

Probeklausur

Informatische Werkzeuge in den Geistes- und Sozialwissenschaften I

06. Februar 2021

The „Lösungen“ der Aufgaben in diesem Dokument sollen den Studierenden als Anfangspunkt für die Beantwortung der Aufgaben dienen. Trotz aller Bemühungen kann es zu unvollständigen oder sogar Fehlern in den „Lösungen“ kommen. Da die Korrektur und Benotung der gestellten Aufgaben niemals lediglich auf einen „Vergleich mit der Musterlösung“ hinausläuft, ist dies auch nicht so schlimm. In jedem Fall sollten die Studierenden die Lösungen nachvollziehen und im Prinzip mittels des Lehrstoffs selbst verifizieren können.

Sollten Sie „Lösungen“ finden, die Sie nicht verstehen oder sogar für fehlerhaft halten diskutieren Sie diese am besten mit den Tutoren oder auf dem Kursforum und benachrichtigen Sie die Lehrenden.

1 Grundlagen & Begrifflichkeiten

Aufgabe 1.1

2 Pkt

Nennen Sie zwei Arten von Kontrollflusselementen und geben Sie jeweils ein Beispiel in der Programmiersprache `python`.

2 min

Lösung:Kontrollflusselemente:

- Bedingte Verzweigung (**if**-statements)
- Schleifen (**for**-loops)

Aufgabe 1.2 (Algorithmen und Funktionen)

5 Pkt

Erklären Sie kurz die Begriffe *Algorithmus* und *Funktion* und machen Sie den Unterschied zwischen beiden klar.

5 min

Lösung:Eine Funktion ist ein mathematischer Begriff. Funktionen weisen allen Werten, für die sie definiert sind, einen anderen Wert zu. Zum Beispiel weist die Funktion `reverse` jeder Liste ihre Umkehrung zu.

Ein Algorithmus ist eine Reihe von Anweisungen um ein bestimmtes Problem oder eine Klasse von Problemen zu lösen. Algorithmen können sowohl in Programmiersprachen als auch in natürlicher Sprache (z.B. Kochrezepte) formuliert sein.

Ein Algorithmus ist nicht das gleiche wie eine Funktion. Es gibt mehrere Algorithmen, die die gleiche Funktion berechnen (eventuell mehr oder weniger effizient, aber gleich korrekt) aber jeder Algorithmus berechnet immer nur (maximal) eine Funktion.

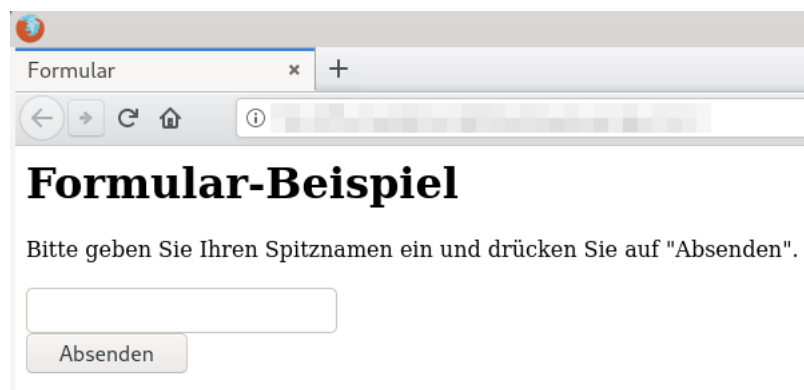
2 Digitale Dokumente

Aufgabe 2.1 (HTML-Seite mit Textfeld)

10 Pkt

Schreiben Sie eine gültige HTML-Datei, die in einem Browser etwa so dargestellt wird wie in folgendem Bild zu sehen:

10 min



Die genaue Art der Überschrift können Sie frei wählen. Das Formularelement muss *keine* besondere Funktionalitäten (wie z.B. eine `page action`) haben. Nur die Elemente sollen alle vorhanden sein.

Lösung: Hier ist die offensichtlichste Lösung. Man beachte das `<title>` Element. Dies ist im Bild zu sehen und darf nicht vergessen werden. Hier ist eine von vielen Möglichkeiten.

```
<html>
  <head>
    <title>Formular</title>
  </head>
  <body>
    <h1>Formular-Beispiel</h1>
    Bitte geben Sie Ihren Spitznamen ein und klicken Sie auf
    "Absenden".<br><br>
    <form action="send.html">
      <input type="text" name="nick"><br>
      <input type="submit" value="Absenden">
    </form>
  </body>
</html>
```

Aufgabe 2.2

6 Pkt

Geben Sie für jede der folgenden Anforderungen eine valide CSS-Regel an:

6 min

- Zentrieren Sie Überschriften (<h1>) horizontal.
- Geben Sie dem Element mit der ID "blue" eine blaue Schriftfarbe (z.B. #0000FF).
- Geben Sie Paragraphen (<p>) der Klasse hasBorder einen Rand der Breite 1px.

Lösung:Korrekte CSS-Regeln sind:

- `h1 {text-align: center;}`
 - `#blue {color:#0000FF;}`
 - `p.hasBorder {border-style: solid; border-width: 1px;}` oder `p.hasBorder {border: solid 1px}`
-

3 Reguläre Ausdrücke

Aufgabe 3.1 (Regulärer Ausdruck für ISBNs)

5 Pkt

Die *Internationale Standardbuchnummer* (ISBN) zur eindeutigen Kennzeichnung von Büchern ist (heutzutage) eine 13-stellige Zahl, geteilt in fünf Gruppen (Verlagsnummer, Titelnummer, Prüfziffer, ...), welche mit Bindestrichen getrennt werden.

5 min

Hier ein paar Beispiel-ISBNs:

978-3-86647-325-6 978-3-446-26425-0 978-3-89771-256-0
978-0-375-71449-8 978-3-499-63186-3

Geben Sie einen regulären Ausdruck an, der auf ISBNs wie oben beschrieben matcht. Genaue Vorgaben, wie viele Ziffern die einzelnen Gruppen enthalten müssen, gibt es (im Rahmen dieser Aufgabe) nicht. Sie müssen auch nicht überprüfen, dass genau 5 Gruppen im Ausdruck enthalten sind. Nur die Zahl der Zeichen insgesamt und die Zusammensetzung aus Ziffern und Bindestrichen ist hier wichtig.

Lösung: Ein solcher Ausdruck wäre zum Beispiel: $[\backslash d-]\{17\}$ – 17 Zeichen die entweder Ziffern ($\backslash d$) oder Bindestriche ($-$) sind.

Aufgabe 3.2 (RegEx für Nummernschilder)

7 Pkt

Autokennzeichen (Nummernschilder) in Deutschland bestehen aus einem “Unterscheidungszeichen” (ein bis drei Großbuchstaben für den Ort, an dem das Fahrzeug registriert wurde) und einer “Kennnummer” (ein bis zwei Großbuchstaben und ein bis vier Ziffern *ohne* führende Null).

7 min



Unterscheidungszeichen und Kennnummer sowie Buchstaben der Kennnummer und Ziffern der Kennnummer werden entweder mit einem Doppelpunkt, einem Bindestrich oder einem Leerzeichen getrennt.

Geben Sie einen regulären Ausdruck an, der Autokennzeichen nach dem obigen Muster matcht. Es ist dabei keine Voraussetzung, dass alle Trennzeichen in einem Autokennzeichen gleich sind.

Beispiele: “KA-PA-777”, “BI KJ 4990”, “LIP Q 9” und “B:AN:1337”

Lösung: Hier ist eine Lösung:

$[A-Z]\{1,3\}[\ :-][A-Z]\{1,2\}[\ :-][1-9][0-9]\{0,3\}$

4 Programmieren in Python

Aufgabe 4.1 (Falsche Route)

5 Pkt

Die `bottle`-Route in Abbildung 1 wurde mit der Intention angelegt, eine WebApplication für Addition zur Verfügung zu stellen. So soll zum Beispiel eigentlich an `/plus/37/13` der Wert 50 zurück gegeben werden. Allerdings ist etwas schief gelaufen und stattdessen wird 1337 zurück gegeben.

5 min

Erklären Sie kurz, welcher Fehler in Abbildung 1 unterlaufen ist und geben Sie eine korrigierte Fassung dieser Route an.

```
@route('/plus/<x>/<y>')
def plus(x,y):
    return y + x
```

Abbildung 1:

Lösung: Das Problem ist, dass die eingegebenen Argumente wie Zahlen verwendet werden, obwohl sie in Wirklichkeit Strings sind. Eine korrekte Variante dieser Route wäre zum Beispiel diese:

```
@route('/plus/<x>/<y>')
def plus(x,y):
    result = int(x) + int(y)
    return str(result)
```

Aufgabe 4.2

10 Pkt

Schreiben Sie eine `python`-Funktion `favoriteColors`, die eine Liste von Dictionaries übergeben bekommt. Jedes Dictionary enthält Informationen über eine Person. Jede Person hat einen Namen (Key: `"name"`), aber nur manche Personen haben eine Lieblingsfarbe (Key: `"favColor"`).

10 min

Anmerkung: Sowohl die Keys, als auch die Values sind Strings.

Ihre Funktion soll für jede Person, für die eine Lieblingsfarbe angegeben ist, den Namen der Person und die jeweilige Farbe auszugeben (**print**) Am Ende soll sie die Anzahl an Personen zurückgeben, für die eine Lieblingsfarbe angegeben ist.

Lösung: Hier ist eine mögliche Lösung:

```
def favoriteColors(dict):
    # Counting variable
    counter = 0

    # Iterate over all given dictionaries.
    for d in dict:
        if "favColor" in d:
            counter = counter + 1

    # Example. Doesn't need to be exactly this.
    print("The favorite color of", d["name"], "is", d["favColor"])

    return counter
```

Aufgabe 4.3 (Dateipfade auslesen und Dateien löschen)

10 Pkt

Schreiben Sie ein `python`-Programm, das eine Datei namens `toDelete.txt` öffnet und ausliest. In dieser Datei ist pro Zeile ein Dateipfad notiert, der zu einer Datei führt, die gelöscht werden soll.

10 min

Ihr Programm soll für alle diese Pfade überprüfen, ob eine Datei mit diesem Pfad tatsächlich existiert. Wenn ja, soll diese Datei gelöscht werden und eine entsprechende Meldung mit `print()` ausgegeben werden. Wenn nicht, dann soll eine entsprechende Fehlermeldung ausgegeben werden.

Hinweis: Die Funktionen `os.path.isfile()` und `os.remove()` könnten sich für diese Aufgabe als nützlich erweisen. Denken Sie auch an die nötigen `import` Statements.

Lösung: Hier ist eine mögliche Lösung. Man beachte die Verwendung von `.strip()`, um überflüssigen Whitespace los zu werden.

`import os`

```
# This with-syntax automatically closes the file after use.
```

```
with open("toDelete.txt") as delFile:
```

```
    for line in delFile:
```

```
        # Remove newline at end of line.
```

```
        path = line.strip()
```

```
        # If corresponding files exists:
```

```
        if os.path.isfile(path):
```

```
            # Delete file
```

```
            os.remove(path)
```

```
            print("Deleted", path)
```

```
        else:
```

```
            print(path, "does not exist!")
```
