

Gedächtnisprotokoll

Webtechnologien WS 17/18

2. Klausur vom 26.03.2018

Es gab vier Klausurteile auf insgesamt 19 Seiten: Frontend, Javascript, Kommunikation, Server. Zu jedem Teil gab es 4-5 Multiple Choice Aufgaben, ca. 5 offene Fragen und 2-3 Programmieraufgaben.

Jede MC-Aufgabe bestand aus 4 Aussagen, die jeweils mit "wahr" oder "falsch" beantwortet werden mussten. Jede MC-Aufgabe gab genau einen Punkt. Um einen Punkt zu bekommen mussten alle 4 Aussagen korrekt beantwortet werden. Es konnten also mehrere Antworten wahr oder falsch sein.

Die offenen Fragen gaben jeweils 1 oder 2 Punkte. Sie konnten i.d.R. in ein oder zwei Sätzen beantwortet werden.

Die Programmieraufgaben bestanden teilweise daraus eigenen Code zu schreiben oder anzugeben was der Output von vorgegebenem Code ist. Bei eigenem Code musste man i.d.R. ca. 5 Zeilen Code schreiben.

Anbei sind diejenigen Aufgaben, an die ich mich gut erinnern kann. Die Fragen sind nicht wortwörtlich, aber sinngemäß. Wo ich mich nicht an Aussagen erinnere habe ich sie leer gelassen.

Fontend

MC

Wer ist/sind der/die Erfinder der Kerntechnologien (HTTP und HTML) des World Wide Web?

Wahr	Falsch	Aussage
		Tim Berners-Lee
		Vint Cerf
		Bob Kahn
		Alan Cooper

Welche Aussagen sind wahr/falsch?

Wahr	Falsch	Aussage
		Im Web 2.0 werden Daten von allen erzeugt (user-generated Content)
		Web 3.0 ist für Maschinen
		Web2.0 steht für semantische Webseiten
		Im Web 1.0 wurden HTML-Dokumente über Hyperlinks mit einander verbunden

Welche Aussagen sind wahr/falsch?

Wahr	Falsch	Aussage
		In HTML5 gibt es neue Elemente für das Layout von Webseiten.
		HTML5 fügt neue Elemente hinzu um die Semantik von Webseiten zu verbessern

Offene Fragen

Was unterscheidet adaptives und flüssiges Design?

Was sind die Vorteile eines vorgefertigtem Rastersystems?

Programmierung

Gegeben ist der folgende Code:

```
<button class="buttonStyle" name="myButton">Klick</button>
```

Schreibe reinen CSS-Code, der in ein Style-Element eingebettet werden kann, der folgendes tut:

Wenn man auf den Button klickt, soll der Hintergrund des Buttons grün werden. Sonst soll der Hintergrund des Buttons blau sein. Hinweis: die *active* Eigenschaft könnte helfen.

Positionierung

```
<html>
<head>

<style type="text/css">
body,p { margin: 0;
padding: 0;
}
body {
border: 1px dashed red;
width: 800px;
display:grid;
grid-template-columns: 1fr 3fr;
}
p {
width: 200px;
height: 200px;
border: 1px solid black;
}
#div_eins { position: relative;
height: 400px;}
#eins {position: absolute;
bottom: 0px;
left: 0px;
}
#zwei {
position: relative;
top: 600px;
left: 200px;
}
#drei { float: right;}
#vier{ position: fixed;
right: 0px;
top: 400px;
}
</style>
</head>

<body>
<div id="div_eins">
<p id="eins">eins</p>
<p id="zwei">zwei</p>
</div>
<div id="div_zwei">
<p id="drei">drei</p>
<p id="vier">vier</p>
</div>
</body>
</html>
```

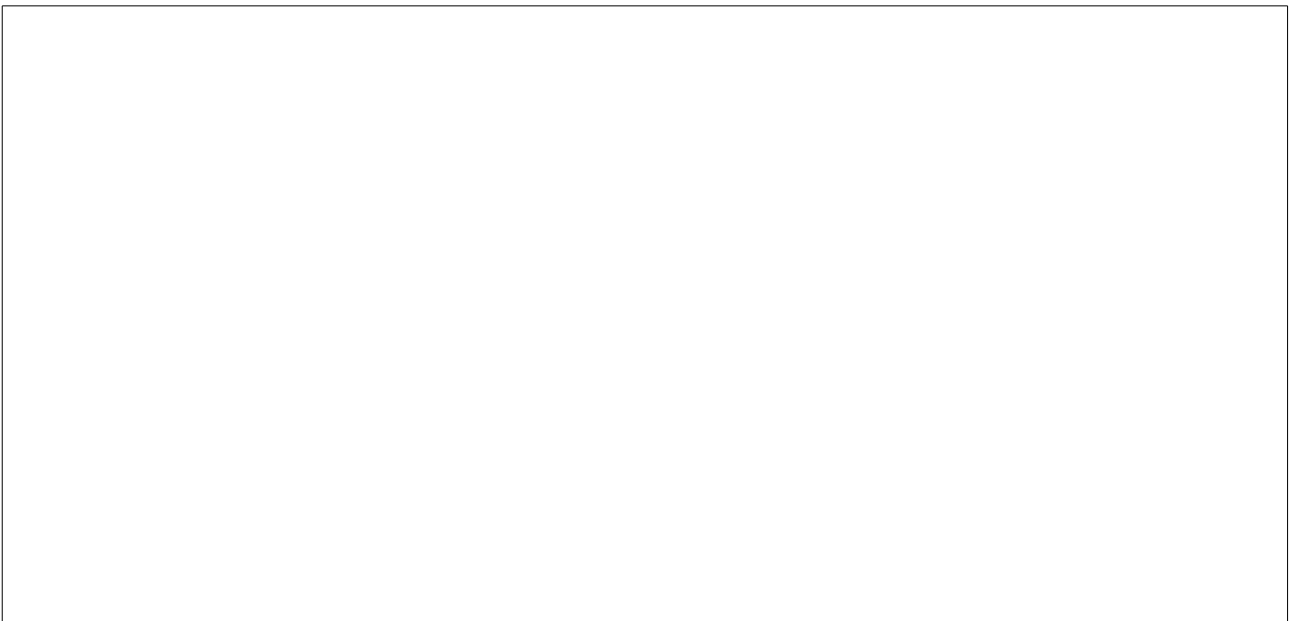
Zeichnen Sie die Rahmen/Border und Text der p-Elemente in die Vorlage ein. Die Vorlage zeigt den gesamten Viewport.

200px	400px	600px	800px

Gegeben ist der folgende Quellcode:

```
<html>
  <head>
    <meta charset="utf-8">
  <style type="text/css">
body{
  margin: 0px;
}
main {
  margin: 1cm auto;
  width: 50%;
  padding: 1cm;
  border: 1px solid black;
}
nav {
  border: 1px solid black;
}
</style>
</head>
<body>
  <footer>Service-centric Networking</footer>
  <nav>
    <label>Name: <input type="text" name="name"></label><br>
    <label>Vorname: <input tpye="text" name="vorname"></label>
  </nav>
  <main>
    <h1>Willkommen Student!</h1>
    <p>Auf dieser Webseite findest du einige Informationen über
dich...</p>
  </main>
</body>
</html>
```

Zeichnen Sie die Webseite in die Vorlage ein, wie sie nach dem vollständigen Parsen im Browser dargestellt wird. Die Vorlage beinhaltet den gesamten Viewport. Es muss nicht exakt sein, aber die Proportionen sollen ersichtlich sein.



Javascript

MC

Welche Aussagen sind wahr/falsch?

Wahr	Falsch	Aussage
		In einer Funktion mit var deklarierte Variablen sind global sichtbar.
		In einem Block (innerhalb geschweifter Klammern) deklarierte Variablen sind nur in diesem Block sichtbar
		Globale Variablen sind in einer Funktion sichtbar, wenn diese nicht in der Funktion überschrieben sind.
		Eine Funktion definiert einen neuen Sichtbarkeits-bereich.

Welche Aussagen sind wahr/falsch?

Wahr	Falsch	Aussage
		Das DOM ist hierarchisch aufgebaut.
		DOM steht für "Document overview mode"
		DOM ist ein plattformspezifisches Interface
		DOM ist ein Interface der Änderung von Dokumenten

Welche Datentypen gibt es bei JavaScript?

Wahr	Falsch	Aussage
		Char
		String
		Int
		null

Offene Fragen

Geben Sie drei Möglichkeiten an Events an ein HTML-Element zu binden (Beispiele sind möglich)

Vererbung

Es gibt Lehrveranstaltungen, welche die Eigenschaften name, raum, zeit und studenten enthalten, die für jede Lehrveranstaltung unterschiedliche Werte haben. Außerdem gibt es zwei Methoden: registrieren(student) und abmelden(student). Erstere fügt dem studenten-Array einen weiteren Namen hinzu und letztere entfernt die Person wieder aus der Liste. Es existieren zwei Arten von Lehrveranstaltungen: Seminar und Vorlesung, die von Lehrveranstaltung erben. Seminar überschreibt die registrieren-Methode jedoch dahingehend, dass ab einer Anzahl größer als 10 registrierte Studenten eine Fehlermeldung zurückgegeben wird, da die Veranstaltung voll besetzt ist.

Sie erstellen zwei Objecte var webtechnologien = new Vorlesung() und var isol = new Seminar().

Geben Sie für den angegebenen Use-Case die vollständige Prototypen-Kette an. Geben sie zu jedem Objekt auch die entsprechenden Eigenschaften mit an.

Schreiben sie kein Javascript! - eine Zeichnung/Beschreibung reicht.

Worauf sollte der constructor von Seminar.prototype verweisen?

Wieso kann es sinnvoll seine Objekte mit Object.create(null) zu erstellen?

Programmierung

```
var myArray = [1,2,3,4,5];
/* Nr. 1*/
console.log(myArray[1]);

for(var i = 5; i > 0; i--) myArray.push(i);

var red = myArray.reverse().filter(function(element){
    if(element - 4 >= 0) {
        // TRUE - Element bleibt in Liste
        return true;
    }
}).reduce(function(prevEl, element){
    return prevEl + element;
});

/* Nr. 2*/
console.log(red);

function textMake(str, len, ch){
    if(ch == undefined) ch = '0';
    len = len - str.length;
    for(var i = 0; i < len; i++){
        str = ch + str;
    }
    return str;
}

/* Nr. 3*/
console.log(textMake("Echo", 5, "#"));
/* Nr. 4*/
console.log(textMake("42", 5));
/* Nr. 5*/
console.log(textMake("Webtech", 6, "?"));
```

Gegeben ist der obige JavaScript Code, der 5 Konsolen-Ausgaben enthält. Geben Sie unten neben der jeweiligen Nummer an was auf der Konsole ausgegeben wird:

- 1.
- 2.
- 3.
- 4.
- 5.

Kommunikation

MC

Was sind Beschränkungen von REST?

Wahr	Falsch	Aussage
		Client-Server
		Cache
		Layered-System
		Uniform Interface

Wozu werden Cookies verwendet?

Wahr	Falsch	Aussage
		Nutzer tracken
		Sitzungsmanagement
		Fehlerkorrektur
		Überlastungskontrolle

Welche Aussagen sind wahr/falsch?

Wahr	Falsch	Aussage
		Cookies gehören zu HTTP
		Es gibt Cookies, auf die nicht mit JavaScript zugegriffen werden kann.
		Mit Cookies kann man Nutzer wiedererkennen

Woran erkennt ein Client (Browser) bei HTTP 1.1 welche Antwort zu welcher Anfrage gehört wenn er mehrere in einer Pipeline geschickt hat?

Wahr	Falsch	Aussage
		An der Anfrage URL
		Header-Feld
		StatusCode
		Antwort-Body

Offene Fragen

Welche zwei Möglichkeiten gibt es bei AJAX Daten abzurufen, sobald diese aktualisiert wurden?

Heutzutage gibt es viele verschiedene Geräte (Desktop, Tablet, Smartphone) mit unterschiedlichen Bedürfnissen bzgl. der gelieferten Daten. Erklären sie kurz anhand eines Beispiels wie man dem Rechnung tragen kann. Wie kommen die richtigen Geräte an die richtigen Daten?

Geben Sie den Aufbau einer HTTP-Anfrage und -Antwort an. Benutzen Sie wenn notwendig dabei die folgenden Begriffe: Header-Felder, Methode, StatusCode, StatusText, Body, Leerzeile, Pfad

Anfrage

Antwort

Sie implementieren eine REST-Schnittstelle. Der Server soll folgende Funktionen ausführen können:

- Alle Artikel ausgeben
- Einen neuen Artikel hinzufügen
- Einen bestimmten Artikel ausgeben
- Einen bestimmten Artikel ändern
- Einen bestimmten Artikel löschen

Andere Funktionen kann der Server nicht.

Geben sie sinnvolle URI's an und die auf der URI ausführbaren HTTP-Befehle um die Schnittstelle zu realisieren!

URI

HTTP-Befehle

Was ist schlecht an der folgenden Abfrage? `POST /artikel/create/new`

Server

Offene Fragen

Nenne zwei Vorteile des Express-Frameworks gegenüber reinem Node.js.

Geben Sie zwei Möglichkeiten an Lesevorgänge auf einer MongoDB-Datenbank zu beschleunigen. (Es wird immer auf die selbe Datenbank und die selben Daten zugegriffen).

Welche Schritte sind notwendig um eine MongoDB-Datenbank mittels Mongoose in Node.js einzubinden?

Wofür steht MVC? Was sind die Aufgaben der einzelnen Komponenten?

Programmierung

Gegeben ist ein Express server:

```
var express = require('express');  
var app = express();  
  
var artikel = [];
```

```
app.listen(8080);
```

Schreiben Sie eine Middleware, die für jede Anfrage an den Server folgendes auf der Konsole ausgibt und danach weiterreicht:

Aktuelles Datum

HTTP-Methode

Pfad

Zusätzlich soll der Server bei einer DELETE Anfrage der Ressource `/artikel/1` das erste Element der Artikel-Ressource auf *null* setzen.

Anmerkung zum Protokoll:

Der folgende Quellcode ist nicht vollständig (und wahrscheinlich auch nicht ausführbar), zeigt aber die entsprechende Komplexität relativ gut. Prinzipiell gab es drei Server. Ein Loadbalancer, der die Anfrage an einen der beiden anderen weiterleitet hat (Apfel an Server1 und Zitronen an Server2), evtl. mit einem x-http-method-override. Die beiden anderen Server haben die Anfrage dann bearbeitet und die Antwort mit einer weiteren Funktion (die hier nicht angegeben ist) zusammengesetzt. Die Antworten gingen zurück an den Loadbalancer, der sie an den Client weiterleitet hat. Der tatsächliche Code in der Klausur war also noch etwas komplexer als dieser.

Gegeben ist der folgende NodeJS-Server:

```
var http = require('http');
var url = require('url');

var server1 = new http.Server();
var server2 = new http.Server();

var apples = [];
var lemons = [];

http.createServer((request, response)=>{
  var httpMethod = request.method;
  var fullUrl = url.parse(request.url);
  if(request.headers['x-http-method-override']){
    httpMethod = request.headers['x-http-method-override'];
  }

  if(fullUrl.query.obst == 'apfel'){
    http.request({
      method: httpMethod,
      host: "127.0.0.1",
      path: fullUrl,
      port: 7000
    });
  }else if(fullUrl.query.obst == 'zitrone'){
    http.request({
      method: httpMethod,
      host: "127.0.0.1",
      path: fullUrl,
      port: 9000
    })
  }else{
    response.statusCode = 500;
    response.end("Interner Fehler");
  }
}).listen(8080, 127.0.0.1);

server1.addListener('request', function(req, res){
  if(req.method == 'DELETE') {
    apples.pop();
    res.statusCode = 200;
    res.end("Element erfolgreich gelöscht");
  } else if (req.method == 'POST') {
    apples.push(url.parse(req.url).path);
    res.statusCode = 200;
    res.writeHead({'location': url.parse(req.url).path + '?obst=apfel'});
    res.end("Element erfolgreich hinzugefügt");
  }else {
    /* Kann mich nicht erinnern ob hier noch etwas war. Wenn, dann war es
    nicht so wichtig.*/
  }
});
```

```

server2.addListener('request', function(req, res){
  if(req.method == 'DELETE') {
    res.statusCode = 400;
    res.end("Nicht erlaubt");
  } else if (req.method == 'POST') {
    lemons.push(url.parse(req.url).path + "?obst=zitrone");
    res.statusCode = 200;
    res.write(lemons.join(" UND "));
    res.end("Element erfolgreich hinzugefügt");
  }else {
    /* Kann mich nicht erinnern ob hier noch etwas war. Wenn, dann war es
nicht so wichtig.*/
  }
});

server1.listen(7000);
server2.listen(9000);

```

Geben Sie für die folgenden HTTP-Anfragen die entsprechende Antwort-Body an. Geben Sie auch die Header-Felder an, die nicht automatisch bei jedem Antwort generiert werden. Hinweis: Die HTTP-Anfragen sind nicht syntaktisch korrekt, aber alle notwendige Information ist enthalten.

Anfrage

Antwort

```

DELETE
/Lugano/13?obst=apfel
Host: 127.0.0.1:8080

```

```

POST
/Lugano/87?obst=zitrone
Host: 127.0.0.1:8080

```

```

DELETE
/Stumpf/87?obst=apfel
Host: 127.0.0.1:8080
x-http-method-override: POST

```

```

POST
/Lugano/13?obst=zitrone
Host: 127.0.0.1:8080

```

```

PUT
/Lugano/74?obst=apfel
Host: 127.0.0.1:8080
x-http-method-override:POST

```