



Name:

Vorname:

Geburtsdatum:

Matrikelnummer:

Platz:

**Klausur**  
**Informatische Werkzeuge in den**  
**Geistes- und Sozialwissenschaften 1**

15. Februar 2024

	To be used for grading, do not write here												
prob.	1.1	1.2	1.3	1.4	2.1	2.2	3.1	3.2	3.3	4.1	4.2	Sum	grade
total	4	4	4	2	5	4	6	6	12	5	8	60	
reached													



## Organisatorisches

**Bitte lesen die folgenden Anweisungen genau und bestätigen Sie diese mit Ihrer Unterschrift.**

1. Bitte legen Sie Ihren Studierendenausweis und einen Lichtbildausweis zur Personenkontrolle bereit!
2. Sie können 60 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 55 Punkte bereits als volle Punktzahl, d.h. 5 Punkte sind Bonuspunkte.
3. Es sind keine Hilfsmittel erlaubt außer eines handgeschriebenen "Spickzettels" von 1 Seite A4 einseitig.
4. Die Bearbeitungszeit beträgt genau 60 min.
5. Schreiben Sie die Lösungen direkt auf die ausgeteilten Aufgabenblätter. Andere Blätter werden nicht bewertet.
6. Wenn Sie die Prüfung aus gesundheitlichen Gründen abbrechen müssen, so muss Ihre Prüfungsunfähigkeit durch eine Untersuchung in der Universitätsklinik nachgewiesen werden. Melden Sie sich in jedem Fall bei der Aufsicht und lassen Sie sich das entsprechende Formular aushändigen.
7. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (12 Seiten inklusive Deckblatt und Hinweise) und einwandfreies Druckbild! **Vergessen Sie nicht, auf dem Deckblatt die Angaben zur Person einzutragen und diese Erklärung zu unterschreiben!**

**Erklärung:** Durch meine Unterschrift bestätige ich den Empfang der vollständigen Klausurunterlagen und die Kenntnisnahme der obigen Informationen.

Erlangen, 15. Februar 2024

.....  
(Unterschrift)

Bitte beachten Sie die folgenden Regeln, um keine Punkte zu verlieren:

- Wenn Sie eine Antwort auf einer anderen Seite fortsetzen, geben Sie bitte die Nummer der Aufgabe auf der neuen Seite mit an und verweisen Sie auf der alten Seite auf die neue.
- Begründen Sie Ihre Aussagen, wenn angebracht (wir würden gerne Teilpunkte für unvollständige Antworten geben). Wenn nicht explizit darum gebeten, antworten Sie möglichst nicht einfach mit „Ja“, „Nein“ oder „42“.

# 1 Grundlagen und Verständnis



## Aufgabe 1.1 (URIs und URNs und URLs)

4 Punkte

Was ist der Unterschied zwischen einer *URI*, einer *URL* und einem *URN*? Geben Sie außerdem an, wie diese Konzepte sich zueinander verhalten.



## Aufgabe 1.2 (11 in verschiedenen Zahlensystemen)

4 Punkte

In Stellenwert-Zahlensystemen (positional number systems) hängt der Wert einer Zahl mit mehr als einer Stelle davon ab, in welchem System wir uns genau befinden.

Geben Sie jeweils den Wert *im Dezimalsystem* an für die Zahlen  $11_2$  (Binärsystem, Basis 2),  $11_{16}$  (Hexadezimalsystem, Basis 16),  $11_8$  (Oktalsystem, Basis 8) und  $11_1$  (Unärsystem, Basis 1) an.

**Aufgabe 1.3 (HTML Requests)**

4 Punkte

Sowohl HTML GET Anfragen wie auch HTML POST Anfragen können dazu verwendet werden, einem Server Informationen zu übermitteln. Dennoch wird aus gutem Grund zwischen ihnen klar unterschieden. Nennen Sie den Unterschied zwischen GET und POST Anfragen. Welche Variante würden Sie für das Übertragen eines Passwortes verwenden? Nennen Sie außerdem noch mindestens eine weitere Sorte von HTML Anfragen.

**Aufgabe 1.4 (Escaping in Python)**

2 Punkte

Bestimmte Zeichenketten in Python tragen besondere Bedeutungen. So kann zum Beispiel die Zeichenkette "\n" dafür genutzt werden, einen Zeilenumbruch auszugeben. Allerdings kommt es manchmal auch vor, dass wir die tatsächliche Zeichenkette und nicht ihre besondere Bedeutung benötigen.

Geben Sie einen (!) Python `print`-Befehl an, der folgenden Text inklusive des Zeilenumbruchs und des nicht-Zeilenumbruchs ausgibt:

```
NEWLINE:  
NOT A NEWLINE:\n
```

## 2 Reguläre Ausdrücke



### Aufgabe 2.1 (RegEx für Polynome)

5 Punkte

Ein Polynom ist (im Rahmen dieser Aufgabe) eine Reihe von Summanden, voneinander getrennt durch Rechenzeichen, also entweder + oder -. Alle Summanden haben dabei die Form  $ax^b$ , wobei  $a$  und  $b$  jeweils beliebige natürliche Zahlen (also z.B. 7, 0 oder 314...) sein können. Der Faktor  $a$  kann allerdings auch entfallen.

#### Beispiele für Polynome:

1.  $5x^2+3x^1+0x^0$
2.  $x^4-x^3+2x^4$
3.  $x^2$

Geben Sie einen regulären Ausdruck für Polynome wie oben beschrieben an. Beachten Sie dabei, dass in einem gültigen Polynom mindestens ein Summand existieren muss und auf den letzten Summand kein Rechenzeichen folgt.

**Aufgabe 2.2 (Reguläre Ausdrücke für IP-Adressen)**

4 Punkte

Geben Sie einen regulären Ausdruck an, der gegen IPv4-Adressen matcht. Diese Adressen bestehen aus genau vier Blöcken, die durch Punkte voneinander getrennt sind. Jeder Block enthält zwischen einer und drei Ziffern von 0 bis 9.

**Beispiele:**

- 127.0.0.1
- 131.188.6.20
- 203.000.113.000

---

*Hinweis:* Blöcke in tatsächlichen IPv4-Adressen nehmen nur Werte zwischen 0 und 255 an, nicht jede beliebige Kombination aus drei Ziffern wie z.B. 999. Diesen Fakt ignorieren wir in dieser Aufgabe.

---

### 3 Digitale Dokumente



#### Aufgabe 3.1 (CSS Grundlagen)

6 Punkte

Geben Sie für jede der folgenden Anforderungen eine valide CSS Regel an. Jede Regel kann (wenn notwendig) mehrere Attribut/Wert-Paare enthalten, es sollte aber pro Aufgabe nur ein Selektor verwendet werden.



1. Geben Sie allen Hyperlinks eine durchgängige, 2px breite Border und einen Margin von 20px in alle Richtungen.



2. Setzen Sie die Schriftgröße des Elements mit der Id `findme` auf 25pt.



3. Setzen Sie sowohl Breite als auch Höhe aller Elemente auf 150px, die *sowohl* der Klasse `limW` *als auch* der Klasse `limH` angehören.

**Aufgabe 3.2 (Javascript und das DOM)**

6 Punkte

Gegeben ist folgende HTML-Website:

```
<html>
  <body>
    <ul id="getraenke">
      <li>Kaffee</li>
      <li>Tee</li>
    </ul>

    <script>
      const newNode = document.createElement("li");
      const textNode = document.createTextNode("Wasser");
      newNode.appendChild(textNode);

      const list = document.getElementById("getraenke");
      list.insertBefore(newNode, list.children[0]);
    </script>
  </body>
</html>
```

Beschreiben Sie jeweils kurz:

1. Welche interagiert das Javascript-Snippet mit dem DOM und welche Veränderungen daran führt es durch?
2. Was sehen Nutzer\*Innen auf der Website *nachdem* das Snippet bereits ausgeführt wurde?



**Aufgabe 3.3 (FizzBuzz Template)**

12 Punkte

Sie treten einen neuen Job beim Unternehmen FizzBuzz Fizzy Drinks Inc. an und Ihre erste Aufgabe ist es, „die Website bunter zu machen“.

Schreiben Sie ein `bottle` (bzw. STPL) *Template* das eine (nicht-nummerierte) Liste von Produktnamen anzeigt. Sie wissen, dass das Template eine Variable `products` übergeben bekommt. Die Elemente von `products` sind jeweils Tupel aus Produktnummer (`int`) und Produktname (`str`).

Ist die jeweilige Produktnummer ganzzahlig durch drei teilbar, so soll *Rot* für den Produktnamen verwendet werden (Hintergrundfarbe oder Textfarbe, Sie entscheiden!). Ist sie durch fünf teilbar soll *Grün* verwendet werden. Wenn sie sowohl durch drei als auch durch fünf teilbar ist, soll *Blau* verwendet werden. In allen anderen Fällen soll keine Hervorhebung passieren.

---

*Hinweis:*

- Gefragt ist nur das Template, Sie brauchen *keinen* Code für den Server.
  - Ordentlich aufgerufen soll ihr Template eine komplette (gültige) HTML-Seite zurück geben, nicht nur die Liste selbst.
  - In Abwesenheit eines Style-Sheets können Sie auch die `style` Attribute der `<li></li>` Tags für CSS-Regeln verwenden.
-



This page was intentionally left blank for extra space

## 4 Programmieren in Python



### Aufgabe 4.1 (Password Reverse-Engineering)

5 Punkte

Gegeben ist folgende Python-Funktion:

```
def check_passfile():  
    with open('passfile.txt') as pfile:  
        lines = pfile.read().splitlines()  
  
    phrase = ""  
    for i in range(len(lines)):  
        phrase = phrase + lines[i][i]  
  
    check = phrase[::-1]  
    return check == "OKAY"
```

Geben Sie einen möglichen Inhalt der Datei `passfile.txt` an, sodass die Funktion `check_passfile()` den Wert `True` zurück gibt.

**Aufgabe 4.2 (Hexadezimal-Wörter)**

8 Punkte

Zahlensysteme mit Basen  $> 10$  (wie z.B. das Hexadezimalsystem) benutzen häufig Buchstaben als Ziffern (A für 10, B für 11, ...). Sie interessiert, ob so auch ganze Wörter Zahlen sein können.

Ihnen liegt eine Datei mit dem Namen `word_list.txt` vor, die häufig benutzte Worte enthält, jeweils eines pro Zeile. Wie viele genau oder welcher Sprache ist hier nicht relevant.

Schreiben Sie ein Programm in Python, welches das längste Wort in der Datei ermittelt (und ausgibt), das gleichzeitig auch eine gültige Zahl im Hexadezimalsystem darstellt (Beispiele: "Bad" oder "ADAC").

---

*Hinweis:* Für diese Aufgabe kann z.B. die Methode `re.fullmatch(rgx, wrd)` verwendet werden, welche einen Boolean (also `True` oder `False`) zurück gibt.

---



This page was intentionally left blank for extra space



This page was intentionally left blank for extra space