

# Klausur Grundlagen der Webentwicklung

PI/KI Bachelor

Prof. Dr. Kretschmer, WS 2019/2020

**Rahmenbedingungen:** 120 Minuten Bearbeitungszeit, keine Hilfsmittel. Schreiben Sie Ihre Lösungen in den dafür vorgesehenen Platz. Sollte dieser nicht ausreichen, können Sie auch die Rückseite verwenden. Vermerken Sie das dann auf der Vorderseite.

Name	Vorname	Matrikelnr.	Studiengang

1	2	3	4	5	6	7	Summe	Note

## Aufgabe 1

(40 Punkte) Beantworten Sie **kurz** folgende Fragen. Das Themengebiet steht jeweils in Klammern.

1.	Was ist ein Codepunkt? (Unicode)
2.	Welche möglichen Längen (in Bytes) kann die Codierung eines Zeichens in UTF-8 haben? (Unicode)
3.	Welche beiden Kindelemente muss das html-Element besitzen? (HTML)
4.	Was versteht man unter einem globalen Attribut? (HTML)
5.	Wie heißen die Elemente für Tabellenzeilen und normale Tabellenzellen? (HTML)
6.	Welches Attribut enthält beim a-Element die URL, auf die verwiesen wird? (HTML)
7.	Welchen Aufbau hat eine CSS-Deklaration? (CSS)
8.	Welche drei Begriffe kürzt HSL ab? (CSS)
9.	Welche Bedeutung haben die Werte 0 und 1 beim Alpha-Kanal von Farben? (CSS)
10.	Was besagt die Längenangabe 30vw ? (CSS)
11.	Was ist ein Typselektor? (CSS)
12.	Aus welchen vier ineinander geschachtelten Boxen besteht eine Box? (CSS)

13.	Wofür steht die Abkürzung DOM und was ist das? (DOM)
14.	Was liefert die Methode <code>element.querySelector(selectors)</code> ? (DOM)
15.	Welchen Vorteil hat die Verwendung von <code>addEventListener</code> gegenüber der Zuweisung an eine Eigenschaft wie z.B. <code>onclick</code> ? (DOM)
16.	Welche drei Phasen gibt es bei der Lieferung von Ereignissen im DOM-Baum? (DOM)
17.	Nennen Sie zwei der sechs Forderungen an eine REST-API, aber nicht „einheitliche Schnittstelle“ mit seinen Unterpunkten. (REST)
18.	Warum entspricht ein Endpunkt wie <code>/update/persons/123</code> nicht den REST-Anforderungen? (REST)
19.	Welche Vorteile bietet die Verwendung eines Schatten-DOMs (shadow DOM)? (DOM)
20.	Was ist eine full-stack language? (allgemein)

## Aufgabe 2

(6 Punkte) Das XML-Element `rezept` beschreibt ein Rezept. Das obligatorische Attribut `rid` identifiziert ein Rezept und hat als Wert ein `r`, gefolgt von einer oder mehreren Ziffern. Das Attribut `schwierigkeit` hat den Wert 1, 2 oder 3. Wenn es fehlt, wird der Wert 1 angenommen. `rezept` enthält die Kindelemente `name`, `zubereitung` und `zutat` in dieser Reihenfolge. Die beiden ersten kommen genau einmal vor, `zutat` beliebig oft, mindestens aber einmal. Geben Sie eine möglichst gut zutreffende (Teil-)DTD für das XML-Element `rezept` und seine Attribute an. Die Kindelemente müssen nicht definiert werden.

## Aufgabe 3

(9 Punkte) Gegeben sei folgende HTML-Datei:

<pre>&lt;!DOCTYPE html&gt;  &lt;html lang="de"&gt;  &lt;head&gt;   &lt;meta charset="utf-8" /&gt;   &lt;title&gt;Grid&lt;/title&gt;   &lt;style&gt;     main {       width: 600px;       display: grid;       grid-template-columns: 1fr 2fr 3fr;     }      div {       border: 2px solid black;       height: 50px;     }      .c1 {       grid-column: 3;       grid-row: 1;     }      .c2 {       grid-column: 1;       grid-row: 2;     }      .c3 {       grid-column: 1 / 3;       grid-row: 3;     }   &lt;/style&gt; &lt;/head&gt;  &lt;body&gt;   &lt;main&gt;     &lt;div class="c1"&gt;111&lt;/div&gt;     &lt;div class="c2"&gt;222&lt;/div&gt;     &lt;div class="c3"&gt;333&lt;/div&gt;   &lt;/main&gt; &lt;/body&gt; &lt;/html&gt;</pre>	
---	--

Skizzieren Sie im unten vorhandenen Rechteck, wie diese Datei in einem Browser dargestellt wird. Das Rechteck habe dabei die Breite des main-Elementes, also 600px. Schreiben Sie die Maße der div-Elemente in der Form Breite x Höhe neben die jeweiligen Elemente.

## Aufgabe 4

(10 Punkte)

a) Geben Sie den Wert der vier Variablen n, r1, r2, r3 an.

```
let numbers = [1, 2, 3]
numbers.push(6)
numbers.shift()
let n = numbers.join('-')
```

```
let cities = ['Berlin', 'Hamburg', 'Trier', 'Saarlouis', 'Bamberg']
```

```
let r1 = cities.filter(v => v.length > 6).map(v => v.length)
let r2 = cities.reduce((r, v) => r + v[v.length - 1], '')
let r3 = cities.filter(v => /b/i.test(v)).map(v => v[3]).join('')
```

n=	r1=	r2=	r3=
----	-----	-----	-----

b) Gegeben sei ein Feld words, das Strings enthält. Geben Sie **eine** Zeile JavaScript-Code an, die das Feld aufsteigend nach der Länge der Strings sortiert.

## Aufgabe 5

(10 Punkte) Gegeben sei das linksstehende body-Element einer HTML-Datei.

- Geben Sie unter den 3 Selektoren auf der rechten Seite die id's der durch die Selektoren selektierten Elemente an.
- Geben Sie je einen Selektor an, der nur das Element mit der id 12 bzw. 17 selektiert und nichts anderes. Dabei dürfen Sie nicht auf das Attribut id zugreifen.

<pre>&lt;body id="1"&gt;   &lt;p id="2"&gt;Brettspiele&lt;/p&gt;   &lt;ul id="3"&gt;     &lt;li id="4"&gt;       &lt;p id="5"&gt;Schach&lt;/p&gt;       &lt;ul id="6"&gt;         &lt;li id="7"&gt;Schnellschach&lt;/li&gt;         &lt;li id="8"&gt;           &lt;p id="9"&gt;Blitzschach&lt;/p&gt;           &lt;ul id="10"&gt;             &lt;li id="11"&gt;3 Minuten&lt;/li&gt;             &lt;li id="12"&gt;5 Minuten&lt;/li&gt;           &lt;/ul&gt;         &lt;/li&gt;         &lt;li id="13"&gt;Blindschach&lt;/li&gt;       &lt;/ul&gt;     &lt;/li&gt;     &lt;li id="14"&gt;Go&lt;/li&gt;     &lt;li id="15"&gt;Gobang&lt;/li&gt;     &lt;li id="16"&gt;       &lt;p id="17"&gt;Dame&lt;/p&gt;       &lt;ol id="18"&gt;         &lt;li id="19"&gt;normale Dame&lt;/li&gt;         &lt;li id="20"&gt;Laskerdame&lt;/li&gt;       &lt;/ol&gt;     &lt;/li&gt;   &lt;/ul&gt; &lt;/body&gt;</pre>	<p>a)</p> <p>ol li+li</p> <p>ul li:nth-child(3)</p> <p>li:first-child&gt;*&gt;li:first-child</p> <p>b)</p> <p>Selektor für 12:</p> <p>Selektor für 17:</p>
---	--

## Aufgabe 6

(10 Punkte) Ein benutzerdefiniertes Element repeat-text hat die Attribute text und count. Das Element enthält so viele p-Elemente, wie count angibt. Die p-Elemente enthalten als Text den Inhalt des Attributes text.

Beispiel: <repeat-text text="Hallo" count="3"></repeat-text> enthält drei p-Elemente mit dem Text Hallo.

Ergänzen Sie den JavaScript-Code, um die geforderte Funktionalität zu erreichen.

<pre>class RepeatText extends HTMLElement {   constructor() {     super()   } }  customElements.define('repeat-text', RepeatText)</pre>
---

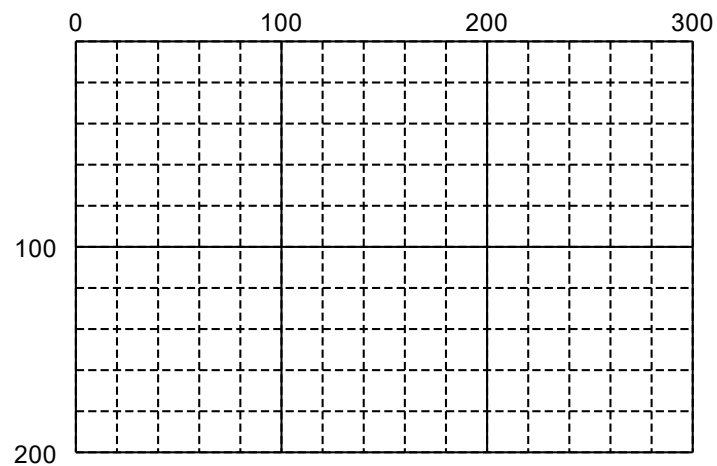
## Aufgabe 7

(15 Punkte) Skizzieren Sie die folgenden drei SVG-Graphiken auf dem dafür vorgesehenen Gitter. Folgende Eigenschaften sind per CSS festgelegt:

```
circle, line, rect, path, polygon {  
    stroke: black; stroke-width: 2; fill: transparent;  
}
```

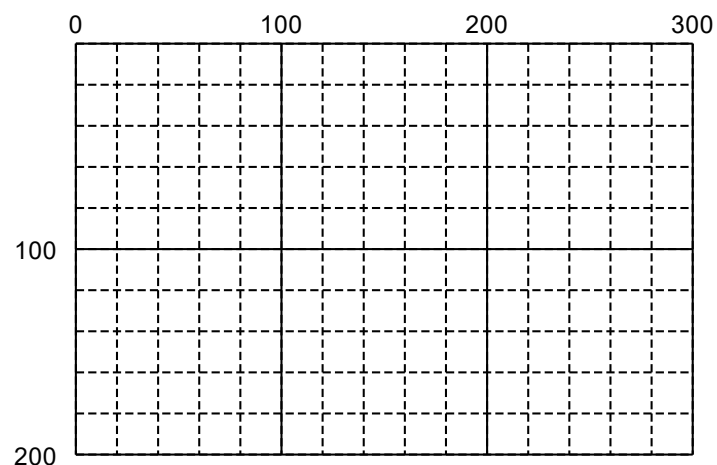
a)

```
<svg width="300" height="200" version="1.1">  
    <circle cx="100" cy="100" r="50" />  
    <circle cx="200" cy="100" r="50" />  
    <polygon points="100,0 100,50 200,0" />  
</svg>
```



b)

```
<svg width="300" height="200" version="1.1">  
    <path d="M 100,0 L 0,120 H 120 v 40 l 60,-60" />  
</svg>
```



c)

```
<svg width="300" height="200" version="1.1">
  <path d="M 0,0 L 60,60" />
  <g transform="translate(60,60)">
    <rect width="200" height="100" x="0" y="0" />
    <g transform="scale(0.5)">
      <rect width="200" height="100" x="120" y="40" />
    </g>
  </g>
</svg>
```

