

## **In Moodle erbrachte Leistungen**

Einleitungsbericht	2
Reflexionsbericht	4
Lernkontrolle: Grundlagen des Web	6
Lernkontrolle: HTML und CSS	7
Lernkontrolle: Responsive Webdesign	10
Lernkontrolle: JavaScript	12
Lernkontrolle: Serverseitige Webentwicklung	17
Onlinetest zum Projekt	22

# Einleitungsbericht

## **Vollständiger Name**

Tamino Fischer

## **Datum**

Montag, 1. Oktober 2018, 15:23

## **Technische Grundlagen des WWW**

Schon mal was damit gemacht

## **Strukturierung von Webseiten mit HTML**

Interessiert mich brennend

## **Gestaltung von Webseiten mit CSS**

Interessiert mich brennend

## **JavaScript und und Document Object Model**

Blutiger Anfänger

## **Serverseitige Webentwicklung (Sprache egal)**

Kenne ich ganz gut

## **Erläuterungen zu deinem Vorwissen**

Durch mein bisheriges Studium an der HSKA:

Programmiersprachen:

- C# , Java, PHP, Kotlin

Frameworks:

-Spring

Datenbanken:

-Oracle, MySQL, Mongo

OS, Netzwerke, IT Security

- Grundlagen

Architektur:

REST, Monolithen, Microservices

Angewendet an: Uniprojekt (Backend eines Webshops mit Microservices)

## **JavaScript als Programmiersprache allgemein**

Interessiert mich brennend

## **Verwendung von JavaScript im Browser**

Interessiert mich brennend

## **Serverseitige Webentwicklung mit Java**

Interessiert mich brennend

## **Welche Themen interessieren dich besonders?**

Super Fach, Super Prof.

WEITER SO!

## **Welche persönlichen Lernziele setzt du dir für diese Vorlesung?**

Mehr über Frontendentwicklung erfahren und diese mit einem Backend verbinden

# Reflexionsbericht

## Vollständiger Name

Tamino Fischer

## Datum

Mittwoch, 14. November 2018, 11:02

## Welche neuen Inhalte (nicht Fähigkeiten) hast du durch die Vorlesung gelernt?

- HTML
- CSS
- Responsive Websdesign
- JS
- Backend
- Gruppenarbeit

## Welche neuen Fähigkeiten (z.B. in der Entwicklung, der Werkzeugnutzung oder im Team) hast du erworben?

- Umgang mit HTML, CSS, Responsive Websdesign, JS
- Versionsverwaltungssysteme
- tieferes Verständniss meiner IDE
- Gruppenarbeit

## Welches Thema hast du deiner Meinung nach am besten verstanden?

Serverseitige Webentwicklung und Javascript (auch wenn ich bei Letzterem mir noch die neue Schreibweise angewöhnen muss).

## Welches Thema hast du deiner Meinung nach am schlechtesten verstanden?

CSS, das liegt aber daran, dass viele Anweisungsbezeichnungen "auswendig" gelernt werden müssen. Abhilfe verschafft hierbei natürlich das Internet.

## Welches Thema fandest du war am schwersten zu lernen?

CSS, da man viele Anweisungen "auswendig" lernen muss.

## Warum sind dir andere Dinge leichter gefallen?

Alle alle Themen, die ein Verständnis erfordern.

## Wie bewertest du eure Gruppenarbeit? Wer hat welche Aufgaben übernommen und wie ist die Zusammenarbeit gelaufen?

Die Gruppenarbeit verlief trotz unserer Gruppengröße sehr gut. Da wir auch in einem anderen Fach bereits als Gruppe zusammen gearbeitet haben, waren wir schnell ein Team und jeder wusste um die Stärken und Schwächen der Anderen.

Meine Aufgaben:

- Technologieauswahl (Spring, Intelij, Postman etc)
- Grundlegende Architektur des Systemes
- Architektur der Schnittstellen
- Anlegen des Basisprojektes (Spring WEB, Spring Data, Spring DEV, Spring MVC)

- Verbindung der DB zusammen mit Benjamin
- HTML, CSS, Javascript, Java in meinen Bereichen
- Template
- Erstellung, Speicherung, Anzeige von einer Liste aus individuellen Nahrungsergänzungsmitteln
- Homebereichsnavigation
- REST API Grundstruktur
- Servlet API Grundstruktur
- Grundsatzüberlegungen mit Benjamin (in welche Richtung gehen wir, welche Technologie verwenden wir wie etc.)

Benjamin Kanzler:

- DB Auswahl
- Verbindung der DB zusammen mit Benjamin
- Hibernate Konfiguration
- HTML, CSS, Javascript, Java in seinen Bereichen
- Grundsatzüberlegungen mit Tamino (in welche Richtung gehen wir, welche Technologie verwenden wir wie etc.)
- ... (siehe Benjamins) Reflexion

Sven Hornung:

- HTML, CSS, Javascript, Java seinen Bereichen

Moritz Kuttler:

- HTML, CSS, Javascript, Java in seinen Bereichen

Fabian Lieb:

- HTML, CSS, Javascript, Java in seinen Bereichen:

### **Sonstige Anmerkungen**

Bester Dozent! Ein wahrer Meister.

WEITER SO!

# Lernkontrolle: Grundlagen des Web

## Frage 1

Vereinfacht gesagt beschreibt das „World Wide Web“ den Abruf von Webseiten über das HTTP-Protokoll.

## Antwort 1

Wahr

## Frage 2

In welchem Jahr veröffentlichte Tim Bernes-Lee den Artikel „Information Management: A Proposal“, indem er das Web erstmals beschreibt?

- » 1979
- » 1983
- » 1989
- » 1993

## Antwort 2

1989

## Frage 3

Ordne jeder der genannten Webtechnologien ihre Funktion zu:

{HTTP; URL; HTML; CSS; WebSockets} -> {Alternatives Protokoll für die Kommunikation mit dem Server; Adressierung von Inhalten im World Wide Web; Beschreibungssprache für den Inhalt Webseiten; Beschreibungssprache für die Darstellung von Webseiten; Übertragungsprotokoll zum Abruf von Webseiten}

## Antwort 3

- » HTTP -> Übertragungsprotokoll zum Abruf von Webseiten
- » URL -> Adressierung von Inhalten im World Wide Web
- » HTML -> Beschreibungssprache für den Inhalt Webseiten
- » CSS -> Beschreibungssprache für die Darstellung von Webseiten
- » WebSockets -> Alternatives Protokoll für die Kommunikation mit dem Server

## Frage 4

Ein Anwender besucht einen Webshop und legt ein paar Artikel in seinen Warenkorb. Wo sollte der Warenkorb vor der Bestellung zwischengespeichert werden?

- » Im Browser
- » Auf dem Server

## Antwort 4

Im Browser

# Lernkontrolle: HTML und CSS

## Frage 1

Welche der folgenden Aussagen ist richtig?

- » CSS ist eine Erweiterung von HTML um weitere Tags, Elemente und Anweisungen
- » HTML ist eine Auszeichnungssprache für Dokumente im World Wide Web
- » CSS beschreibt das Aussehen einer Seite, HTML ihren Inhalt
- » HTML ist eine Auszeichnungssprache für Webseiten und maschinell auswertbare Datensätze

## Antwort 1

- » HTML ist eine Auszeichnungssprache für Dokumente im World Wide Web
- » CSS beschreibt das Aussehen einer Seite, HTML ihren Inhalt

## Frage 2

Vervollständige den folgenden Lückentext

```
<html>
  <head><title>Alfred J. Kwak</title></head>
  <body>
    _____Alfred J. Kwak</h1>
    <p>
      <b> Alfred J. Kwak _____ ist eine Ente und Hauptfigur der gleich-
      namigen Zeichentrickserie aus den Jahren 1989 und 1990. Die Serie, die
      1990 in Deutschland ausgestrahlt wurde, gewann viele Preis - darunter
      auch die <a href="kamera.html">Goldene Kamera_____
    _____
    _____
      Henk ist ein Maulwurf und anerkannter Bergbau-Experte, der Alfred nach
      dem Tod dessen Eltern aufzieht. Alfred betrachtet ihn als seinen Vater
      und besten Freund.
    </p>
    <h2>Hauptfiguren</h2>
    _____
    _____Alfred J. Kwak_____
    _____Henk_____
    _____Winnie Wanna_____
    _____Professor Paljas_____
  </ul>
</body>
</html>
```

## Antwort 2

Teil 1: <h1>; Teil 2: </b>; Teil 3: </a>; Teil 4: </p>; Teil 5: <p>; Teil 6: <ul>; Teil 7: <li>; Teil 8: </li>;  
Teil 9: <li>; Teil 10: </li>; Teil 11: <li>; Teil 12: </li>; Teil 13: <li>; Teil 14: </li>

## Frage 3

Was ist der Unterschied zwischen einem Block-Element und einem Inline-Element?

- » Block-Elemente belegen immer die maximale Breite und stehen untereinander, Inline-Elemente nicht
- » Inline-Elemente belegen immer die maximale Breite und stehen untereinander, Block-Elemente nicht
- » Inline-Elemente sind einfacher mit CSS auswählbar als Block-Elemente

**Antwort 3**

Block-Elemente belegen immer die maximale Breite und stehen untereinander, Inline-Elemente nicht

**Frage 4**

Welche der folgenden HTML-Elemente sind Block-Elemente?

- » <h1>
- » <h2>
- » <img>
- » <p>
- » <head>
- » <a>
- » <div>

**Antwort 4**

- » <h1>
- » <h2>
- » <p>
- » <div>

**Frage 5**

Welche der folgenden HTML-Elemente sind Inline-Elemente?

- » <img>
- » <header>
- » <ul>
- » <span>
- » <b>
- » <p>
- » <div>

**Antwort 5**

- » <img>
- » <ul>
- » <span>
- » <b>

**Frage 6**

Welche Funktion hat das HTML-Element <header>?

- » Es ist ein Block-Element, das allgemein den Kopfbereich eines Layouts kennzeichnet.
- » Es handelt sich um die erste Zeile mit den Überschriften einer Tabelle.
- » Es definiert den Kopfbereich einer HTML-Datei mit nicht sichtbaren Metadaten.



### **Antwort 6**

Es ist ein Block-Element, das allgemein den Kopfbereich eines Layouts kennzeichnet.

### **Frage 7**

Welche Elemente werden durch den CSS-Selektor `p.info` ermittelt?

- » Alle Absätze mit der Klasse "info"
- » Alle Elemente der Klasse "p" innerhalb eines `<info>`-Elements
- » Alle Elemente mit der Klasse "info" innerhalb eines Absatzes

### **Antwort 7**

Alle Elemente der Klasse "p" innerhalb eines `<info>`-Elements

### **Frage 8**

Welche Elemente werden durch den CSS-Selektor `p > .info` ermittelt?

- » Alle Elemente mit der Klasse "info", deren direktes Elternelement ein Absatz ist
- » Alle Elemente mit der Klasse "info" irgendwo innerhalb eines Absatzes

### **Antwort 8**

Alle Elemente mit der Klasse "info", deren direktes Elternelement ein Absatz ist

### **Frage 9**

Welche Elemente werden durch den CSS-Selektor `#info` ermittelt?

- » Alle `<info>`-Elemente
- » Alle Elemente mit der Klasse "info"
- » Das Element mit der ID "info"

### **Antwort 9**

Das Element mit der ID "info"

### **Frage 10**

Mit welchen der folgenden Möglichkeiten kann CSS in HTML eingebunden werden?

- » Innerhalb des speziellen HTML-Elements `<css>`
- » Für einzelne HTML-Elemente über das `css`-Attribut
- » Für einzelne HTML-Elemente über das `style`-Attribut
- » Als externe Datei, die mit `<link rel="stylesheet">` im `<head>` eingebunden wird

### **Antwort 10**

- » Als externe Datei, die mit `<link rel="stylesheet">` im `<head>` eingebunden wird

# Lernkontrolle: Responsive Webdesign

## Frage 1

Welche der folgenden Vorgehensweisen entspricht nicht dem Responsive Webdesign?

- » Eine getrennte Webseite für mobile Geräte und eine Webseite für Desktop-Computer erstellen.
- » Getrennte Stylesheets mit Anpassungen für unterschiedliche Bildschirmgrößen erstellen.
- » Bei der Gestaltung einer Webseite zunächst von kleinen Bildschirmen ausgehen und sich dann hoch arbeiten.
- » Eine Seite so zu gestalten, dass sie sowohl auf kleinen als auch auf großen Bildschirmen gut aussieht.

## Antwort 1

Bei der Gestaltung einer Webseite zunächst von kleinen Bildschirmen ausgehen und sich dann hoch arbeiten.

## Frage 2

Welche Techniken kann man zum Erstellen einer responsiven Webseite anwenden?

- » Mit Media Queries Fallunterscheidungen in seine Stylesheets einbauen.
- » Unterschiedliche HTML-Dateien für unterschiedliche Bildschirmgrößen erstellen.
- » Unterschiedliche CSS-Dateien für unterschiedliche Bildschirmgrößen erstellen.
- » Gestaltung der Webseite mit einem CSS-basierten Layout Grid und Media Queries.

## Antwort 2

- » Mit Media Queries Fallunterscheidungen in seine Stylesheets einbauen.
- » Unterschiedliche CSS-Dateien für unterschiedliche Bildschirmgrößen erstellen.
- » Gestaltung der Webseite mit einem CSS-basierten Layout Grid und Media Queries.

## Frage 3

Wie verhält sich ein CSS-basiertes Layout Grid, wenn der Bildschirm eine bestimmte Mindestgröße unterschreitet?

- » Alle Elemente werden untereinander und mit maximaler Breite angezeigt.
- » Alle Elemente werden untereinander und mit minimaler Breite angezeigt.
- » Alle Elemente werden nebeneinander und mit minimaler Höhe angezeigt.

## Antwort 3

Alle Elemente werden untereinander und mit maximaler Breite angezeigt.

## Frage 4

Wie verhält sich ein Media Query, wenn der Bildschirm eine bestimmte Mindestgröße unterschreitet?

- » Die ausgewählten Elemente werden untereinander angezeigt.
- » Die ausgewählten Elemente werden ausgeblendet.
- » Kann man so nicht sagen, kommt ganz auf das Media Query an.

## Antwort 4

Kann man so nicht sagen, kommt ganz auf das Media Query an.

### Frage 5

Vervollständigen Sie den folgenden Quellcode für Bootstrap. Gehen Sie dabei von einem Layout für mittelgroße Bildschirme aus.

```
<div class="____">  
  <div class="____">  
    <div class="____-12">...</div>  
  </div>  
  <div class="row">  
    <div class="____-4">...</div>  
    <div class="____-6">...____  
    <div class="____-md-2">...</div>  
  </div>  
</div>
```

### Antwort 5

Teil 1: container; Teil 2: row; Teil 3: col-md; Teil 4: col-md; Teil 5: col-md; Teil 6: </div>; Teil 7: col

# Lernkontrolle: JavaScript

## Frage 1

Welche Eigenschaften besitzt die Programmiersprache JavaScript?

- » JavaScript läuft traditionell im Browser, in jüngerer Zeit wird es aber auch vermehrt als Allzweckssprache genutzt.
- » Wie von Java gewohnt, müssen JavaScript-Programme vor der Ausführung kompiliert werden.
- » Java und JavaScript haben eine ähnliche Syntax.
- » JavaScript-Programme bestehen aus mindestens einer Klasse und einer main()-Methode.

## Antwort 1

- » JavaScript läuft traditionell im Browser, in jüngerer Zeit wird es aber auch vermehrt als Allzweckssprache genutzt.
- » Java und JavaScript haben eine ähnliche Syntax.

## Frage 2

Welcher Zusammenhang besteht zwischen JavaScript und HTML5?

- » HTML5 ist eine Beschreibungssprache für einfache Textdokumente. Komplexe Webseiten sollten eher mit XHTML erstellt werden.
- » HTML5 und JavaScript sind auf die Entwicklung plattformunabhängiger Browser Apps ausgerichtet.
- » Durch die neuen Funktionen von HTML5 soll JavaScript mittelfristig abgeschafft werden.
- » JavaScript ist ein wichtiger Bestandteil von HTML5, ohne den sich viele Funktionen nicht nutzen lassen.

## Antwort 2

- » HTML5 und JavaScript sind auf die Entwicklung plattformunabhängiger Browser Apps ausgerichtet.
- » JavaScript ist ein wichtiger Bestandteil von HTML5, ohne den sich viele Funktionen nicht nutzen lassen.

## Frage 3

Was bedeutet "dynamische Typisierung" bzw. "Duck Typing"?

- » Der Typ einer Variable ergibt sich allein aus ihrem Inhalt. Durch Zuweisen eines anderen Werts kann sich der Typ einer Variable daher ändern.
- » Eine Methode eines Objekts kann nur aufgerufen werden, wenn sie in seiner Klasse definiert wurde. Die Prüfung findet schon vor Ausführung des Quellcodes statt.
- » Bei der Deklaration einer Variable muss ihr Typ (z.B. String oder int) angegeben werden.
- » Die Attribute und Methoden eines Objekts können jederzeit verändert werden. Ob eine Methode überhaupt aufgerufen werden kann, kann daher erst zur Laufzeit festgestellt werden.

## Antwort 3

- » Der Typ einer Variable ergibt sich allein aus ihrem Inhalt. Durch Zuweisen eines anderen Werts kann sich der Typ einer Variable daher ändern.

## Frage 4

Mit welcher Syntax wird in JavaScript eine lokale Variable definiert?

- » buch = "Per Anhalter durch die Galaxis"

```
» window.buch = "Das Restaurant am Ende des Universums"
» let buch = "Das Leben, das Universum und der ganze Rest"
» var buch = "Macht's gut und vielen Dank für all den Fisch"
» set buch = "Einmal Rupert und zurück"
```

#### Antwort 4

```
» let buch = "Das Leben, das Universum und der ganze Rest"
» var buch = "Macht's gut und vielen Dank für all den Fisch"
```

#### Frage 5

Mit welcher Syntax wird in JavaScript eine Fallunterscheidung programmiert?

```
» if (ergebnis == 42) { ... }
» if (ergebnis = 42) { ... }
» IF ergebnis = 42 THEN ... ENDIF
» switch (ergebnis) { case 42: ... }
```

#### Antwort 5

```
» if (ergebnis == 42) { ... }
» switch (ergebnis) { case 42: ... }
```

#### Frage 6

Mit welcher Syntax wird in JavaScript eine Funktion definiert?

```
» addiere = (a,b) => { return a + b
» }
» addiere(a, b) { return a +b }
» addiere = (a,b) => a + b
» function addiere(a, b) { return a + b
» }
» addiere = a + b
```

#### Antwort 6

```
» addiere = (a,b) => { return a + b
» }
» addiere = (a,b) => a + b
» function addiere(a, b) { return a + b
» }
```

#### Frage 7

Welche Eigenschaften haben Listen in JavaScript?

```
» Anders als in Java können die Einträge einer Liste unterschiedliche Datentypen haben.
» Die Einträge einer Liste werden anhand eines Index angesprochen. Der erste Eintrag hat dabei immer den Index 0.
» Die Einträge einer Liste werden über einen frei wählbaren String als Schlüsselwert angesprochen.
```

#### Antwort 7

```
» Anders als in Java können die Einträge einer Liste unterschiedliche Datentypen haben.
```

» Die Einträge einer Liste werden anhand eines Index angesprochen. Der erste Eintrag hat dabei immer den Index 0.

### Frage 8

Welche Eigenschaften haben Dictionaries in JavaScript?

- » Die Einträge eines assoziativen Arrays werden anhand eines Index angesprochen. Der erste Eintrag hat dabei immer den Index 0.
- » Listen und Dictionaries können beliebig geschachtelt werden, um komplexe Datenstrukturen abzubilden.
- » Die Einträge eines assoziativen Arrays können unterschiedliche Datentypen haben. Dies entspricht nach am ehesten einer Map<String, Object> in Java.
- » Die Einträge eines assoziativen Arrays werden über einen frei wählbaren String als Schlüsselwert angesprochen.

### Antwort 8

- » Die Einträge eines assoziativen Arrays werden anhand eines Index angesprochen. Der erste Eintrag hat dabei immer den Index 0.
- » Listen und Dictionaries können beliebig geschachtelt werden, um komplexe Datenstrukturen abzubilden.
- » Die Einträge eines assoziativen Arrays können unterschiedliche Datentypen haben. Dies entspricht nach am ehesten einer Map<String, Object> in Java.

### Frage 9

Welche Schreibweise definiert eine Liste und welche ein Dictionary?

```
{var entries = {"id": 4612, "vorlesung": "Webprogrammierung"};; var entries = {"places": ["Karlsruhe", "Ettlingen", "Malsch"]};; var entries = [[1, 2], [3, 4], 5];} -> {Dictionary; Liste; Sowohl als auch}
```

### Antwort 9

- » var entries = {"id": 4612, "vorlesung": "Webprogrammierung"}
- » -> Liste
- » var entries = {"places": ["Karlsruhe", "Ettlingen", "Malsch"]}
- » -> Sowohl als auch
- » var entries = [[1, 2], [3, 4], 5]
- » -> Liste

### Frage 10

Mit welcher Anweisung kann man an einer beliebigen Stelle den Debugger starten?: debug; debugger; startdebug; breakpoint; halt

### Antwort 10

- » debugger

### Frage 11

Mit welchen Methoden wird das HTML-Element mit der ID „wichtig“ ermittelt?:

document.getElementsByClassName("wichtig");	document.getElementById("wichtig");
document.createTextNode("wichtig");	document.querySelector("#wichtig");
document.getElementsByName("wichtig")	

### Antwort 11

- » document.getElementById("wichtig")
- » document.querySelector("#wichtig")

### Frage 12

Ordne den folgenden DOM-Methoden ihre Aufgabe zu: {element.removeChild(e1); element.appendChild(e1); document.createElement("div"); document.createTextNode("Hallo, Welt"); document.querySelectorAll(".info")} -> {Entfernt ein Element aus dem DOM-Baum; Erzeugt ein neues HTML-Element; Fügt ein Element einem anderen als Unterelement hinzu; Erzeugt ein neues Text-Element; Sucht alle Elemente mit der HTML-Klasse info}

### Antwort 12

- » element.removeChild(e1) -> Entfernt ein Element aus dem DOM-Baum
- » element.appendChild(e1) -> Fügt ein Element einem anderen als Unterelement hinzu
- » document.createElement("div") -> Erzeugt ein neues HTML-Element
- » document.createTextNode("Hallo, Welt") -> Erzeugt ein neues Text-Element
- » document.querySelectorAll(".info") -> Sucht alle Elemente mit der HTML-Klasse info

### Frage 13

Welche DOM-Methoden erfüllen die folgenden Aufgaben? {Registrierung eines Ereignisbehandlers; Suche eines Elements anhand seiner ID; Suche des ersten Elements zu einem CSS-Selektor; Suche aller Elemente einer bestimmten Klasse} -> {document.getElementById("..."); document.getElementsByClassName("..."); element.addEventListener("event", func); document.querySelector("...")}

### Antwort 13

- » Registrierung eines Ereignisbehandlers -> element.addEventListener("event", func)
- » Suche eines Elements anhand seiner ID -> document.getElementById("...")
- » Suche des ersten Elements zu einem CSS-Selektor -> document.querySelector("...")
- » Suche aller Elemente einer bestimmten Klasse -> document.getElementsByClassName("...")

### Frage 14

Vervollständige die folgenden DOM-Manipulationen:

```
// CSS-Eigenschaft border-width überschreiben  
element.____.borderWidth = "2px";
```

```
// CSS-Eigenschaft margin-bottom überschreiben  
element.____ = "1em";
```

```
// HTML-Klassen hinzufügen und entfernen  
element.____.add("selected");  
element.____.remove("invisible");
```

```
// HTML-Inhalt eines Elements überschreiben  
element.____ = "<b>Achtung, Sonderangebot!</b> Sparen Sie jetzt 5 Cent!";
```

```
// Textinhalt eines Elements überschreiben  
element.____ = "Vielen Dank für Ihren Einkauf.";
```

**Antwort 14**

Teil 1: style; Teil 2: style.marginBottom; Teil 3: classList; Teil 4: classList; Teil 5: innerHTML; Teil 6: textContent



# Lernkontrolle: Serverseitige Webentwicklung

## Frage 1

HTTP folgt einem Anfrage/Antwort-Mechanismus. Auf eine Anfrage des Servers schickt der Browser eine Antwort.

## Antwort 1

Falsch

## Frage 2

Bei einer GET-Anfrage können keine Daten im HTTP Request Body mitgeschickt werden.

## Antwort 2

Wahr

## Frage 3

Bei einer PUT-Anfrage können keine Daten im HTTP Response Body mitgeschickt werden.

## Antwort 3

Falsch

## Frage 4

Ordne den folgenden Statuscodes ihre jeweilige Bedeutung zu.

{200 Ok; 202 Accepted; 301 Moved Permanently; 404 Not Found; 501 Not Implemented} -> {Die gesuchte Information wurde nicht gefunden; Die gewünschte Funktion ist nicht verfügbar; Die gesuchte Information hat eine neue Adresse; Anfrage war erfolgreich; Anfrage wurde empfangen, wird aber später verarbeitet}

## Antwort 4

- » 200 Ok -> Anfrage war erfolgreich
- » 202 Accepted -> Die gewünschte Funktion ist nicht verfügbar
- » 301 Moved Permanently -> Die gesuchte Information hat eine neue Adresse
- » 404 Not Found -> Die gesuchte Information wurde nicht gefunden
- » 501 Not Implemented -> Die gewünschte Funktion ist nicht verfügbar

## Frage 5

Welche der folgenden Aussagen über HTTP/2 ist wahr?

- » Semantisch ist es eine Erweiterung zu HTTP/1, jedoch werden die Nachrichten nun binär kodiert.
- » Semantisch ist es eine komplette Neuentwicklung, die Nachrichten werden aber weiterhin im Textformat gesendet.
- » Aufgrund der neuen Art, wie die Nachrichten kodiert werden, ist HTTP/2 effizienter als HTTP/1.
- » HTTP/2 sieht vor, dass der Server Dateien an den Browser schicken kann, die der Browser sehr wahrscheinlich benötigen wird.

## Antwort 5

- » Aufgrund der neuen Art, wie die Nachrichten kodiert werden, ist HTTP/2 effizienter als HTTP/1.

### Frage 6

Welchen Weg nimmt eine HTTP-Anfrage typischerweise auf dem Server? Schiebe die Begriffe an die richtige Stelle in der Zeichnung.; [[Dropzone 1]] -> {1. URL-Routing / 2. Request Handler / 3. HTML-Template / 4. HTTP-Anfrage / 5. HTTP-Antwort}; [[Dropzone 2]] -> {1. URL-Routing / 2. Request Handler / 3. HTML-Template / 4. HTTP-Anfrage / 5. HTTP-Antwort}; [[Dropzone 3]] -> {1. URL-Routing / 2. Request Handler / 3. HTML-Template / 4. HTTP-Anfrage / 5. HTTP-Antwort}; [[Dropzone 4]] -> {1. URL-Routing / 2. Request Handler / 3. HTML-Template / 4. HTTP-Anfrage / 5. HTTP-Antwort}; [[Dropzone 5]] -> {1. URL-Routing / 2. Request Handler / 3. HTML-Template / 4. HTTP-Anfrage / 5. HTTP-Antwort}

### Antwort 6

Dropzone 1 -> {1. URL-Routing} Dropzone 2 -> {2. Request Handler} Dropzone 3 -> {3. HTML-Template} Dropzone 4 -> {4. HTTP-Anfrage} Dropzone 5 -> {5. HTTP-Antwort}

### Frage 7

Welche Aufgabe hat der URL-Router einer Webanwendung?: Er entscheidet anhand der Webadresse der Anfrage, an welchen Webserver sie gesendet wird.; Er entscheidet anhand der Webadresse der Anfrage, welches HTML-Template aufgerufen wird.; Er entscheidet anhand des Statuscodes der Antwort, auf welche Folgeseite der Anwender weitergeleitet wird.; Er entscheidet anhand der Webadresse der Anfrage, welcher Request Handler aufgerufen wird.

### Antwort 7

» Er entscheidet anhand der Webadresse der Anfrage, welcher Request Handler aufgerufen wird.

### Frage 8

Welche Aufgabe haben die Request Handler einer Webanwendung?: Sie führen die serverseitige Anwendungslogik zur Bearbeitung einer Anfrage aus.; Sie reagieren auf verschiedene Ereignisse, die im JavaScript-Code einer Webseite ausgelöst werden.; Sie werten die Kopfdaten aller HTTP-Anfragen aus, um sie an den richtigen URL-Router weiterzuleiten.; Sie protokollieren alle empfangenen HTTP-Anfragen, um die Fehlersuche zu vereinfachen.

### Antwort 8

» Sie führen die serverseitige Anwendungslogik zur Bearbeitung einer Anfrage aus.

### Frage 9

Welche Aufgabe haben die HTML-Templates einer Webanwendung?: Sie entscheiden, welcher Request Handler zur Verarbeitung einer Anfrage aufgerufen wird.; Es handelt sich um Kopiervorlagen, die zur Erstellung eigener Webseiten angepasst werden können.; Sie erzeugen den finalen HTML-Code, der als Antwort an den Browser gesendet wird.; Sie beinhalten das Document Object Model, mit dem die HTTP-Anfragen verarbeitet werden.

### Antwort 9

» Es handelt sich um Kopiervorlagen, die zur Erstellung eigener Webseiten angepasst werden können.

### Frage 10

Was sind Servlets?

- » Selbsterstellte Javaklassen, die zwar keinen ganzen Webserver beinhalten, jedoch von einem aufgerufen werden können, um HTTP-Anfragen zu bearbeiten.
- » Von Java vordefinierte Klassen, die einen eigenen Webserver implementieren und daher in eigenen

Programmen aufgerufen werden können.

» Selbsterstellte Javaklassen mit einer `public static void main()`-Methode, die einen eigenen Webserver implementieren.

#### **Antwort 10**

» Von Java vordefinierte Klassen, die einen eigenen Webserver implementieren und daher in eigenen Programmen aufgerufen werden können.

#### **Frage 11**

Welche Unterschiede und Gemeinsamkeiten haben HTML- und JSP-Dateien auf einem Webserver?

» So oder so versteht der Server die HTML-Anweisungen nicht. Wohl aber versteht er die Java-Anweisungen in einer JSP.

» In beiden Fällen steht in der URL, welche Datei vom Server abgerufen werden soll.

» JSP-Dateien beinhalten nur HTML-Code, eigentlich ist es daher dasselbe.

» Bei einer HTML-Datei werden nur die HTML- und JavaScript-Anweisungen vom Server ausgeführt, bei JSP zusätzlich noch Java.

#### **Antwort 11**

» Bei einer HTML-Datei werden nur die HTML- und JavaScript-Anweisungen vom Server ausgeführt, bei JSP zusätzlich noch Java.

#### **Frage 12**

Wie ist die `doGet()`-Methode eines HTTP-Servlets definiert (ohne Exceptions)?

» `public HttpServletResponse doGet(HttpServletRequest request)`

» `public ServletResponse doGet(ServletRequest request)`

» `public void doGet(ServletRequest request, ServletResponse response)`

» `public void doGet(HttpServletRequest request, HttpServletResponse response)`

#### **Antwort 12**

» `public void doGet(HttpServletRequest request, HttpServletResponse response)`

#### **Frage 13**

Welche Aufgabe hat der Request Context eines Servlets?

» Zwischenspeichern von Werten, um sie mit anderen Benutzern zu teilen.

» Zwischenspeichern von Werten zwischen zwei HTTP-Anfragen.

» Zwischenspeichern von Werten, um sie an ein anderes Servlet oder eine JSP weiterzugeben.

#### **Antwort 13**

» Zwischenspeichern von Werten zwischen zwei HTTP-Anfragen.

#### **Frage 14**

Welche Aufgabe hat der Session Kontext eines Servlets?

» Zwischenspeichern von Werten, um sie an ein anderes Servlet oder eine JSP weiterzugeben.

» Zwischenspeichern von Werten zwischen zwei HTTP-Anfragen.

» Zwischenspeichern von Werten, um sie mit anderen Benutzern zu teilen.

#### **Antwort 14**

» Zwischenspeichern von Werten, um sie mit anderen Benutzern zu teilen.

#### **Frage 15**

Mit welcher Variante ist der Quellcode einer Webanwendung am übersichtlichsten?

- » Am besten nur JSP, HTML und CSS verwenden. Servlets sollte man vermeiden.
- » Am besten nur Servlets, HTML und CSS verwenden. JSP sollte man vermeiden.
- » Wenn eine einfache JSP nicht ausreicht, die Anfrage erst von einem Servlet und dann von einer JSP bearbeiten lassen.
- » Wenn ein einfaches Servlet nicht ausreicht, die Anfrage erst von einer JSP und dann von einem Servlet bearbeiten lassen.

#### **Antwort 15**

» Wenn eine einfache JSP nicht ausreicht, die Anfrage erst von einem Servlet und dann von einer JSP bearbeiten lassen.

#### **Frage 16**

Mit welcher Syntax wird in JSP der Wert einer Berechnung an den Browser geschickt?

- » `<%@page a+b %>`
- » `<%-- a + b --%>`
- » `<%= a + b %>`
- » `<% a + b %>`

#### **Antwort 16**

» `<% a + b %>`

#### **Frage 17**

Warum sollte eine JSP so wenig wie möglich Javacode beinhalten (idealerweise keinen)?

- » Weil der Quellcode der Seite immer unübersichtlicher wird, je mehr Javacode sie enthält.
- » Weil der Browser nur JavaScript aber kein Java ausführen kann.
- » Weil sich im Extremfall dann HTML, CSS, JavaScript und Java in einer einzigen Daten mischen.
- » Weil die Gefahr besteht, dass der Browser den Javacode als JavaScript auszuführen versucht.

#### **Antwort 17**

» Weil der Quellcode der Seite immer unübersichtlicher wird, je mehr Javacode sie enthält.

#### **Frage 18**

Was bewirkt der folgende JSP-Quellcode?

```
<h1>Willkommen, ${username}!</h1>
<c:forEach var="message" items="${messages}">
    ${message} <br/>
</c:forEach>
```

- » Es werden die Inhalte zweier Variablen angezeigt.

- » Es werden der Wert einer Variable und der Inhalt einer Liste angezeigt.
- » Es wird eine Meldung in der bevorzugten Sprache des Anwenders angezeigt.

**Antwort 18**

Es werden der Wert einer Variable und der Inhalt einer Liste angezeigt.

**Frage 19**

Welche Vorteile bietet das in den Folien gezeigte Verfahren der Templatevererbung?

- » Man kann damit eine Vorlage für das HTML-Grundgerüst mehrerer Seiten erstellen und deren Platzhalter dann mit Inhalten füllen.
- » Man kann das Verfahren nutzen, um PDF, MP3 oder JPG-Dateien auf dem Server zu generieren.
- » Man kann das Layout mehrerer Seiten noch mehr vereinheitlichen, als es mit CSS alleine möglich wäre.
- » Man kann damit sicherstellen, dass auf allen Seiten einer Webanwendung immer bestimmte Inhalte erscheinen.
- » Man kann jeden beliebigen Webserver für diese Technik verwenden.

**Antwort 19**

- » Man kann damit eine Vorlage für das HTML-Grundgerüst mehrerer Seiten erstellen und deren Platzhalter dann mit Inhalten füllen.

# Onlinetest zum Projekt

## Frage 1

In welcher Projektgruppe bist du?

## Antwort 1

WWI17B4 Gruppe MyVitality

## Frage 2

Für welche Art von Webanwendung habt ihr euch entschieden? (1P)

- \* Browserseitige Single Page App ohne Server-Backend
- \* Browserseitige Single Page App mit Server-Backend
- \* Browserseitige App (keine Single Page App) ohne Server-Backend
- \* Browserseitige App (keine Single Page App) mit Server-Backend
- \* Serverseitige Webanwendung

HINWEIS: Bei einer Single Page App wird die Anwendungslogik überwiegend mit JavaScript im Browser realisiert. Bei einer serverseitigen Webanwendung liegt die Anwendungslogik auf dem Server, wobei in der Regel nach jeder Aktion eine neue HTML-Seite generiert wird.

## Antwort 2

Der überwiegende Teil der Anwendung läuft im Backend, jedoch ist auch viel Logik in Form von JS Dateien, welche im Browser laufen implementiert.

Aus diesen Gründen würde ich unsere Anwendung als Browserseitige App mit großem Server Backend bezeichnen.

## Frage 3

Warum habt ihr euch für diese Art von Webanwendung entschieden? Nenne drei Vorteile im Vergleich zu den anderen beiden Arten. (3P)

## Antwort 3

1) Da wir auf jeden Fall ein Serverbacked mit Persistenzschicht wollten kam Nr.1 und Nr. 3 nicht in Frage. ->Vorteil: wir haben eine Persistenzschicht

2) Bei einem Http Request soll nicht die gesamte Anwendung geladen werden, dh. kommt Nr.2 nicht in Frage. -> Vorteil: wir geben nur den Benötigten Content aus

3) Da wir nicht bei jedem Http Request eine neue HTML Seite generieren lassen wollen, kommt Nr. 5 nicht in Frage. -> Vorteil: Wir entlasten den Server, da in Zeiten leistungsstarker Clients, diese auch Arbeit übernehmen können.

## Frage 4

Angenommen, ihr hättet euch für eine der anderen beiden Arten entschieden. Was hättet ihr dann anders machen müssen? Nenne zwei Auswirkungen und begründe sie. (4P)

## Antwort 4

Hätten wir uns für eine Browserseitige single Page App entschieden, so müssten wir beim Aufruf der

Anwendung via Http Request die gesamte Anwendung senden, sodass diese im Browser geladen werden kann.

Hätten wir uns für eine rein Serverseitige Anwendung entschieden, so müsste jedesmal wenn ein Http Request gesendet wird eine neue Seite Serverseitig generiert werden und diese an den Browser gesendet werden.

### **Frage 5**

Wie funktioniert die Navigation in eurer Anwendung? Erkläre anhand des Quellcodes was Schritt für Schritt passiert, wenn der Anwender von der Startseite auf eine andere Seite navigiert.

Falls die Navigation nur aus einem Link von einer HTML-Datei auf eine andere HTML-Datei besteht, erkläre stattdessen, wie ihr einen einheitlichen Aufbau und ein einheitliches Layout sicherstellt oder sicherstellen könntet. (10P)

### **Antwort 5**

Allgemeine Beschreibung:

Beim Aufruf unserer Anwendung via Http GET empfängt der ServletContorller den Request und Mappt diesen auf eine entsprechende JSP.

Von der Startseite aus:

Hat sich ein User Angemeldet, so gelangt er in den für ihn entsprechenden HomeBereich. Hier werden im die einzelnen Bereiche unserer Anwendung in Form einer HTML Card visualisiert. Diese Card ist in einem Ankerelement eingebunden, welche einen entsprechenden Link enthält.

Einheitliches Layout:

Für ein einheitliches Layout verwenden wir ein Template, welches wir selbst erstellt haben. Dieses Template befindet sich im package templates und nennt sich base.tag.

In diesem sind alle Layout und head-Elemente enthalten, welche auf jeder Seite benötigt werden. Dazu gehören unter Anderem: Header, Footer, die Einbindung von Bootstrap und unsere allgemeinen Stylesheet (css).

Innerhalb des Templates sind Bereiche für den Head, Header, Main und Footer bereitgestellt, die in anderen JSP aufgegriffen und verändert werden können. So kann man überall das selbe Layout verwenden und trotzdem die einzelnen Bereiche individuelle gestalten.

### **Frage 6**

Suche dir 10 zusammenhängende Codezeilen in einem von dir selbst geschriebenen Stylesheet aus und erkläre, was diese bewirken. (10P)

### **Antwort 6**

```
#aktualisieren{
    background-color: green;
}
table {
    overflow-x: auto;
```

```
font-family: arial, sans-serif;
border-collapse: collapse;
width: 100%;
}

td, th {
border: 1px solid #dddddd;
text-align: left;
padding: 8px;
}

tr:nth-child(even) {
background-color: darkseagreen;
}

.h1{
font-size: 4vw;
}
```

- 1) Der Button aktualisieren soll die Farbe grün haben.
- 2) Eigenschaft gibt an, ob der Content beschnitten, eine Bildlaufleiste hinzugefügt oder der Überlaufinhalt eines Elements auf Blockebene angezeigt werden soll, wenn es links oder rechts überläuft.
- 3) Schriftart innerhalb der Tabelle soll arial,sans serif sein
- 4) Border-collapse legt fest, ob Tabellenrahmen zu einem einzelnen Rand zusammenfallen oder getrennt werden soll.
- 5)Die Breite soll 100% betragen
- 6) Die Tabellenspalten und Überschriften sollen einen Rahmen von 1px stärke haben, solid sein und die Farbe #dddd haben.
- 7) Textausrichtung ist links
- 8) Der Abstand des Contents zum Rahmen beträgt 8 px
- 9) Jedes zweite Zeile soll die Fare darkseagreen haben
- 10) Die Schriftgröße der h1 Überschrift beträgt 4vw;

### **Frage 7**

Such dir eine von dir selbst geschriebene Java/JavaScript/...-Klasse (oder Codedatei) aus und erkläre, was ihre Aufgabe im Vergleich zu den anderen Klassen (oder Codedateien) ist und für jede Methode was sie bewirkt und warum sie in dieser Klasse (oder Codedatei) enthalten ist. (10P)

### **Antwort 7**

```
ackage de.dhbw.myvitality.services;

import de.dhbw.myvitality.entities.Article;
```



```
import de.dhbw.myvitality.entities.SupplementConfiguration;
import de.dhbw.myvitality.repositories.SupplementConfigurationRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import java.util.ArrayList;

/**
 * Anwendungslogik für die individuelle Nahrungsergänzungsmittelkonfiguration inkl. Datenbankzugriff per Repository
 * @author Tamino Fischer alias CodeKeks
 */
@Service
public class SupplementConfigurationService {

    @Autowired
    private SupplementConfigurationRepository supplementConfigurationRepository;

    //Find All
    public ArrayList<SupplementConfiguration> findAll(){
        ArrayList<SupplementConfiguration> list = new ArrayList<SupplementConfiguration>();
        supplementConfigurationRepository.findAll().forEach(list::add);
        return list;
    }

    //Find Article List By a customerId
    public ArrayList<Article> findArticleListByCustomerId(String customerId){
        ArrayList<Article> list = new ArrayList<Article>();
        for (Article l: supplementConfigurationRepository.findByQuery(customerId).get().getArticleList()) {
            list.add(l);
        }
        return list;
    }
}
```

Die gewählte Klasse lautet SupplementConfigurationService.

Allgemeine Erklärung:

Wir ein Http Request von dem jeweiligen Controllen entgegengenommen, so soll die benötigte Serviceklasse aufgerufen werden. Daraus ergibt sich folgendes:

Der Controller ist leiglich für ein Mapping der einzelnen Requests zuständig. Die eigentliche Anwendungslogik, der Anwendungskern steckt inden Serviceklassen. Diese regeln somit auch den Datenbankzugriff (dazu in der speziellen Erklärung mehr). Daten die verarbeitet werden, stecken in Instanzen der Entity klassen und bilden somit die Querschnittsfunktionalität vertikal zur Schichtenarchitektur.

Spezielle Erklärung:

Der `SupplementConfigurationService` befindet sich im package `Services`, wodurch unserer monolithischen Anwendung eine grundlegende Struktur gegeben wird.

Es handelt sich um eine Java-Klasse, welche mit der Annotation `@Service` versehen ist. Durch diese Annotation wird per Context and Dependency Injection der Lifecycle von daraus erzeugten Instanzen über unser Framework (spring) geregelt. Das bedeutet wir instanzieren nicht selbst Objekte dieser Klasse sondern lassen sie uns erzeugen, wenn wir sie brauchen (Thread-Sicherheit).

Als Feld der Klasse haben wir ein `supplementConfigurationRepository`, dessen Lifecycle ebenfalls per CDI gemanagt wird. Dieses erbt vom Interface `CrudRepository`, welches den Datenbankzugriff regelt.

Die Methode `findAll` liest aus der Datenbank alle `SupplementConfiguration`-Instanzen aus und gibt diese als Liste zurück.

Die Methode `findArticleListByCustomerId` liest aus der Datenbank alle Artikel, welche zu einem Kunden gehören aus und gibt diese als Liste zurück.