

Webentwicklung		Name	
Sommersemester xx		Semester	
Prof. Dr. S. Kruse		Matrikel-Nr.	
<i>Prüfungsart</i>	Beispielklausur	<i>Hilfsmittel</i>	Code aus den Laborübungen
<i>Prüfungsdatum</i>	xx, xx.xx.xx	<i>Max. Punkte</i>	100
<i>Prüfungsdauer</i>	90 min.	<i>Punkte</i>	
		<i>Note</i>	

Hinweise:

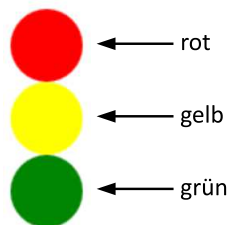
- Die Klausur umfasst 4 Seiten mit 5 Aufgaben. Prüfen Sie bitte zuerst, ob Sie alle Blätter erhalten haben.
- Geben Sie auf allen Lösungsblättern Ihren Namen, das Semester und die Matrikelnummer an.
- Falls Sie zusätzliches Papier benötigen, melden Sie sich bitte.
- Schreiben Sie bitte deutlich, sauber und leserlich und mit einem dokumentenechten Schreibgerät, also nicht mit Bleistift.
- Benutzen Sie keine rote Farbe.
- Streichen Sie Ungültiges deutlich und eindeutig durch. Verwenden Sie kein Tipp-Ex, Korrekturband u. ä.
- Bieten Sie zu einer Fragestellung nicht mehrere Lösungen an, sondern entscheiden Sie sich für eine und streichen Sie die anderen durch.
- Lösungswege müssen nachvollziehbar sein. Die Angabe von Endergebnissen genügt nicht.
- Die zu verwendende Programmiersprache ist JavaScript. Der Methodenaufruf `document.getElementById(id)` kann durch `$(id)` abgekürzt werden. Der Methodenaufruf `document.write(str)` durch `w(str)`. Das Schlüsselwort `function` durch `f`.
- Viel Erfolg!

1. Aufgabe: Canvas (20 Punkte)

Gegeben sei das folgende Canvas-Element:

```
<canvas id="c" width="100" height="200"></canvas>
```

Schreiben Sie JavaScript-Code, der in diesem Canvas-Element die Grafik einer Ampel zeichnet, bestehend aus einem roten, gelben und grünen gefüllten Kreis ohne Rand. Siehe die Abbildung unten.



Die drei Kreise haben jeweils einen Durchmesser von 50 Pixeln. Der Mittelpunkt des roten Kreises befindet sich im Canvas-Element an der Position $(x=50, y=50)$. Alle Farben werden mit maximaler Intensität gezeichnet.

2. Aufgabe: Objektorientierte Programmierung (15 Punkte)

1. Definieren Sie eine Konstruktorfunktion `function Kreis(radius)` zum Zeichnen von Kreisen mit einem Radius mit dem Wert `radius` in cm.

Die Konstruktorfunktion verfügt über eine einzelne Methode:

```
this.macheKreis = function(color).
```

Diese erzeugt einen Kreis in der Farbe `color` mit dem Radius `radius`. Das Erzeugen des Kreises wird durch eine Textausgabe in das Browserfenster simuliert. Siehe hierzu die Programmausgabe unten. Halten Sie sich bei der Definition der Klasse streng an die gegebenen Funktions- bzw. Methodenköpfe.

2. Definieren Sie eine Konstruktorfunktion `function Ampel()` zum Erzeugen einer Verkehrsampel. Verkehrsampel-Objekte werden auf der Grundlage eines Objekts vom Typ `Kreis` mit einem Radius von 25cm erzeugt, der als Prototyp dient.

Die Konstruktorfunktion verfügt über eine einzelne Methode:

```
this.macheAmpel = function().
```

Diese erzeugt drei Kreise mit den Radien 25cm in den Farben rot, gelb und grün. Siehe hierzu wieder die Programmausgabe unten. Bei der Definition der Methode werden soweit wie möglich die Eigenschaften von `Kreis` weiter genutzt, um Coderedundanz zu vermeiden. Halten Sie sich bei der Definition der Klasse wieder streng an die gegebenen Funktions- bzw. Methodenköpfe.

3. Schreiben Sie ein Hauptprogramm, in dem eine Ampel, entsprechend zu der folgenden Programmausgabe erzeugt wird.

Programmausgabe:

```
Erzeuge roten Kreis mit Radius 25
Erzeuge gelben Kreis mit Radius 25
Erzeuge grünen Kreis mit Radius 25
```

Alle Ausgaben in das Browserfenster erfolgen mit `document.write()` bzw. abgekürzt durch `w()`.

3. Aufgabe: XML (10 Punkte)

Gegeben sei die DTD `Fernsehprogramm.dtd`, die den Aufbau von XML-Dateien für das Speichern von Fernsehprogrammen beschreibt:

```
<!ELEMENT Fernsehprogramm (Sender+)>
<!ELEMENT Sender (Name, Sendung*)>
<!ELEMENT Sendung (Titel, Beschreibung)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Titel (#PCDATA)>
<!ELEMENT Beschreibung (#PCDATA)>
<!ATTLIST Sendung Tag (Mo|Di|Mi|Do|Fr|Sa|So) #REQUIRED>
<!ATTLIST Sendung Start CDATA #REQUIRED>
<!ATTLIST Sendung Ende CDATA #REQUIRED>
```

Schreiben Sie den Inhalt einer XML-Datei `Fernsehprogramm.xml`, die die oben angegebene DTD erfüllt und die Daten zur ARD-Sendung mit dem Titel `Tatort` und der Beschreibung `Die Gentleman bitten zur Kasse` enthält. Die Sendung läuft am Sonntag, beginnt um 20:15 und endet um 21:45.

Webentwicklung	Name	
Sommersemester xx	Semester	
Prof. Dr. S. Kruse	Matrikel-Nr.	

4. Aufgabe: Templates (20 Punkte)

Gegeben sei der folgende HTML-Code für eine Tabelle mit Klausurergebnissen:

```
<html>
<body>
  <table>
    <tr>
      <td>Name</td>
      <td>Geschichte</td>
      <td>Deutsch</td>
      <td>Erdkunde</td>
    </tr>
    <tr>
      <td>Otto</td>
      <td>1,0</td>
      <td>2,3</td>
      <td>3,7</td>
    </tr>
  </table>
</body>
</html>
```

Die Tabelle soll mittels Vue erzeugt und dabei in der Weise generalisiert werden, dass eine beliebige Anzahl von Zeilen dargestellt werden kann. Für jede dieser Zeilen gilt, dass zusätzlich zum Namen eine beliebige Anzahl von Fächern bzw. deren Noten dargestellt werden können soll. D. h., die Zeilen sowie die Einträge pro Zeile sollen variabel sein.

Schreiben Sie zu dem HTML-Code oben:

1. Den Code für ein Vue-Template für eine Tabelle wie oben beschrieben.
2. JavaScript-Code, der das Vue-Template nutzt, um eine Tabelle zu erzeugen, in der die Daten zu den oben angegebenen Klausurergebnissen wieder stehen. Definieren Sie zu diesem Zweck ein JavaScript-Objekt mit den Daten aus der Tabelle.

Vue kann als bereits in die HTML-Seite eingebunden betrachtet werden.

5. Aufgabe: JSON, AJAX, Zeichenketten, DOM, GUI (35 Punkte)

Sie entwickeln die Anwendung zu einer Online-Hotelsuche. Nach der Eingabe eines Orts und/oder eines Hotelnamens oder Teilen davon in einer Bedienoberfläche sollen vorhandene Hoteleinträge aufgelistet werden.

1. Schreiben Sie den Inhalt einer JSON-Datei `hotels.json`, in der die folgenden Hotels als Feld von Objektliteralen abgespeichert sind:

Name	Ort	Land
Steigenberger	Berlin	Deutschland
Novotel	Berlin	Deutschland
Kensington Hotel	London	England

Die Namen der Objekteigenschaften entsprechen dabei den in Zeile 1 angegebenen Spaltennamen.

2. Schreiben Sie JavaScript-Code für das Einlesen der in der JSON-Datei `hotels.json` gespeicherten Daten. Speichern Sie diese in einer Variablen mit dem Namen `hotels`.
3. Schreiben Sie HTML-Code für die folgende Bedienoberfläche, die die Eingabe von Orts- und Hotelnamen oder Teilen davon und die Ausgabe von Suchergebnissen erlaubt:

Ort:

Name:

Abbildung 1: Bedienoberfläche

Durch Drücken des Buttons "Suchen", wird eine Funktion `finden()` aufgerufen.

Es ist kein HTML-Rahmen aus `<html><head><body>` erforderlich. Für die Anordnung der Bedienelemente keine Tabelle oder Flexbox. Es ist ebenfalls keine Formatierung mittels CSS oder Angaben zu Fonts, Ausrichtungen und Maßen erforderlich.

4. Schreiben Sie die Funktion `finden()`. Die Funktion durchsucht mit den in der Bedienoberfläche gemachten Angaben beliebig viele in der Variable `hotels` gespeicherte Hotelangebote und gibt alle passenden Angebote unterhalb der Bedienoberfläche in ein dafür vorgesehenes HTML-Element aus. Bei der Suche wird Groß- und Kleinschreibung berücksichtigt.

Beispielausgaben für zwei unterschiedliche Eingaben von Start- und Zielort:

Ort: Berlin	Ort:
Name:	Name: Ken
Suchen	Suchen
Ergebnis: Steigenberger, Berlin, Deutschland Novotel, Berlin, Deutschland	Ergebnis: Kensington Hotel, London, England

Abbildung 2: Links und rechts: Zwei Anfrageergebnisse für zwei verschiedene Eingaben