweck erfüllen Sie (Stichwortartig)?

2

Gegeben sei die Funktionsdeklaration `function doAjaxRequest(url)`, welche dazu dient eine **AJAX**-Anfrage durchzuführen

- (e) Was für eine Art von Objekt muss ein Aufruf von `doAjaxRequest` zurückgeben, damit mittels des Schlüsselwortes `await` darauf gewartet werden kann (Fachbegriff)?

1

- (f) Welche syntaktische Randbedingung muss gegeben sein, damit das Schlüsselwort `await` genutzt werden kann?

1

Anhang: Aufgabe 1

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Anmeldung</title>
5   </head>
6   <body>
7     <div class="header">
8       Anmeldung zur Klausur
9     </div>
10    <main>
11      <form action="/api/register" method="POST">
12        <ol>
13          <li>
14            <input type="text" name="name">
15          </li>
16          <li>
17            <label>
18              <input type="text"> Ihre Klausur
19            </label>
20          </li>
21        </ol>
22      </form>
23    </main>
24  </body>
25 </html>
```

(a) Fragwürdiges HTML

Anhang: Aufgabe 2

```
1 <main>
2   <div class="card" id="clara">
3     <h3>Clara Oswald</h3>
4     <p>
5       Clara Oswald ist ein Mensch des 21. Jahrhunderts und wird zu
6       einer Begleiterin des Elften und Zwölften Doctors.
7     </p>
8     <p>
9       In der Episode The Name of the Doctor stellt sich heraus,
10      dass Clara bereits seit der ersten Inkarnation des Doctors über
11      ihn wacht - und somit die einzige Person überhaupt ist, die dem
12      Doctor in all seinen Inkarnationen begegnete.
13    </p>
14  </div>
15
16  <div class="card" id="adele">
17    <h3>Adele Blanc-Sec</h3>
18    <p>
19      Adele Blanc-Sec lebt in einem Pariser Appartement und bestreitet
20      ihren Lebensunterhalt als Detektivroman-Autorin. Zumeist gegen
21      ihren Willen wird sie immer wieder in mysteriöse Abenteuer
22      verwickelt, in deren Verlauf ihr wiederholt nach dem Leben
23      getrachtet wird.
24    </p>
25  </div>
26 </main>
```

(a) HTML-Fragment

```
1 .card p:last-child { text-align: center; }
2 main                { background-color: fuchsia; }
3 main > h3           { background-color: yellow; }
4 main *              { background-color: white; }
5 h3                  { color: blue; }
6 main                { display: flex; }
7 .card p             { text-align: left; }
```

(b) CSS-Dokument

Anhang: Aufgabe 3

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

(a) Programm

Anhang: Aufgabe 4 (Leer)



(a) Dies ist ein Platzhalter

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

Anhang: Aufgabe 5

```
1 class Person {
2   constructor(name, job) {
3     this.name = name;
4     this.job = job;
5   }
6   describe() { return (`${this.name}: ${this.job}`) }
7   hashCode() { return (this.name.length + this.job.length) }
8 }
9
10 class Actor extends Person {
11   constructor(name, movies) {
12     super(name, "Actor");
13     this.movies = movies;
14   }
15 }
16
17 class Writer extends Person {
18   constructor(name, books) {
19     super(name, "Autor");
20     this.books = books;
21   }
22   describe() {
23     return (`${this.name} schrieb ${this.books.length} Bücher`);
24   }
25 }
```

(a) Programm mit class

```
1 const Person = function(name, job) {
2   this.name = name;
3   this.job = job;
4 }
5 Person.prototype.describe = function() {
6   return(`${this.name}: "${this.job}"`);
7 }
8 Person.prototype.hashCode = function() {
9   return(this.name + this.job);
10 }
11
12 const Actor = function(name, movies) {
13   this.movies = movies;
14 }
15 Actor.prototype = Object.create(Person.prototype);
16
17 const Writer = function(name, books) {
18   Person.call(this, name, "Autor");
19   this.books = books;
20 }
21 Writer.prototype.describe = function() {
22   return(`${this.name} schrieb ${this.books.length} Bücher`);
23 }
```

(b) Programm mit expliziter Verwendung von prototype

Anhang: Aufgabe 6 (Web-Technologien)

```
1 <html>
2   <head>
3     <script src="load-my-auctions.js"></script>
4     <link rel="stylesheet" type="text/css" href="linked.css"/>
5   </head>
6   <body>
7     <div id="auctions">Ihre Auktionen mit Bild werden umgehend geladen.</div>
8   </body>
9   
10 </html>
```

(a) index.html

Anhang: Aufgabe 7 (Web-Anwendungen)

```
1 | const a = ["J", "P", "B"];
2 | const b = ["T", "K"];
3 | b.push(["K", "G"]);
4 | const c = [...a, ...b];
5 |
6 | let d = 1, e = 2, f = 3;
7 | [d, e, ...f] = [...a, ...b];
8 |
9 | console.log(d, f);
```

(a) Programm 1

```
1 | const f = (a, b) => {
2 |   return ((c) => a(b,c));
3 | };
4 |
5 | const a = f((a,b) => a + b, "1");
6 | console.log(a("2"));
```

(b) Programm 2