Name:

Geburtsdatum:

Matrikelnummer:

Klausur
Informatische Werkzeuge in den Geistes- und Sozialwissenschaften 1

06. Februar 2020

	Nur zur Korrektur, bitte freilassen!										
Aufgabe	1.1	1.2	2.1	2.2	3.1	3.2	4.1	4.2	4.3	Summe	Note
Möglich	4	4	8	4	5	8	5	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 (13 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, 06. Februar 2020 .....

(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“.

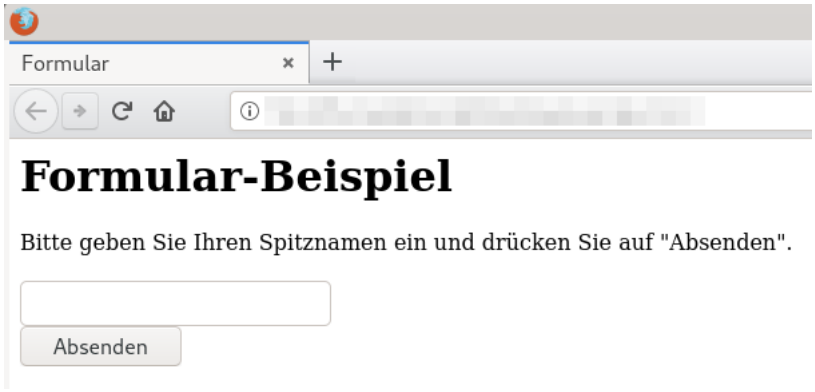
1 Grundlagen und Verständnis

Aufgabe 1.1 (HTML und XML)
Erklären Sie kurz sowohl die wichtigsten *Gemeinsamkeiten* von und die wichtigstem *Unterschiede* zwischen HTML und XML. Wie verhalten sich die beiden zueinander? 4 Pkt

Aufgabe 1.2 (Bäume)
Was ist ein *Baum* im Kontext von Informatik? Welche Bestandteile haben Bäume? Erklären Sie außerdem kurz die Begriffe *Wurzel* und *Blatt*. 4 Pkt

2 HTML und CSS

Aufgabe 2.1 (HTML-Seite mit Textfeld)
Schreiben Sie eine gültige HTML-Datei, die in einem Browser etwa so dargestellt wird wie in folgendem Bild zu sehen: 8 Pkt



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.

Diese Seite wurde für mehr Platz absichtlich leer gelassen.

Aufgabe 2.2 (CSS für Überschriften)
Geben Sie vier CSS-Regeln an, die folgende Änderungen bewirken, wenn sie in eine HTML-Datei eingebunden werden: 4 Pkt

1. Färben Sie alle `<h1>`-Überschriften rot.
2. Zentrieren Sie `<h2>`-Überschriften horizontal mittig auf der Seite.
3. Setzen Sie die Schriftgröße für `<h3>`-Überschriften auf `10pt`.
4. Geben Sie `<h4>`-Überschriften einen `margin` von 35 Pixeln in alle Richtungen.

3 Reguläre Ausdrücke

Aufgabe 3.1 (Regulärer Ausdruck mit Klammerung)
Geben Sie einen regulären Ausdruck an, der auf Ausdrücke matcht, die aus einer Reihe (der Länge 0 oder länger) von beliebig Kleinbuchstaben in genau einem Paar eckiger Klammern (`[]`) bestehen. Dabei sollen ebenfalls keine Vokale (a,e,i,o,u) oder Umlaute (ä,ö,ü,ß) vorkommen. 5 Pkt

Beispiele: `[]`, `[q]`, `[rhythm]`, `[nrnbrg]`, `[lynx]`, `[fcknzs]`, `[twttr]`

Aufgabe 3.2 (Regulärer Ausdruck für DOIs)

Um auf digitale Objekte (wie Journals, Berichte, Paper, Artikel, ...) verlässlich verweisen zu können, wurde von der International Organization for Standardization (ISO) im Jahr 2012 ein Standard für *Digital Object Identifiers* (**DOI**) veröffentlicht. 8 Pkt

Ein **DOI** besteht aus einem Präfix und einem Suffix, die von einem „`/`“ getrennt werden. Das Präfix beginnt (für unsere Zwecke) immer mit dem String „`10.`“ gefolgt von einer ganzen Zahl ≥ 1000. Das Suffix ist beliebig und unterliegt keinerlei Einschränkungen. In der Praxis ist es oft numerisch, der Standard erlaubt allerdings jede (nichtleere) Aneinanderreihung von Zeichen.

In normalen Texten ohne Hyperlinks wird ein **DOI** oft mit dem String „`doi:`“ eingeführt, wie auch in den folgenden Beispielen:

```
doi:10.1000/182
doi:10.20347/WIAS.PREPRINT.2431
doi:10.1016/j.disc.2012.08.018
doi:10.1007/978-3-662-44199-2_5
doi:10.1007/s10817-012-9271-4
```

Geben Sie einen regulären Ausdruck an, der auf textuelle **DOI**-Ausdrücke wie oben beschrieben (also inklusive „`doi:`“) matcht.

4 Programmieren in Python

Aufgabe 4.1 (Falsche Route)

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 Pkt

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:

Aufgabe 4.2 (Dateipfade auslesen und Dateien löschen)

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 Pkt

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.

Aufgabe 4.3 (Bäume mit lxml)

12 Pkt

Setzen Sie sich genau mit folgendem Schnippsel python-Code auseinander. Es wird die in der Vorlesung besprochene `lxml`-Bibliothek verwendet um iterativ (d.h. in mehreren Durchläufen) eine Baum-Datenstruktur aufzubauen, in der jeder Knoten in seinem .text-Attribut einen String enthält, der einer Zahl entspricht.

```
from lxml import etree

# Lege die Wurzel des Baumes an
root = etree.Element("node")
root.text = "0"
counter = 1

# Tue genau zwei Mal folgendes:
for _ in range(0,2):

    # Fuer "root" und alle seine Nachfahren tue folgendes:
    for element in root.iter():
        # Wenn der Knoten keine Kinder hat...
        if len(element) == 0:
            # ... fuege diesem Knoten (element) zwei Kinder hinzu ...
            fst_child = etree.SubElement(element, "node")
            snd_child = etree.SubElement(element, "node")

            # ... und gebe allen neuen Kindern den aktuellen counter als text.
            for child in element:
                child.text = str(counter)
                counter = counter + 1
```

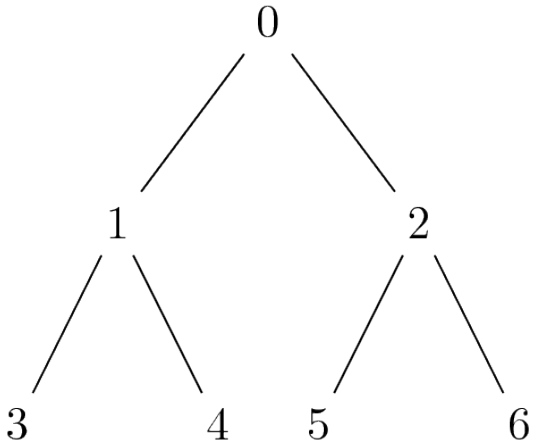
Wird der obige Code ausgeführt, so ist es möglich, den dort generierten Baum mit der Wurzel root wie folgt darzustellen (einmal in XML-Syntax und einmal als Grafik):

```
<node>
        0

        <node>
                1
                <node>3</node>
                <node>4</node>
        </node>

        <node>
                2
                <node>5</node>
                <node>6</node>
        </node>

</node>
```



Eine bildliche Darstellung des links beschriebenen Baumes.

Schreiben Sie nun eine python-Funktion renameNodes, die den Wurzelknoten eines solchen Baums als Argument bekommt. Die Funktion soll alle Knoten in dem Baum besuchen und deren .text-Attribut ersetzen. Dabei soll der neue Wert des Attributs "gerade" sein, wenn der ursprüngliche Wert des Attributs eine gerade Zahl darstellt. Andernfalls soll der neue Wert "ungerade" sein. Bedenken Sie, dass der Wert des .text-Attributs ein String ist, auch wenn er nur Ziffern enthält.

Hinweis: Alle Methoden und Funktionen, die Sie für die Beantwortung dieser Aufgabe benötigen, werden auch im obigen Codeschnippsel verwendet und werden deshalb hier nicht noch einmal aufgezählt.

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.