

Klausur Web-Applikationen WS 2022/2023

Name:

Matrikelnummer:

| Aufgabe | -1- | -2- | -3- | -4- | -5- | -6- | -7- |
|----------|-----|-----|-----|-----|-----|-----|-----|
| Punkte | 23 | 11 | 8 | 23 | 4 | 5 | 16 |
| erreicht | | | | | | | |

Summe (max. 90 Punkte):

Note:

Zeit: 90 Minuten

Erlaubte Hilfsmittel: ein DIN-A4-Blatt mit eigenen Notizen (beidseitig)

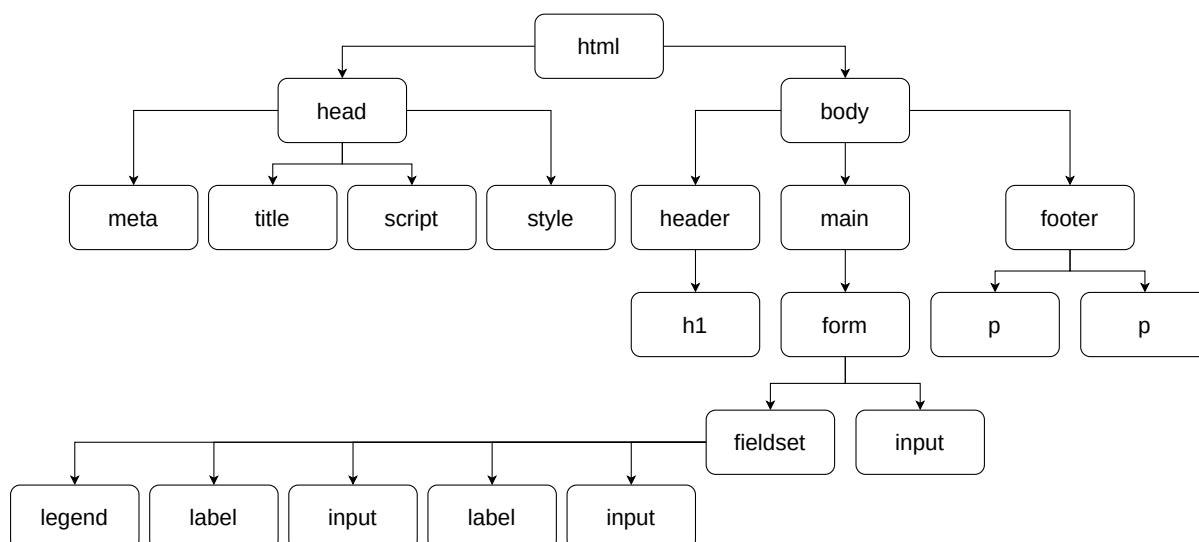
Viel Erfolg!

Aufgabe 1 (HTML & JavaScript) (23 Punkte)

Skizzieren Sie den zum HTML zugehörigen DOM-Tree (ohne Text-Knoten). Denken Sie daran, dass die Reihenfolge der Knoten eine Rolle spielt. **(5 Punkte)**

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>Wordle | Login</title>
    <script defer src="main.js"></script>
    <style>#secret { visibility: hidden; }</style>
  </head>
  <body>
    <header><h1>You need to login first</h1></header>
    <main>
      <form id="loginForm" action="https://wordle.de">
        <fieldset>
          <legend>Login</legend>
          <label for="username">Username</label>
          <input type="text" name="username" id="user" />
          <label for="password">Password</label>
          <input type="password" name="password" id="key" />
        </fieldset>
        <input type="submit" id="login" value="Login" />
      </form>
    </main>
    <footer>
      <p id="license">licensed under MIT</p>
      <p id="secret">Your password is 123456</p>
    </footer>
  </body>
</html>
```

DOM-Tree:



Sie wollen am Ende des Formulars, d.h. direkt vor `</form>`, dynamisch via DOM-Manipulation (JavaScript) einen Reset-Button (`<input type="reset"/>`) hinzufügen. Auf dem Button soll der Text "Clear" stehen. Was schreiben Sie dazu in die Datei `main.js`?

Tipp: Nutzen Sie `createElement()`, `appendChild()`, sowie die Attribute `type` und `value`. **(6 Punkte)**

`main.js`:

```
const form = document.getElementById("loginForm");
const reset = document.createElement("input");
reset.type = "reset";
reset.value = "Clear";
form.appendChild(reset);
```

Das `<script>`-Tag hat das Attribut `defer`. Kreuzen Sie alle korrekten Aussagen an. **(2 Punkte)**

- ☒ Das Skript wird asynchron geladen.
- ☐ Das Skript wird nur auf dem Server ausgeführt.
- ☒ Das Skript wird erst ausgeführt, wenn das HTML-Dokument vollständig geparkt wurde.

Unten stehend finden Sie einen anderen Code-Ausschnitt aus `main.js`. Was geschieht bei Aufruf dieses Codes, wenn `defer` auf `false` gesetzt wird (bzw. fehlt)? **(2 Punkte)**

```
// main.js
alert(document.getElementById("license").firstChild.nodeValue);
```

Das HTML-Element mit der ID "license" steht noch nicht zur Verfügung. Somit wertet `document.getElementById("license")` zu `null` aus. Dies erzeugt einen Fehler, sodass keine Alert-Box erscheint.

In das Formular wird als Username `Maria` und als Passwort `123456` eingegeben. Welche URL (inklusive Query-String) wird beim Klicken auf `Login` aufgerufen? **(2 Punkte)**

`https://wordle.de/?username=Maria&password=123456`

Sie wollen nachfolgende Tabelle in ein HTML-Dokument übertragen. Schreiben Sie das hierfür notwendige HTML innerhalb des `<table>`-Tags. Verwenden Sie dabei auch die HTML-Elemente `thead` und `tbody`. Styling und Umrahmung können Sie außer Acht lassen. **(6 Punkte)**

| Nachricht | Art | Statuscode |
|-----------------------|--------------|------------|
| OK | Success | 200 |
| Unauthorized | Client Error | 401 |
| Internal Server Error | Server Error | 500 |

`<table>`

`<thead>`

`<tr>`

`<th>Nachricht</th>`

`<th>Art</th>`

`<th>Statuscode</th>`

`</tr>`

`</thead>`

`<tbody>`

`<tr>`

`<td>OK</td>`

`<td>Success</td>`

`<td>200</td>`

`</tr>`

`<tr>`

`<td>Unauthorized</td>`

`<td>Client Error</td>`

`<td>401</td>`

`</tr>`

`<tr>`

`<td>Internal Server Error</td>`

`<td>Server Error</td>`

`<td>500</td>`

`</tr>`

`</tbody>`

`</table>`

Aufgabe 2 (HTTP & WWW) (11 Punkte)

GET ist ein Beispiel für eine HTTP-Anfragemethode (HTTP-Verb). Nennen Sie **zwei** weitere und beschreiben Sie diese in je einem Satz. **(4 Punkte)**

- **POST** - hängt Daten an den Header an, um eine Ressource auf dem Server zu erstellen oder zu verändern.
- **DELETE** - veranlasst das Löschen einer Ressource auf dem Server.

Ein sogenannter URL-Shortener kürzt URLs. Beispiel: bei Eingabe der URL <https://3.ly/amXxi> erfolgt eine Weiterleitung auf die URL https://de.wikipedia.org/wiki/Breaking_Bad#Episodenliste.

1. Nennen Sie einen Grund, weshalb jemand eine URL kürzen möchte. **(1 Punkt)**

- um Platz zu sparen, z.B. beim Ausdrucken der URL oder im Einsatz bei Kurznachrichtendiensten
- um sie einprägsamer zu machen
- um Klicks auf dem weiterleitenden Server tracken zu können
- um später die URL, auf die verwiesen wird, noch ändern zu können

2. Welcher HTTP-Statuscode (außer 200) kommt im Beispiel wahrscheinlich zum Einsatz? Begründen Sie Ihre Antwort in einem Satz. **(2 Punkte)**

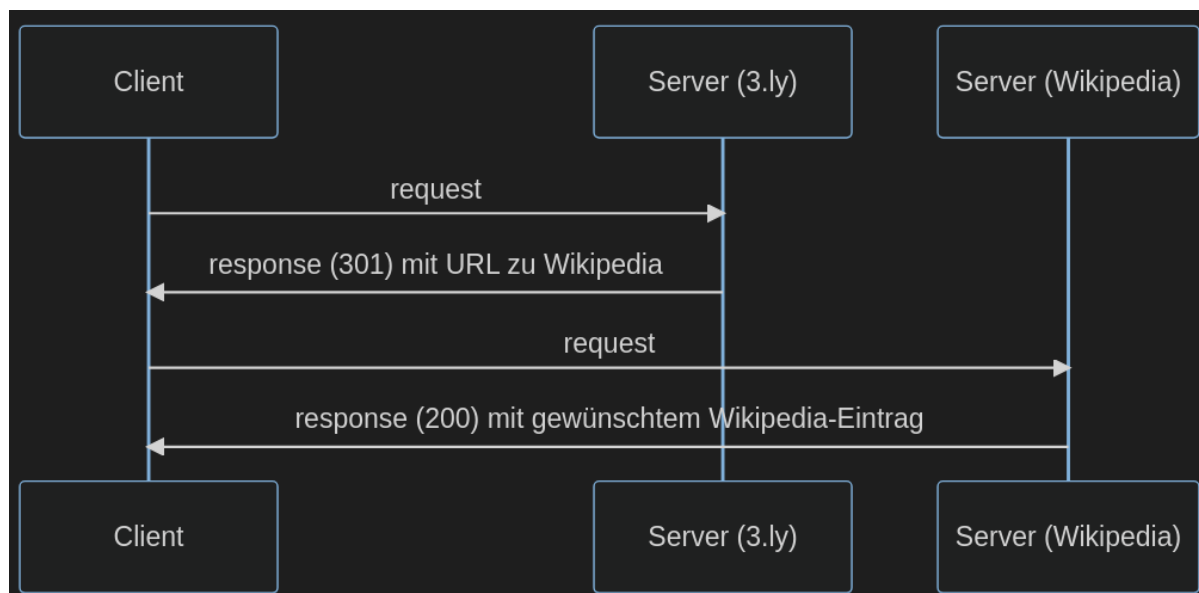
gewünschte Antwort:

- **301 (Moved Permanently)** - da die angefragte Ressource permanent bei Wikipedia zu finden ist

akzeptierte Antworten (wenn korrekt begründet):

- **3xx (Redirect)** - da man weitergeleitet wird
- **401 (Unauthorized)** - falls eine Authentifizierung notwendig ist und diese nicht erfolgte
- **404 (Not Found)** - die angefragte Ressource ist auf einem der Server nicht vorhanden
- **500 (Internal Server Error)** - der Server stößt bei der Bearbeitung auf ein nicht behandelbares Problem

3. Skizzieren Sie ein Diagramm, dass die wesentlichen HTTP-Verbindungen ab Eingabe der kurzen URL bis zur Ansicht der Wikipedia-Seite in einem Client-Server-Modell veranschaulicht. Beachten Sie, dass hierbei mind. 2 Server involviert sind. **(4 Punkte)**



Aufgabe 3 (JSON) (8 Punkte)

Geben Sie ein Beispiel für einen Unterschied zwischen der JSON Syntax und der JavaScript Objekt Syntax. **(2 Punkte)**

- in JSON müssen Schlüssel (Keys) Strings sein: `"name": "Hochschule"`, in JS: `name: "Hochschule"`
- in JS sind einfache Anführungszeichen für Strings möglich: `'Hochschule'`
- in JS kann ein Wert auch `undefined` oder eine Funktion sein

Das folgende JSON ist nicht valide. Markieren Sie die vier fehlerhaften Stellen und beschreiben Sie stichwortartig, welche Änderung jeweils getätigt werden muss. **(6 Punkte)**

- Klammern `()` von `employees` durch `[]` ersetzen, da Array
- `undefined` ist kein gültiger Wert
- Schlüssel `hasLibrary` muss ein String sein -> `"hasLibrary"`
- Komma nach dem letzten Eintrag (`students`) entfernen

```
{
  "name": "Hochschule Fulda",
  "street": "Leipziger Straße",
  "houseNumber": "123",
  "postalCode": 36037,
  "city": "Fulda",
  "employees": (
    {
      "firstName": "D.",
      "lastName": "Biezä",
      "number": 3050,
      "areas": ["AI-Support", "Betreuung Software-Labore", undefined]
    },
    {
      "firstName": "C.",
      "lastName": "Pape",
      "number": 379,
      "areas": ["Betreuung WI-Labor", "Virtualisierung", null]
    }
  ),
  hasLibrary: true,
  "students": 9300,
}
```

Aufgabe 4 (CSS) (23 Punkte)

Nennen und beschreiben Sie **zwei** CSS Eigenschaften (CSS properties). **(3 Punkte)**

- `background-color`: setzt die Hintergrundfarbe eines Elements, z.B. durch einen Hexcode `#f4aa12`
- `font-size`: legt die Schriftgröße fest, z.B. durch einen `em`-Wert oder ein Schlüsselwort wie `large`

Nennen Sie **eine** CSS-Eigenschaft, die an ein Kindelement vererbt wird. **(1 Punkt)**

`color`, `cursor`, `font`, `font-family`,...

Ein über `<link>` verlinktes externes Stylesheet ist meist die beste Wahl, um CSS einzubinden. Beschreiben Sie zwei weitere Möglichkeiten. **(4 Punkte)**

- direkt im HTML-Element (inline) über das Style Attribut, z.B. `<p style="color: blue;">`
- mit dem `<style>`-Tag direkt im `<head>` des HTML-Dokuments, z.B. `<style> p {color: blue;}</style>`

Nennen Sie jeweils ein Block-Element und ein Inline-Element (Hinweis: relevant für die CSS-Eigenschaft `display`). Was ist ein wesentlicher Unterschied zwischen diesen beiden HTML-Elementtypen? **(3 Punkte)**

- Block-Elemente (wie `<div>` oder `<p>`) erzwingen einen Zeilenumbruch um das Element und nehmen die komplette verfügbare Breite ein
- Inline-Elemente (wie `` oder `<a>`) erzwingen keinen Zeilenumbruch und nehmen nur so viel Breite wie nötig ein.

Diese Code-Ausschnitte sind für die nächsten zwei Teilaufgaben.

```
<body>
  <header><h1>Food</h1></header>
  <main>
    <article class="fruits">
      <h2 class="header">Fruits</h2>
      <p id="raspberry" class="berry">Raspberry</p>
      <p id="grape" class="berry">Grape</p>
      <p id="orange" class="citrus">Orange</p>
    </article>
    <article class="legumes">
      <h2 class="header">Legumes</h2>
      <p id="black" class="pea">Black Eyed Peas</p>
      <p id="green" class="pea">Green Peas</p>
    </article>
  </main>
</body>
```

CSS:

```
body {
  color: black;
}

.fruits {
  color: red;
}

.berry {
  color: green;
}

#orange {
  color: orange;
}

.pea {
  color: purple;
}

.pea:first-child {
  color: yellow;
}

#grape {
  border-color: pink;
}
```


Betrachten Sie den `<body>` eines HTML-Dokuments auf der vorherigen Seite. In welcher Farbe **(Englisch)** erscheinen die Wörter auf der Webseite, wenn das Stylesheet angewandt wird?

(9 Punkte)

| Wort | Farbe |
|-----------------|--------|
| Fruits | red |
| Raspberry | green |
| Grape | green |
| Orange | orange |
| Legumes | black |
| Black Eyed Peas | purple |

Nennen Sie **drei** CSS-Selektoren, um in dem HTML-Dokument **nur** das Element mit dem Text "Green Peas" zu selektieren. **(3 Punkte)**

```
#green
p#green
.pea#green
#green.pea
#black + p
.pea + #green
.pea ~ #green
.pea ~ .pea
.legumes > #green
article > #green
.pea:nth-child(odd)
.pea:nth-child(3)
p[id='green']
.pea:last-child
...
```

Aufgabe 5 (Reguläre Ausdrücke) (4 Punkte)

Nennen Sie **vier** Strings mit einem Match zu dem regulären Ausdruck (JavaScript):

`^[Ll]e[he]re?n?$`

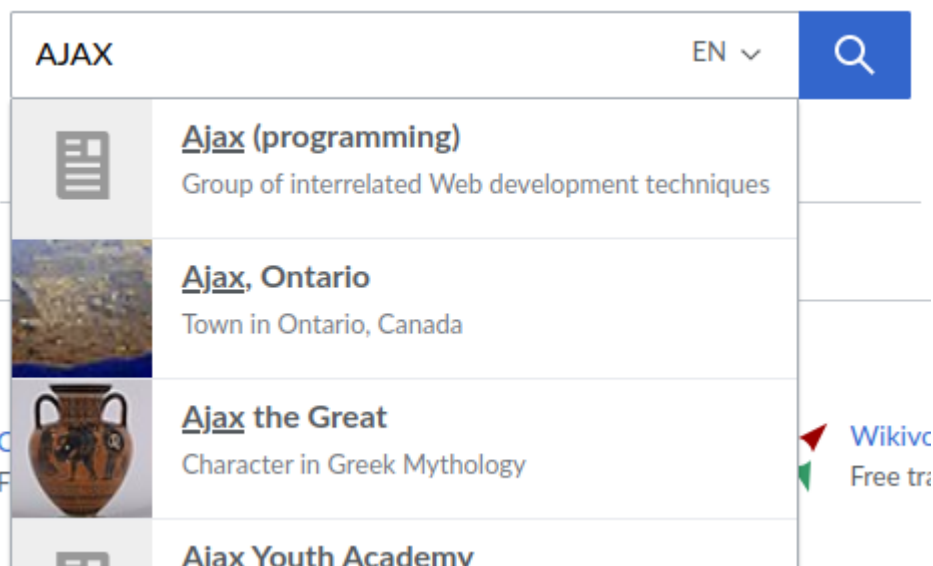
(4 Punkte)

- leer
- Leere
- leeren
- Lehre
- Lehren
- lehren
- ...

Aufgabe 6 (Ajax) (5 Punkte)

Beschreiben Sie anhand eines selbst gewählten Beispiels den Begriff "AJAX" (Asynchronous JavaScript and XML)? **(5 Punkte)**

Der Link [wikipedia.org](https://www.wikipedia.org) führt zur Wikipedia-Seite mit einem Suchfeld. Die Eingabe eines Suchbegriffs sorgt dafür, dass über JavaScript-Code "im Hintergrund" (asynchron) über eine HTTP-Anfrage nach Ergebnissen dazu gesucht wird. Die Webseite bleibt währenddessen bedienbar. Sobald ein Ergebnis zur Verfügung steht, wird dieses mittels DOM-Manipulation als Vorschlag unter dem Suchfeld hinzugefügt. Mit einem weiteren Request wird ggf. ein Vorschaubild angefragt und ebenfalls angezeigt. Der Rest der Webseite bleibt unverändert. Der Zusatz "and XML" ist etwas zu restriktiv, da auch andere Austauschformate (z.B. JSON) zum Einsatz kommen können.



Aufgabe 7 (Objektorientierung) (16 Punkte)

Der folgende Code-Ausschnitt ist für die verbleibenden drei Teilaufgaben.

```
const Shape = function (color) {
  this.type = "generic shape";
  this.color = color;
};
Shape.prototype.area = function () {
  return "unknown";
};
Shape.prototype.describe = function () {
  console.log(`A ${this.color} ${this.type} with ${this.area()} px².`);
};

class Rectangle extends Shape {
  type = "rectangle";

  constructor(color, width, height) {
    console.log("Rectangle constructor called.");
    super(color);
    this.width = width;
    this.height = height;
  }

  area() {
    console.log("area() called.");
    return this.width * this.height;
  }
}

class Circle extends Shape {
  type = "circle";

  constructor(color, radius) {
    super(color);
    this.radius = radius;
    console.log("Circle constructor finished.");
  }

  area() {
    return "TODO";
  }
}
```

Inspizieren Sie den objektorientierten Entwurf von Formen auf der vorherigen Seite. Schreiben Sie die **fünf** Zeilen auf, die bei Ausführung des folgenden Codes per `console.log()` auf die Konsole geloggt werden. **(5 Punkte)**

```
Rectangle constructor called.
```

```
Circle constructor finished.
```

```
circle
```

```
area() called.
```

```
A red rectangle with 20 px².
```

Ändern Sie die Funktion `area()` der Klasse `Circle`, sodass sie den Flächeninhalt zurück gibt. Verwenden Sie für π die Konstante `Math.PI` bei der Berechnung. **(2 Punkte)**

Hinweis: Die Formel für den Flächeninhalt eines Kreises ist $A = \pi r^2$

```
area() {  
    return Math.PI * this.radius * this.radius;  
}
```

Entwerfen Sie eine weitere Klasse `Triangle` (Dreieck), die von `Shape` erbt.

- Ein Dreieck hat als `type` den Wert "triangle"
- Ein Dreieck hat außerdem die Eigenschaften `color`, sowie `sideA`, `sideB` und `sideC` für die Länge der Seiten.
- Die Klasse hat eine spezielle Funktion `semiperimeter()`, welche als Rückgabewert den halben Umfang des Dreiecks hat, also $s = \frac{sideA + sideB + sideC}{2}$
- Die Klasse implementiert die Funktion `area()` nach Satz des Herons zur Berechnung des Flächeninhalts eines Dreiecks:
 - $A = \sqrt{s(s-a)(s-b)(s-c)}$, wobei s für den halben Umfang (`semiperimeter`) und a, b, c für die Seitenlängen stehen.
 - Hinweis: `Math.sqrt(x)` gibt die Wurzel aus x zurück.

Sie können ES5 oder ES6 Schreibweise verwenden.

(9 Punkte)

ES6-Syntax:

```
class Triangle extends Shape {
  type = "triangle";

  constructor(color, sideA, sideB, sideC) {
    super(color);
    this.sideA = sideA;
    this.sideB = sideB;
    this.sideC = sideC;
  }

  semiperimeter() {
    return (this.sideA + this.sideB + this.sideC) / 2;
  }

  area() {
    const s = this.semiperimeter();
    return Math.sqrt(s * (s - this.sideA) * (s - this.sideB) * (s - this.sideC));
  }
}
```

ES5-Syntax:

```
const Triangle = function (color, sideA, sideB, sideC) {  
  Shape.call(this, color);  
  this.sideA = sideA;  
  this.sideB = sideB;  
  this.sideC = sideC;  
  this.type = "triangle";  
};  
  
Triangle.prototype = Object.create(Shape.prototype);  
Triangle.prototype.constructor = Triangle;  
  
Triangle.prototype.semiperimeter = function () {  
  return (this.sideA + this.sideB + this.sideC) / 2;  
};  
  
Triangle.prototype.area = function () {  
  const s = this.semiperimeter();  
  return Math.sqrt(s * (s - this.sideA) * (s - this.sideB) * (s - this.sideC));  
};
```

Beispielhafter Aufruf:

```
const triangle = new Triangle("blue", 2, 4.4, 6);  
triangle.describe();
```