

Test: Python

Name: _____

Modul: Scripting (FS 2016)

Klasse: TSBE 14 - Systemtechnik

Datum: Samstag, 28. Mai 2015

Zeitdauer: 45 Minuten

Hilfsmittel: Cheatsheet (3 A4 Blätter)

Viel Erfolg!

Punkte: _____ / 15

Note: _____

Aufgabe 1: Single-Choice (3 Punkte)

Kreuze jeweils **eine** Lösung an.

Welche der folgenden Aussagen zu Zahlen in Python (Version 3) ist **wahr**?

- ☐ Fließkommazahlen (Bsp. 2.781828 ...) können theoretisch beliebig gross werden.
- ☐ Der Ausdruck `7 / 2` ergibt als Resultat: 3
- ☐ Der Ausdruck `2 * 3` ergibt als Resultat: 6
- ☐ `12 < 5` ergibt als Resultat: True

Welche der Aussage zu Manipulationen an Zeichenketten in Python ist **falsch**?

- ☐ Der Ausdruck `"I've been looking for freedom".split(" ")[3:5]` ergibt: `['for', 'freedom']`
- ☐ Der Ausdruck `"I've been looking for freedom"[0]` ergibt: `'I'`
- ☐ Der Ausdruck `"I've been looking for freedom".split(" ")[0]` ergibt: `'I'`
- ☐ Der Ausdruck `"I've been looking for freedom".split()[-1]` ergibt: `'for'`

Mit welchem der folgenden Konstrukte kann man **keine** Schleife programmieren?

- ☐ `for`-Loop mit einer mit einer Liste als Argument.
- ☐ Mit beliebig vielen `elif`-Klauseln.
- ☐ `while`-Loop mit einer Zählervariabel.
- ☐ Rekursive (sich selbst aufrufende) Funktion mit einer Abbruchsbedingung.

Aufgabe 2: Python-Skript verstehen (4 Punkte)

Lese das folgende Python-Skript `my_script.py`:

```
#!/usr/bin/python3

print('Hello', "Great", """"Universe""")

vorname = "Karin"
nachname = "Meier"
sortierung = 'normal'
if sortierung == 'sortorder':
    namensformat = "{1}, {0}"
else:
    namensformat = "{0} {1}"
print("Hallo", namensformat.format(vorname, nachname))

print("Rechnung 1:", 1 == 1)
print("Rechnung 2:", "1" > "2")
print("Rechnung 3:", int("2") ** 3)

zahlen = {1: 'eins', 2: 'zwei', 3: 'drei'}
for key, value in zahlen.items():
    print('{}: {}'.format(key, value))

print("Ein Element:", zahlen[1])

zahl = '2.718281828'
try:
    ganzzahl = int(zahl)
    print("Eine Ganzzahl:", ganzzahl)
except ValueError:
    try:
        fliesskommazahl = float(zahl)
        print("Eine Dezimalzahl:", fliesskommazahl)
    except ValueError:
        print("Keine gültige Zahl")
```

Was ist die Ausgabe wenn man `my_script.py` ausführt?

\$ `python3 my_script.py`

Aufgabe 3: Code-Analyse (4 Punkte)

Lese das folgende Python-Skript `codeanalyse.py`, welches ein erster Prototyp für ein Code-Analyse-Tool ist:

```
1 #!/usr/bin/python3
2 filename = "codeanalyse.py"  # Dieses Skript selber
3 with open(filename) as f:
4     # Ganzer Dateiinhalt einlesen
5     inhalt = f.read()
6     zeilen = inhalt.split('\n')
7     codezeilen = len(zeilen)
8     kommentare = 0
9     for zeile in zeilen:
10         zeile = zeile.strip()
11         if '#' in zeile:
12             kommentare += 1
13
14     if kommentare > (codezeilen * 0.6):
15         print("Sehr gut dokumentierter Code")
16     elif kommentare > (codezeilen * 0.3):
17         print("Gut dokumentierter Code")
18     else:
19         print("Ungenügend dokumentierter Code, bitte nachbessern!")
```

Was ist die Ausgabe wenn man `codeanalyse.py` ausführt?

\$ `python3 codeanalyse.py`

Beschreibe die Funktion des Skriptes `codeanalyse.py` möglichst detailliert:

Beschreibe einen möglichen Fall, bei welchem der Code-Analyse-Prototyp zu einem falschen oder ungenauen Ergebnis kommt:

Aufgabe 4: Python-Skript schreiben (4 Punkte)

Implementiere eine Funktion mit dem Namen `zeilenlaengen` in Python, welches für eine beliebige Textdatei für jede Zeile die Zeilenlänge berechnet, so dass folgende Verwendung (siehe Zeile 6) möglich ist:

```
1 #!/usr/bin/python3
2 filename = input("Gebe einen Dateinamen an: ")
3 zu_lang = 0
4 limit = 80
5
6 for laenge in zeilenlaengen(filename):
7     if laenge > limit:
8         zu_lang += 1
9
10 print("Die Datei '{}' enthält {} zu lange Zeilen.".format(filename, zu_lang))
```

Beachte folgende Vorgaben:

- Die Funktion `zeilenlaengen` gibt eine Liste mit Zahlen zurück.
- Die einzelnen Zahlen entsprechen der jeweiligen Zeilenlängen.
- Der Dateiname wird im Funktionsparameter `filename` übergeben.

```
def zeilenlaengen ( filename ) :
```

Schreibe die Zeilen 6 bis 8 so um, dass für alle zu langen Zeilen zusätzlich die betreffende Zeilennummer mit einem kurzen Hinweis auf der Konsole ausgegeben wird: