

Klausur Grundlagen der Webentwicklung

PI/KI Bachelor

Prof. Dr. Kretschmer, WS 2022/2023

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 bitte auf der Vorderseite. Insgesamt sind 78 Punkte erreichbar. Dies entspricht 100%. Mit 31 Punkten haben Sie bestanden.

Name	Vorname	Matrikelnr.	Studiengang

1	2	3	4	5	6	Summe	Note

Aufgabe 1

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

1.	Was versteht man unter einem globalen Attribut? (HTML)
2.	Welche zwei Elemente werden bei ungeordneten Listen verwendet? (HTML)
3.	Welche drei Begriffe kürzt RGB ab? (CSS)
4.	Was besagt die Längenangabe 50vmin ? (CSS)
5.	Nennen Sie zwei strukturelle Pseudoklassen. (CSS)
6.	Aus welchen vier ineinander geschachtelten Boxen besteht eine Box? (CSS)
7.	Wofür steht die Abkürzung DOM und was ist das? (DOM)
8.	Nennen Sie drei primitive Datentypen. (JavaScript)
9.	Was ist ein Versprechen (promise)? (JavaScript)
10.	Wozu wird eine viewBox verwendet? (SVG)
11.	Wie nennt man die Punkte, die das Aussehen von Bézierkurven beeinflussen? (SVG)
12.	Mit welchem Element können Pfeilspitzen erzeugt werden? (SVG)

13.	Nennen Sie zwei mögliche Typen des input-Elementes. (Formulare)
14.	Nennen sie zwei gebräuchliche MIME-Typen. (HTTP)
15.	Wo wird die Angabe Transfer-Encoding: chunked verwendet und was bedeutet sie? (HTTP)
16.	Warum entspricht ein Endpunkt wie /delete/article nicht den REST-Anforderungen? (REST)

Aufgabe 2

(10 Punkte) Ein XML-Dokument enthält die Beschreibung eines endlichen Automaten. Der Automat besteht aus Zuständen, die durch Übergänge miteinander verbunden sind.

Das XML-Element `FiniteAutomaton` hat als Kindelemente erst `States`, dann `Transitions`. `States` enthält ein oder mehrere Kindelemente `State`. Das Element `State` (Zustand) hat als Inhalt beliebigen Text und kann die optionalen Attribute `start` und/oder `final` haben. Ihr Wert kann `true` oder `false` sein. Wenn eines dieser Attribute fehlt, ist sein Wert `false`. Außerdem gibt es noch das Attribut `id`, das eine eindeutige ID für den Zustand enthält.

`Transitions` hat kein, ein oder mehrere Kindelemente `Transition`. `Transition` (Übergang) ist ein leeres Element mit den Attributen `from`, `to` und `symbol`. `from` und `to` enthalten die ID je eines Zustandes. `symbol` enthält den Kleinbuchstaben `a` oder `b`.

Geben Sie eine möglichst gut zutreffende DTD für alle XML-Elemente und deren Attribute an. Wenn nichts anderes gesagt ist, sind die Attribute erforderlich.

Aufgabe 3

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

- Geben Sie unter den zwei 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 19 bzw. 13 selektiert und nichts anderes. Dabei dürfen Sie nicht auf das Attribut id zugreifen.

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

Aufgabe 4

(8 Punkte)

Geben Sie den Wert der vier Konstanten r1, r2, r3, r4 an.

```
const f = (g, x) => x * g(x)
```

```
const h = x => x + 2
```

```
const r1 = f(h, 3)
```

```
const cities = ['Konz', 'Hamburg', 'Trier', 'Saarlouis', 'Bern']
```

```
const r2 = cities
  .slice(-2).map(v => v.length)
```

```
const r3 = cities
  .filter(v => /[m-t][m-n]/.test(v))
  .reduce((r, v) => r * v.length, 1)
```

```
const r4 = cities
  .map(v => ({ n: v, l: v.length}))
  .sort((a,b) => b.l - a.l)[0].n
```

r1=	r2=	r3=	r4=
-----	-----	-----	-----

Aufgabe 5

(10 Punkte)

Schreiben Sie eine Funktion `createButton`. Die Funktion wird in einem HTML-Dokument aufgerufen und erzeugt einen Button, dessen Beschriftung über den Parameter `text` angegeben wird. Der Button erhält die Funktion `changeFirstList` als Event-Listener für das Ereignis `click`. Dann hängt die Funktion den Button als letztes Kind an das `body`-Element des Dokumentes. Die Funktion `changeFirstList` ersetzt das erste `ul`-Element im Dokument durch ein neues `ol`-Element. Dabei sollen alle Kindelemente des `ul`-Elementes umgehängt werden, so dass sie in der gleichen Reihenfolge Kindelemente des `ol`-Elements werden. Das `ul`-Element wird danach aus dem Dokument entfernt.

Ergänzen Sie den JavaScript-Code, um die geforderte Funktionalität zu erreichen. Verwenden Sie weder `innerHTML` noch `insertAdjacentHTML`.

```
function createButton(text) {  
  
  
  
  
  
  
}  
  
function changeFirstList() {
```

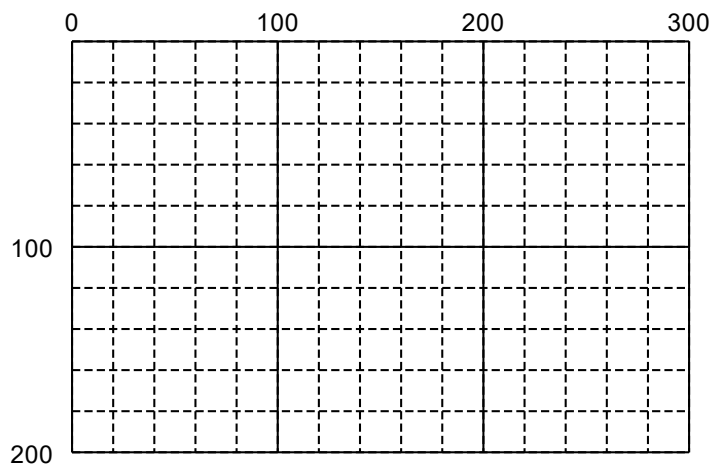
Aufgabe 6

(10 Punkte) Skizzieren Sie die folgenden zwei SVG-Graphiken auf den dafür vorgesehenen Gittern. Folgende Eigenschaften sind per CSS festgelegt:

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

a)

```
<svg width="300" height="200" version="1.1">  
  <circle cx="60" cy="60" r="40" />  
  <circle cx="200" cy="100" r="60" />  
  <polyline points="0,200 200,160 60,160 60,100" />  
</svg>
```



b)

```
<svg width="300" height="200" version="1.1">  
  <path d="M 0,0 L 120,40" />  
  <g transform="translate(-20,20)">  
    <rect width="100" height="100" x="140" y="20" />  
    <g transform="rotate(45 190 70)">  
      <rect width="100" height="100" x="140" y="20" />  
    </g>  
  </g>  
</svg>
```

