

Name:

Geburtsdatum:

Matrikelnummer:

Nachklausur Informatische Werkzeuge in den Geistes- und Sozialwissenschaften 1

17. April 2019

	Nur zur Korrektur, bitte freilassen!												
Aufgabe	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2	4.1	4.2	4.3	Summe	Note
Möglich	2	2	4	6	8	6	4	4	3	8	8	55	
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 55 Punkte erreichen, wenn Sie alle Aufgaben vollständig lösen. Allerdings zählen 50 Punkte bereits als volle Punktzahl, d.h. 5 Punkte sind Bonuspunkte.
8. Überprüfen Sie Ihr Exemplar der Klausur auf Vollständigkeit (17 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, 17. April 2019

.....
(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

Aufgabe 1.1 In digitalen Dokumenten, was ist der Unterschied zwischen einem *Plaintext*-Format und einem *Markup*-Format? 2 Pkt

Aufgabe 1.2 Beschreiben Sie kurz die Rollen von „Servern“ und „Clients“ im World Wide Web und wie diese typischerweise miteinander interagieren. 2 Pkt

Aufgabe 1.3 Nennen Sie zwei Arten von Kontrollflusselementen und geben Sie jeweils ein Beispiel in der Programmiersprache `python`. 4 Pkt

2 HTML und CSS

Aufgabe 2.1 Schreiben Sie eine gültige HTML-Datei, die in einem Browser wie folgt 6 Pkt dargestellt wird:



Die genaue Art der Überschrift können Sie frei wählen. Das Wort „Hier“ soll ein Link sein, der auf folgende Seite führt: <https://tiergarten.nuernberg.de>

Aufgabe 2.2 Schreiben Sie eine gültige HTML-Seite, die eine passende Überschrift enthält, 8 Pkt
einen kurzen Text der nach dem Lieblingstier fragt und letztlich ein HTML-Formular, das aus einem Dropdown-Menü und einem „Absenden“-Button besteht.

Im Dropdown-Menü sollen zwei verschiedene Tierarten oder mehr auswählbar sein. Wird das Formular mit dem „Absenden“-Button abgeschickt, soll der Benutzer oder die Benutzerin auf eine Seite mit einer Empfangsbestätigung weiter geleitet werden (siehe auch Aufgabe 2.1).

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

Aufgabe 2.3 Geben Sie für jede der folgenden Anforderungen eine valide CSS-Regel an: 6 Pkt

- Geben Sie allen `<p>` und `<h1>`-Elementen mit einer(!) Regel eine rote Schriftfarbe.

- Geben Sie dem Element mit der ID "imblue" eine blaue Hintergrundfarbe.

- Geben Sie allen `<div>`-Elementen einen Margin von 15px in alle Richtungen.

- Zentrieren Sie allen Text in Elementen der Klasse `centered` horizontal.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

3 Reguläre Ausdrücke

Aufgabe 3.1 In dieser Aufgabe sollen Sie einen regulären Ausdruck (Regex) für korrekte International Bank Account Numbers (IBANs) angeben. Die genaue Form von IBANs ist abhängig vom Land, wir suchen nach einem regulären Ausdruck für korrekte IBANs aus allen DACH-Ländern (d.h. Deutschland, Österreich und Schweiz). 4 Pkt

Deutsche IBANs sind die Buchstaben “DE” gefolgt von 20 Ziffern, Österreichische IBANs sind die Buchstaben “AT” gefolgt von 18 Ziffern und Schweizer IBANs sind die Buchstaben “CH” gefolgt von 19 Ziffern.

Beispiele:

- DE89370400440532013000 (Deutschland)
- AT611904300234573201 (Österreich)
- CH9300762011623852957 (Schweiz)

Aufgabe 3.2 Alle IBANs haben ein „elektronisches Format“ (ohne Leerstellen, z.B. 4 Pkt DE89370400440532013000) und ein „Druckformat“ (mit Leerstelle nach jeweils vier Zeichen, z.B. DE89 3704 0044 0532 0130 00).

Geben Sie einen regulären Ausdruck an, der für deutsche IBANs (!) sowohl gegen das elektronische Format als auch gegen das Druckformat korrekt matcht.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

4 Programmieren in Python

Aufgabe 4.1 Was gibt das folgende python Programm aus, wenn es ausgeführt wird? 3 Pkt

```
foo = "IWGS"  
bar = foo + foo + foo  
baz = bar + bar + bar  
  
print(len(baz))
```

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Aufgabe 4.2 Schreiben Sie eine python-Funktion `dict2list`, die zwei Argumente (ein Dictionary und einen Booleschen Wert) übergeben bekommt und eine Liste zurück gibt. Wird der Funktion `True` als Boolescher Wert übergeben, so soll die Liste alle *Keys* des Dictionarys enthalten. Wird stattdessen `False` übergeben, soll die Liste alle *Values* des Dictionarys enthalten. 8 Pkt

Beispiel: Betrachten Sie den folgenden Fall:

```
# Source: https://en.wikipedia.org/wiki/Mae\_Jemison
```

```
exampleDictionary = {  
    "name" : "Mae Carol Jemison",  
    "birthday" : "1956-10-17",  
    "gender" : "female",  
    "occupation" : "astronaut",  
    "mission" : "STS-47"  
}
```

Wenn Sie die Funktion mit `dict2list(exampleDictionary,True)` aufrufen, soll die Liste `['name', 'birthday', 'gender', 'occupation', 'mission']` zurück gegeben werden.

Bei dem Aufruf `dict2list(exampleDictionary,False)` stattdessen hingegen die Liste `['Mae Carol Jemison', '1956-10-17', 'female', 'astronaut', 'STS-47']`

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Aufgabe 4.3

8 Pkt

Schreiben Sie ein Programm in `python` das den Nutzer oder die Nutzerin nach einem Passwort fragt. Ihnen steht dabei eine `validate`-Funktion zur Verfügung, die ein Argument (String) nimmt und entweder `True` oder `False` zurück gibt, je nachdem ob es sich um das korrekte Passwort handelt oder nicht. Diese `validate`-Funktion müssen Sie *nicht* implementieren.

Gibt der Nutzer oder die Nutzerin das korrekte Passwort ein, so soll „Zugang gewährt!“ ausgegeben werden. Ist das eingegebene Passwort falsch, soll Ihr Programm erneut fragen, jedoch maximal zehn Mal. Ist das Passwort auch bei der zehnten Eingabe nicht korrekt, soll das Programm „Zugang verweigert!“ ausgeben.

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.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.