
UE - Zwischenklausur

Web-Technologien

07.12.2018

WS 18

Alpen-Adria-Universität Klagenfurt
Institut für Informatik-Systeme

Hilfsmittel: Unterlagen gestattet. Elektronische Geräte nicht gestattet.
Dauer: 60 Minuten
Hinweis: Jedes zusätzliche Blatt ist mit Namen und Matrikelnummer zu versehen

Matrikelnummer:
Familienname:
Studienrichtung:

Gruppe: ☐ Köpke (Mo) ☐ Köpke (Do) ☐ Schöffmann / Leibetseder (Mi)

Bitte beachten Sie:

Lesen Sie die Aufgabenstellung sorgfältig. Schreiben Sie leserlich. Verwenden Sie keinen Rotstift und keinen Bleistift. Für die Antworten ist Platz auf den Aufgabenseiten vorgesehen. Bei Platzmangel verwenden Sie bitte den Anhang. Sollten Sie Konzeptpapier benötigen, wenden Sie sich an die Aufsicht. Werden zu einer Aufgabe mehrere Lösungen angegeben, so gilt die Aufgabe als nicht gelöst. Es gelten die Prüfungsregeln des Instituts für Informatik-Systeme.

Bewertung		
Aufgabe	Mögliche Punkte	Erreichte Punkte
1.	12	
2.	12	
3.	21	
4.	15	
Summe:	60	

1. HTML & CSS (12 Punkte)

Gegeben sei das folgende HTML Dokument.

```
<html lang="de">
<body>
  <header>
    <h1>My Web Page</h1>
  </header>
  <div id="main">
    <nav>
      <ul>
        <li>Education</li>
        <li>Hobbies</li>
        <li>Blog</li>
      </ul>
    </nav>
    <div class="content">
      <section>
        <h1>Education</h1>
        Master program at <a href="https://www.aau.at">AAU.AT</a>.
      </section>
      <section>
        <h1>Contact</h1>
        <label for="emailField">Email: </label>
        <input type="email" id="emailField">
      </section>
      <section>
        <h1>Blog</h1>
        Read my Blog at <a href="http://myblog.at">myblog.at</a>
      </section>
    </div>
  </div>
  <footer>
    <ul>
      <li>Impressum</li>
      <li>Contact</li>
    </ul>
  </footer>
</body>
</html>
```

Schreiben Sie CSS Regeln um die nachfolgenden Anforderungen zu erfüllen. Dabei muss das Aussehen von nicht erwähnten Elementen unverändert bleiben.

- a. Die Schrift des Input Elements mit der ID „*emailfield*“ soll blau dargestellt werden. [2 Punkte]

- b. Alle Links, die auf sichere Seiten (https://) verweisen sollen den nachgestellten Hinweis *sicher* erhalten. Beispiel: „... Master program at AAU.AT (*sicher*)...“. [3 Punkte]

- c. Der Text der Überschrift der letzten Section soll rot, der Text der Überschrift der ersten Section grün ausgegeben werden. [3 Punkte]
- d. Das Menü soll links neben dem Inhalt ausgegeben werden und 20% der Breite des *main* div Elementes einnehmen. Entsprechend soll der Inhalt 80% einnehmen. [4 Punkte]

2. JavaScript I (12 Punkte)

Gegeben sei folgende HTML Datei:

```
<html>
  <head>
    <script type="text/javascript" src="sum.js" defer></script>
  </head>
  <body>
    <table id="myTable">
      <thead>
        <tr>
          <th scope="col">Produkt</th><th scope="col">Umsatz</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <th scope="row">P123</th><td class="umsatz">100</td>
        </tr>
        <tr>
          <th scope="row">P124</th><td class="umsatz">10</td>
        </tr>
        <tr>
          <th scope="row">P125</th><td class="umsatz">1000</td>
        </tr>
      </tbody>
      <tfoot>
        <tr>
          <th scope="row">Summe</th><td id="total"><input type="button"
value="berechnen" id="calcb"></td>
        </tr>
      </tfoot>
    </table>
  </body>
</html>
```

Implementieren Sie die JavaScript Datei *sum.js*, die folgende Funktionalität realisiert. Wenn der Nutzer auf den Button mit der ID *calcb* klickt wird der Button entfernt und stattdessen die Gesamtsumme aller Umsätze angezeigt.

Ihre Lösung muss für Tabellen mit beliebig vielen Zeilen (>0) funktionieren. Der HTML Code der Seite darf nicht verändert werden.

TIPP: Die zu summierenden Elemente können einfach über die CSS Klasse *umsatz* adressiert werden. Achten Sie jedoch darauf, die Elemente nur dann zu summieren, wenn sie Nachfahren des Elements mit der id *myTable* sind.

3. JavaScript II (21 Punkte)

- a) Was wird nach Ablauf des folgenden JavaScript Codes in der Konsole eines aktuellen Browsers ausgegeben? [5 Punkte]

```
function myFunction(a) {
    let i= function () {
        a++;
        console.log(a);
    }

    i();
    return(i);
}

let x = myFunction(10);
x();
x();
x = myFunction(5);
x(2);
```

Konsole: Zeile 1:	Zeile 2:	Zeile 3:	Zeile 4:	Zeile 5:
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50
51	52	53	54	55
56	57	58	59	60
61	62	63	64	65
66	67	68	69	70
71	72	73	74	75
76	77	78	79	80
81	82	83	84	85
86	87	88	89	90
91	92	93	94	95
96	97	98	99	100

- b) Gegeben sei folgendes JavaScript. Was wird auf der Konsole ausgegeben? [6 Punkte]

```
function A(b, c) {
  this.b = b;
  this.c = c;
  this.add = function() {
    return(this.b+this.c);
  }
}

A.prototype.mult = function() {
  return(this.b*this.c);
}

let a1 = new A(10,20);
console.log(a1.add());
console.log(a1.mult());

A.prototype.add = function() {
  return(this.b-this.c);
}

console.log(a1.add());
console.log(a1.mult());

function B(b,c,d) {
  A.call(this, b, c);
  this.d = d;
}

B.prototype = Object.create(A.prototype);
B.prototype.constructor = B;
B.prototype.div = function() {return this.mult() / this.d};

let b1 = new B(10,20,2);
console.log(b1.add());
console.log(b1.div());
```

Konsole: Zeile 1: Zeile 2: Zeile 3: Zeile 4: Zeile 5: Zeile 6:

- c) Gegeben sei folgender JavaScript Code, der auf einer HTML Seite mit zwei Input Elementen mit den Ids *idEl* und *descEl* mit der *defer* Option eingebunden ist.

```
class Product {
  constructor(idEl, descEl) {
    this.id = idEl.value;
    this.desc = descEl.value;
    this.idEl = idEl;
    this.descEl = descEl;
  }

  syncData(e) {
    console.log(this);
    console.log(e.target.id);
    this.id = this.idEl.value;
    this.desc = this.descEl.value;
    console.log(this.id + " " + this.desc);
  }

  register() {
    this.idEl.addEventListener("change", this.syncData);
    this.descEl.addEventListener("change", this.syncData);
  }
}

let p = new Product(document.getElementById("idEl"), document.getElementById("descEl"));
p.register();
```

Das Script soll ein neues Produkt Objekt *p* erzeugen, dessen Instanzvariablen *id* und *desc* bei Änderungen der Input Elemente *idEl* und *descEl* mitgeändert werden. Leider klappt dies nicht wie erwartet. Ändert man z.B. das Feld *idEl* wird in der Konsole folgendes angezeigt:

- a. `<input id="idEl" type="text"></input>`
- b. *idEl*
- c. *Unable to get property 'value' of undefined or null reference*

- a) Erklären Sie das Verhalten [5 Punkte]:

- b) Lösen Sie das Problem und geben Sie den geänderten Code hier an [5 Punkte]:

4. Node.js (15 Punkte)

Erweitern Sie das gegebene Code-Fragment, welches mit Node.JS ausgeführt werden soll um folgende Funktionalitäten: Bei Aufruf der URL `/show/articles/id` mittels *GET request*, soll der Artikel mit der *id* im JSON Format und dem HTTP Status 200 zurückgegeben werden. Sollte es keinen solchen Artikel geben muss ein Fehler 404 zurückgegeben werden.

Der Zugriff auf die Artikel Datenbank muss von Ihnen nicht implementiert werden. Verwenden Sie dazu die Methode `getArticle(id)` des bereits geladenen Moduls `articleservice`.

Die Methode `getArticle(id)` gibt als Rückgabe ein Promise Objekt zurück. Das Promise wird rejected, falls es keinen Artikel mit der *id* gibt. Andernfalls wird das Promise fulfilled und das Artikel Objekt als erster Parameter an die Callback Funktion übergeben.

```
const app = require("express")();
const articleservice = require("./articleservice.js");
const port = 3000;

app.get("/", (request, response) => {
  response.send("Welcome");
});

// Add your code here. Use articleservice.getArticle() to load an article
// asynchronously.

app.listen(port, () => {
  console.log("Server is running at http://127.0.0.1:" + port + " ...");
});
```

Ihre Lösung:

