

Name:

Geburtsdatum:

Matrikelnummer:

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

18. Februar 2021

	Nur zur Korrektur, bitte freilassen!										
Aufgabe	1.1	1.2	2.1	2.2	2.3	3.1	3.2	4.1	4.2	Summe	Note
Möglich	4	3	10	5	4	5	7	10	12	60	
Erreicht											

Klausurnote:

Bonuspunkte:

Endnote:

## Organisatorisches

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

1. Bitte legen Sie Ihren Studentenausweis und einen Lichtbildausweis zur Personenkontrolle bereit!
2. Die angegebene Punkteverteilung gilt unter Vorbehalt.
3. Es sind keine Hilfsmittel erlaubt außer eines handgeschriebenen "Spickzettels" von 1 Seite A4 einseitig.
4. Die Lösung einer Aufgabe muss auf den vorgesehenen freien Raum auf dem Aufgabenblatt geschrieben werden; die Rückseite des Blatts kann mitverwendet werden. Wenn der Platz nicht ausreicht, können bei der Aufsicht zusätzliche Blätter angefordert werden.
5. 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.
6. Die Bearbeitungszeit beträgt 60 min.
7. 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.
8. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (15 Seiten exklusive 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, 18. Februar 2021

.....  
(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 (Listen und Dictionaries)

4 Pkt

Erklären Sie kurz Unterschiede und Gemeinsamkeiten von *Listen* und *Dictionaries* in der Programmiersprache `python`.

## Aufgabe 1.2 (Dateigrößen und Dateneinheiten)

3 Pkt

Sortieren Sie diese folgenden Dateien der Größe nach, von der kleinsten zur größten. Geben Sie dazu nur die Nummer ( $n$ ) in der ersten Spalte an.

- |     |   |                 |
|-----|---|-----------------|
| (1) | <code>totally_legally_downloaded_movie.mkv</code> | (3.6 Gigabytes) |
| (2) | <code>datensatz.sql</code>                        | (400 Exbibytes) |
| (3) | <code>final.pdf</code>                            | (1 Mebibyte)    |
| (4) | <code>empty.txt</code>                            | (0 Byte)        |
| (5) | <code>exercise_routine.pdf</code>                 | (1025 Kibibyte) |
| (6) | <code>ling_corpus.txt</code>                      | (3.5 Gibibytes) |

## 2 Digitale Dokumente

### Aufgabe 2.1 (HTML Formular Personendaten)

10 Pkt

Schreiben Sie eine gültige HTML-Seite, die eine passende Überschrift enthält und ein HTML-Formular, das aus zwei Textfeldern mit erklärender Beschriftung, zwei beschrifteten Radio-Buttons sowie einem „Absenden“-Button besteht.

Die Textfelder sollen mit „Name:“ und „Geburtsjahr:“ beschriftet sein. Die Radio-Buttons (von denen jeweils nur einer ausgewählt sein können soll!) sollen mit „Lebendig“ und „Verstorben“ beschriftet sein.

Wird das Formular mit dem „Absenden“-Button abgeschickt, soll der Benutzer oder die Benutzerin auf eine Seite mit einer Empfangsbestätigung weiter geleitet werden.

Sie können hierfür annehmen, dass die (bereits existierende!) entsprechende Datei mit Namen `bestaetigung.html` im gleichen Verzeichnis abgelegt ist. Eine weitere Funktionalität (wie z.B. Auswertung oder Einbindung der Angaben) ist nicht notwendig.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

## Aufgabe 2.2 (Mentales CSS)

5 Pkt

Gegeben ist der Quelltext der folgenden HTML-Seite:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {background-color: green;}
      h1 {color: blue;}
      div {
        background-color : red;
        font-size: 20px;
        padding: 20px;
      }
    </style>
  </head>
  <body>
    <h1>Dies ist eine Ueberschrift!</h1>
    <div><p>Und dies ein Paragraph.</p></div>
  </body>
</html>
```

Beschreiben Sie, welche Elemente auf der fertig dargestellten HTML-Seite zu sehen sind, wie sie angeordnet sind und welche Farbe sie haben.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

**Aufgabe 2.3 (XPath)**

4 Pkt

Im Kontext von XML-Dokumenten, was versteht man unter einem *XPath*? Geben Sie ein Beispiel für einen XPath mit mindestens drei Komponenten an und erklären Sie, was damit erreicht wird.

### 3 Reguläre Ausdrücke

#### Aufgabe 3.1 (RegEx für Solfège)

5 Pkt

In der Musiktheorie und -didaktik wird oft auf die so genannte Solfège-Tonlehre zurück gegriffen. Hierbei werden den einzelnen Noten bestimmte Silben zugeordnet (traditionellerweise *do, re, mi, fa, so(l), la* und *ti*). Oft wird Musik auch (teilweise) mit diesen Silben notiert.

**Happy Birthday**

♩ = 80

Voice Lead

Hap - py birth - day to you! Hap - py birth - day to  
sol sol la sol do ti sol sol la sol re

5

you! Hap - py - birth - day dear (say name) Hap - y birth - day to  
do sol sol sol mi do ti la fa fa mi do re

9

you!  
do

#### Beispiele für Musikstücke in Solfège:

1. sol sol la sol do ti sol sol la sol re do sol sol sol mi do ti la fa fa mi do re do
2. mi re do re mi mi mi re re re mi so so mi re do re mi mi mi mi re re mi re do
3. so la so fa mi fa so re mi fa mi fa so so la so fa mi fa so re so mi do

Geben Sie einen regulären Ausdruck an, der gültige Musikstücke, also Aneinanderreihungen beliebig vieler dieser Silben, jeweils getrennt durch Whitespace, akzeptiert. Sie können davon ausgehen, dass auch die letzte Silbe von Whitespace gefolgt wird. Ihr Ausdruck soll sowohl mit der Silbe *so* als auch mit der Silbe *sol* funktionieren. Musikalische Aspekte sollen in dieser Aufgabe keine Rolle spielen.

### Aufgabe 3.2 (RegEx für Polynome)

7 Pkt

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.

## 4 Programmieren in Python

#### Aufgabe 4.1 (Ermittlung der Lebensspanne)

10 Pkt

Gegeben sind drei Dictionaries. Das erste, `names`, enthält Strings als Werte, die Namen von historischen Personen entsprechen. Die anderen beiden, `births` und `deaths`, enthalten Integers als Werte, die Geburts- bzw. Todesjahren entsprechen.

Die Schlüssel in allen drei Dictionaries sind Integers, sodass Name einer Person und deren Geburts- bzw. Todesjahre alle den gleichen Schlüssel im jeweiligen Dictionary haben:

```
names = {
    1 : "Sophie Germain",
    17 : "Emmy Noether",
    8 : "Evariste Galois",
    10 : "Leonhard Euler"
    # ...
}

births = {
    8 : 1811,
    1 : 1776,
    17 : 1882,
    10 : 1707
    # ...
}

deaths = {
    17 : 1935,
    1 : 1831,
    8 : 1832,
    10 : 1783
    # ...
}
```

Schreiben Sie ein `python`-Programm, das ermittelt, welche Person in diesem Datenset am jüngsten gestorben ist. Es soll diese Information und das Alter, in dem die Person verstarb ausgeben.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

## Aufgabe 4.2 (Ping-Pong Webserver)

12 Pkt

Geben Sie den python-Quellcode für folgende drei Routen eines `bottle`-Webservers an. Dieser Webserver soll auf einen HTML GET-Request mit dem Pfad ...

1. .../ping ein HTML-Fragment zurück geben, das nur einen Link auf den Pfad /pong enthält.
2. .../pong ein HTML-Fragment zurück geben, das einen HTML-Link auf den Pfad /ping enthält, aber darüber hinaus an Log-Datei `zugriffe.log` eine Zeile mit der aktuellen Uhrzeit anhängt. Außerdem soll auf der Seite angezeigt werden, wie oft dieser Prozess schon durchgeführt wurde (diese Information ist aus der Log-Datei zu ermitteln, benutzen Sie dafür keine Variable außerhalb der Route).
3. /reset abermals ein HTML-Fragment mit einem Link zurück auf den Pfad /ping zurück geben, aber außerdem die Log-Datei löschen und den Zähler damit auf 0 zurück setzen.

Dabei können Sie auf eine bereits implementierte Funktion `uhrzeit()` zurück greifen, die die aktuelle Uhrzeit als String zurück gibt. Vermeiden Sie dabei auch eventuelle Fehler die bei wiederholtem Aufruf der /reset-Route (wie zum Beispiel beim Löschen einer Datei, die nicht existiert) vorkommen könnten.

---

**Hinweis:** Zur Erinnerung: die `os` Bibliothek bietet die Methoden `os.path.isfile(path)` zum Überprüfen der Existenz einer Datei und `os.remove(path)` zum Löschen einer Datei an.

---

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.