

Name:

Matrikelnummer:

Webapplikationen  
20.02.2020

Name:

Vorname:

Matrikelnr.:

Semester:

Aufgabe	1	2	3	4	5	6	7
Punkte							

Summe (max. 90 Punkte):

Note:

Erlaubtes Hilfsmittel: ein einseitig und handschriftlich beschriebenes DIN-A4-Blatt

**Viel Erfolg!**

Name:

Matrikelnummer:

## 1. Aufgabe (HTML & DOM)

15 Punkte

a) Skizzieren Sie den zum HTML-Code gehörigen DOM-Tree.

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Klausur WS 19/20</title>
  </head>
  <body>
    <h1>Login page</h1>
    <form action="ageCheck.php" method="get">
      <input type="text" name="uname">
      <br>
      <input type="text" name="age">
      <br>
      <input type="submit">
    </form>
    <p>
      You are <span id="checkRes"></span>
      allowed to use this service.
    </p>
    <script>
      // TODO; impl.!
    </script>
  </body>
</html>
```

Name:

Matrikelnummer:

**b)** Wie lautet der Query-String, wenn Sie Ihren Namen sowie Ihr Alter in die beiden Formularfelder eingeben und das Formular absenden?

---

**c)** Der obige HTML-Code enthält ein `<script>`-Element mit einem TODO. Implementieren Sie stattdessen folgendes:

Navigieren Sie dazu zunächst per JavaScript mittels DOM-Scripting zum `<span>`-Element und ändern mit dessen Hilfe den umgebenden Satz ab in „You are **not** allowed to use this service.“

Den Fettdruck erzielen Sie durch programmatisches Ändern des Styles!

Fügen Sie sodann dem übergeordneten `<p>`-Element einen Link hinzu. Die URL des Links sei „secretArea.php“, der Link-Text soll lauten: „Attention, access classified“.

Name:

Matrikelnummer:

## **2. Aufgabe (Programmierkenntnisse in JavaScript)**

**10 Punkte**

Schreiben Sie eine JavaScript-Funktion `vor9()`, die für ein als Parameter übergebenes Eingabefeld mit Zahlen ein neues Array zurückliefert, welches alle Werte aus dem Feld enthält, die **vor** der ersten **9** stehen, also z.B.:

```
vor9 ( [1, 2, 9, 1] )    → [1, 2]
vor9 ( [3, 1, 8, 9] )    → [3, 1, 8]
vor9 ( [1, 9, 9] )        → [1]
vor9 ( [1, 7, 9823] )    → [1, 7, 9823]    // Zahl 9 muss alleine stehen
```

Name:

Matrikelnummer:

### 3. Aufgabe (Fehler finden)

15 Punkte

- a) Finden und korrigieren Sie mindestens zehn Fehler (syntaktisch oder semantisch). Beschreiben Sie ggfs. kurz das Problem, falls der Fehler nicht offensichtlich ist (wie z.B. bei einem vergessenen Semikolon).
- b) Ergänzen Sie die Zeilen a) – e) mit den geforderten Angaben.

#### CSS-Code:

```
...
#lightbox.show {
    display: block
}
#lightbox.hide {
    display: not;
}
...
```

#### HTML-Code:

```
...
<div class="thumbs">
    
</div>
<div class="lightbox" class="hide">
    <a id="close" href="#">X</a>
    <img source="" id="targetImage">
</p>
```

#### JavaScript-Code:

```
let lightbox = document.getElementById("lightbox");
let images   = document.getElementsByClassName("thumb");

// Bei Klick wird Lightbox geoeffnet
for (const i=images.length-1; i >= 0; i++ {
```

- a) Bitte fehlenden Code ergänzen:

```
}
```

```
document.getElementById("close")
```

```
.addEventListener(_____);
```

Name:

Matrikelnummer:

function openLightbox(evt) {  
    b) Bitte fehlenden Code ergänzen:

\_\_\_\_\_  
    loadContent(evt.target);  
}

Function closeLightbox(Event evt) {  
    evt.preventDefault();

c) Erklären Sie kurz, was ohne die vorherige Zeile passieren würde:

\_\_\_\_\_  
    lightbox.className = "hide";  
}

function loadContent(img) {  
    const imgSrc = img.dataset.src;  
    const targetImg = document.getElementById('targetImage')  
    targetImg.setAttribute('src', imgSrc);  
}

d) Erklären Sie, was die Funktion loadContent() macht:

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

e) Erklären Sie, was auf dieser Seite passiert bzw. welchem Zweck der JS-Code dient.

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Name:

Matrikelnummer:

**4. Aufgabe (JSON)**

**10 Punkte**

Stellen Sie die folgende Tabelle im JSON-Format dar!

Hochschule Fulda  
Leipziger Straße 123  
36037 Fulda

Mitarbeiter	Durchwahl	Zuständigkeit
<i>D. Biezā</i>	<i>3050</i>	<i>AI-Support, Betreuung Software-Labore, Lehre</i>
<i>C. Pape</i>	<i>379</i>	<i>Betreuung WI-Labor, Virtualisierung</i>
		Dekanin: <i>B. Bomsdorf</i>

Name:

Matrikelnummer:

### **5. Aufgabe (Rekursion)**

**10 Punkte**

Schreiben Sie in JavaScript eine rekursive Funktion, die für einen übergebenen String prüft, ob ein sog. Palindrom vorliegt (ein String der von vorne und von hinten gelesen je dasselbe ergibt, z.B. ABBA, Otto, Rentner, Hannah, usw.). Wenn Sie nicht direkt mit Strings arbeiten wollen, dürfen Sie der Einfachheit halber auch annehmen, dass der String bereits als Array von Zeichen vorliegt (z.B. ['O', 't', 't', 'o']). Bei letzterem Beispiel würde *true* geliefert werden (Groß-/Kleinschreibung muss nicht beachtet werden).



Name:

Matrikelnummer:

## 6. Aufgabe (Objekte und Klassen)

**10 Punkte**

Schreiben Sie eine Klasse „Rechteck“, die von „Form“ erbt (letztere bitte auch schreiben). Jede Form hat ein Attribut `farbe` sowie die Methode `info()`. Rechtecke haben zusätzlich die Methode `flaeche()` und nehmen im Konstruktor Werte für `farbe`, `hoehe`, `breite`, `positionX` und `positionY` entgegen. Die Methode `flaeche()` gibt den errechneten Flächeninhalt des Rechtecks zurück und die Methode `info()` schreibt die Art der Form, den Flächeninhalt und die Farbe in die Konsole.

Instanzieren Sie anschließend ein grünes Rechteck an Position (50, 100) mit der Breite 40 und der Höhe 30. Lassen Sie zudem noch den Flächeninhalt und die Farbe auf der Konsole ausgeben.

Hinweise: Es muss nichts auf ein Canvas gezeichnet werden und der Benutzer übergibt alle Variablen im korrekten Format – es ist also ausnahmsweise kein Error-Handling notwendig.

Name:

Matrikelnummer:

## 7. Aufgabe (Ajax und Node.js)

20 Punkte

Eine Website soll um ein Formular erweitert werden, mit dem man sich für eine LAN-Party anmelden kann. Zur Visualisierung ist unten der Screenshot eines ersten Prototyps gezeigt.



The screenshot shows a web browser window with the address bar displaying 'localhost:63342/Beispiele/'. The page title is 'Anmeldung LAN-Party'. The form consists of two text input fields, one labeled 'Name:' and one labeled 'Team:'. Below these fields are two buttons: 'Zurücksetzen' (Reset) and 'Anmelden' (Login).

Implementieren Sie mit Ajax und Node.js eine Webanwendung, welche im Query-Parameter den Wert des Eingabefelds namens `name` (mit dem Namen des/der Angemeldeten) sowie den Wert des Eingabefelds namens `team` (mit dem Team, falls vorhanden) entgegennimmt und beides durch Semikolon getrennt an das Ende einer Datei `anmeldungen.txt` abspeichert (pro Anmeldung eine Zeile). Falls die Datei noch nicht existiert, soll sie erzeugt werden.

Nach dem Speichern soll auf der Website unterhalb der Buttons (ohne Neuladen der HTML-Seite) eine Nachricht erscheinen, dass man erfolgreich zur LAN-Party angemeldet wurde. Dabei muss diese Nachricht natürlich vom Server kommen.

Setzen Sie zunächst den *clientseitigen* Teil um, wobei Sie lediglich den JavaScript-Code mit dem Ajax-Request (d.h. XMLHttpRequest oder `fetch()`) sowie die für die Programmierung nur absolut notwendigen HTML-Elemente aufzuschreiben brauchen.

Name:

Matrikelnummer:

Ergänzen Sie für den *serverseitigen* Teil nun noch den folgenden JavaScript-Code.<sup>1</sup>

```
const http = require('http');
const url = require('url');
const fs = require('fs');

const server = http.createServer();

server.on('request', function(req, res) {
```

```
});
```

```
server.listen(8080);
```

---

<sup>1</sup> Zur Erinnerung hier die Funktionen zum asynchronen Lesen und Schreiben von Dateien:

```
fs.readFile(file, callback);
fs.appendFile(file, content, callback);
fs.writeFile(file, content, callback);
```

Name:

Matrikelnummer:

**Zusatzblatt**