IWGS1_WS1920_Zettel3

July 7, 2020

1 Informatische Werkzeuge in den Geistes- und Sozialwissenschaften I

1.1 Hausaufgabe 3 (Kontrollfluss in Python, 100 Punkte)

Erschienen: 02.11.2019
Abgabe bis: 10.11.2019

Bitte laden Sie Ihre Notebooks bis 23:59 Uhr am Abgabetag in Ihrer Übungsgruppe bei StudOn hoch.

Wenn Ihnen einige der hier verwendeten Konzepte unbekannt sind oder Sie nicht wissen, wie Sie fortfahren sollen, können Sie die Vorlesungsunterlagen zu Rate ziehen oder jederzeit Fragen im Forum stellen, im Tutorium nachfragen, oder Ihrem Tutor eine Mail schreiben.

1.1.1 Aufgabe 3.1 (Input von Menschen, 15 Punkte)

Eine der wichtigsten Sachen, die es über eine Programmiersprache zu lernen gibt, ist, wie Eingaben von Nutzer*innen eingelesen werden können. Schließlich kann nur so ein Programm wirklich interaktiv werden. Eine Möglichkeit für diese Zwecke in Python ist die `input()`-Funktion.

Zum Beispiel: Wenn Sie in Ihrem Programm `antwort = input("Mögen Sie Haie?")` schreiben, wird von Python eine Eingabe von dem Menschen vor dem Computer abgewartet (in Notebooks z.B. in Form eines Textfeldes). Der String (= die Zeichenkette), die eingegeben wurde, ist danach in der Variable `antwort` gespeichert und kann wie jeder andere Wert in einer Variable verwendet werden.

Aufgabe: Schreiben Sie ein Programm, das die Person vor dem Bildschirm mit einer generischen Botschaft grüßt. Danach soll nach dem Namen der Person gefragt und das Ergebnis in einer Variable gespeichert werden. Zum Schluss soll eine Verabschiedungsnachricht geprintet werden, die den eingegebenen Namen der Person benutzt.

[10]:
```
# Ihr Code hier! (Doppelklick zum Editieren, Shift + Enter um die Zelle␣
↪auszuführen)
```

1.1.2 Aufgabe 3.2 (Kontrollfluss -- Verzweigung, 25 Punkte)

Das nächste wichtige Konzept ist *Kontrollfluss*. Ein Programm, das jedes Mal das gleiche tut wird schnell langweilig. Wir wollen meistens Programme schreiben, die unter unterschiedlichen Umständen unterschiedliche Dinge tun. Ein wichtiges Tool in Python dafür sind *Bedingte Anweisungen* und *Verzweigungen*, auch bekannt als `if / else` Anweisungen.

Aufgabe: Schreiben Sie ein Programm, das die Person vor dem Bildschirm fragt, ob sie ein Lieblingstier hat. Wenn die Antwort `"Ja"` lautet, soll eine weitere Frage gestellt werden, nämlich um welches Tier es sich handelt. Da Faultiere (zumindest im Rahmen dieser Aufgabe) die süßesten Tiere sind, soll das Programm `"Awww!"` printen, wenn die Antwort auf die zweite Frage `"Faultier"` war, und bei jeder anderen Antwort lediglich `"Cool!"`. Wenn die Antwort auf die erste Frage nicht `"Ja"` war, soll das Programm einfach nur `"Okay. Auf Wiedersehen."` oder vergleichbares printen und nichts weiter tun.

[11]:
```
# Ihr Code hier! (Doppelklick zum Editieren, Shift + Enter um die Zelle␣
↪auszuführen)
```

Hinweis:
Denken Sie daran, dass Python *whitespace-sensitiv* ist. Es spielt also eine wichtige Rolle, dass Sie Ihre Anweisungen korrekt einrücken. Zu viel oder zu wenig Einrückung kann an der falschen Stelle unerwartetes Verhalten oder Fehler verursachen.

1.1.3 Aufgabe 3.3 (Kontrollfluss -- Schleifen, 25 Punkte)

Einer der größten Vorteile, Berechnungen von einem Computer durchführen zu lassen ist, dass Computer leicht und schnell eine Aktion sehr oft hintereinander ausführen können. In Python können wir dafür zum Beispiel *`for`-Schleifen* verwenden. Eine solche Schleife besteht normalerweise aus einem *Schleifenkopf*, der beschreibt wie oft die Schleife durchgeführt werden soll (und eventuell eine sogen. "Laufvariable" benennt, oft `i` oder `x` genannt, mit der man sich auf den aktuellen Schleifendurchlauf beziehen kann) und einem (eingerückten) *Schleifenkörper*, der beschreibt, was in jedem Durchlauf geschehen soll. Hier ein Beispiel:

[12]:
```
# range(n) ist hier sehr nützlich. Es enthält alle Werte von 0 bis (n-1)
for i in range(5):
    print("Das hab ich dir schon", i, "Mal gesagt!")
```

```
Das hab ich dir schon 0 Mal gesagt!
Das hab ich dir schon 1 Mal gesagt!
Das hab ich dir schon 2 Mal gesagt!
Das hab ich dir schon 3 Mal gesagt!
Das hab ich dir schon 4 Mal gesagt!
```

Aufgabe: Schreiben Sie ein Programm, dass eine Schleife 100 mal durchläuft. Beim ersten Durchlauf soll `"Es sind noch 99 Flaschen Limo übrig!"` geprintet werden, beim zweiten Durchlauf dann `"Es sind noch 98 Flaschen Limo übrig"` und so weiter, bis runter auf 0. Wie müssen Sie dafür die Laufvariable benutzen?

[13]:
```
# Ihr Code hier! (Doppelklick zum Editieren, Shift + Enter um die Zelle␣
↪auszuführen)
```

1.1.4 Aufgabe 3.4 (Verständnisaufgabe 2, 35 Punkte)

Für diese Aufgabe werfen wir nochmal ein kurzes Spotlight auf zwei nützliche Konstrukte in Python. Sie haben in Aufgabe 3.2 schon mit bedingten Anweisungen mit Verzweigungen in Form von `if / else` gearbeitet. Hier nur eine kurze Erweiterung des Konzeptes. Oft ist die Situation, dass abgefragt werden soll, welche Bedingung aus einer Reihe von Bedigungen genau zutrifft. Um das Schachteln von mehreren `if`s zu vermeiden (was schnell unüberschaubar würde), gibt es in `if`s (die nur unter einer Bestimmten *Bedingung* aufgeführt werden) neben `else` (in allen Fällen *außer* in der Bedingung des ursprünglichen `if`s) auch so genannte `elif`s (kurz für `else if`), mit denen zwischen dem `if` und einem eventuellen `else` auch andere Bedingungen abgefragt werden können. Hier dazu ein Beispiel:

[14]:
```
x = 0 # Hier könnte auch jede andere Zahl eingesetzt werden.

# Normaler If-Block:
if x > 0:
    print("x ist größer als 0.")
# Elif-Block, wird ausgeführt, wenn "x > 0" falsch ist und "x < 0" wahr ist.
elif x < 0:
    print("x ist gleiner als 0.")
# Zweiter Elif-Block. Sie können beliebig viele von diesen verwenden.
elif x == 0:
    print("x ist genau gleich 0.")
# Else-Block, wird ausgeführt wenn keine der If- oder Elif-Bedingungen zutrifft.
else:
    print("x ist weder größer noch kleiner als 0 und auch nicht gleich 0.")
    print("Wie haben Sie das geschafft?")
```

```
x ist genau gleich 0.
```

Als nächstes werden wir kurz über Datentypen reden. Wie andere Programmiersprachen (z.B. Java) auch, unterscheidet Python Beispielsweise zwischen Strings (= Zeichenketten) und Zahlen. Manche Befehle in Python funktionieren nur korrekt, wenn alle argumente den korrekten Typ haben. So können Sie zum Beispiel mit `+` zwei Zahlen addieren (`3 + 4 == 7`) oder zwei Zeichenketten aneinander hängen (`"IW" + "GS" == "IWGS"`). Wenn sie aber `+` mit einem String und einer Zahl verwenden, wird Python einen Fehler melden.

Um dieses Problem zu umgehen, gibt es die Möglichkeit, zwischen Zahlen und Zeichenketten hin- und her- zu konvertieren. Wenn Sie zum Beispiel eine Zahl in der Variable `number` gespeichert haben, können Sie mit `str(number)` einen String draus machen. Haben Sie einen String in der Variable `text` haben, der als Ganz- oder Kommazahl interpretiert werden kann (z.B.: `text = "0.48"`), dann können Sie mit `float(text)` diesen in eine tatsächliche Zahl verwandeln.

Beide Funktionen (sowohl `str()` als auch `float()`) können selbstverständlich nicht nur mit Variablen verwendet werden, sondern auch mit Zahlen / Strings direkt.

[15]:
```
# Dieses Beispiel funktioniert nicht. String + Zahl ist nicht definiert, wirft␣
 ↪einen Fehler.
# print("Vier: " + 4.0)

# So ist es richtig: Erst wird 4.0 zu einem String umgewandelt, der dann mit␣
 ↪einem anderen verkettet werden kann.
print("Vier: " + str(4.0))

# Diese Zeichenkette stellt eine Zahl dar und kann umgewandelt werden.
print(float("3.1415"))

# Diese Zeichenkette ist keine Zahl und würde einen Fehler schmeißen.
# print(float("falsch"))
```

```
Vier: 4.0
3.1415
```

Nun zur eigentlichen Aufgabe. In der nächsten Zelle finden Sie ein fertiges interaktives Programm von Ihrer Kollegin, Beatrice Beispiel, zur Umrechnung zwischen der Fahrenheit- (beliebt im englischsprachigen Raum, insbesondere in den USA) und der Celsius-Temperaturskala (überall sonst). Die (korrekten) Formeln für die Umwandlung zwischen den beiden Skalen sind wie folgt:

$$[°F] = [°C] \cdot \frac{9}{5} + 32 \qquad [°C] = ([°F] - 32) \cdot \frac{5}{9}$$

Leider, so Beatrice, gibt das Programm nicht immer korrekte Ergebnisse. Sie selbst kann den Fehler selbst nach intensiver Suche nicht finden und bittet Sie um Hilfe.

Aufgabe: Machen Sie sich mit dem Programm in der nächsten Zelle vertraut. Welche Sorte inkorrekter menschlicher Input wird abgefangen? Welche nicht? Gehen Sie danach sicher, dass tatsächlich ein Fehler vorliegt. Welche Ergebnisse werden falsch ausgegeben? Welcher Fehler ist Beatrice beim Programmieren unterlaufen? Sammeln Sie alle ihre Erkenntnisse in der Markdown-Zelle nach dem Programm und korrigieren Sie das Programm selbst, sodass es ab sofort korrekte Ergebnisse ausgibt.

[ ]:
```
print("Willkommen beim Beispieltemperaturkonverter!")

# Zunächst bitten wir um Input von Start- und Zielskala
skala_von  = input("Von welcher Skala wollen Sie konvertieren?")
skala_nach = input("In welche Skala wollen Sie konvertieren?")

# Hier lesen wir einen String ein und wandeln ihn in eine Zahl um.
temp = float(input("Was ist die Temperatur in Grad " + skala_von + "?"))

if skala_von == skala_nach:
    # Wenn Start- und Zielskala identisch sind ist keine Umrechnung notwendig.
    print("Das sind exakt " + temp + "° " + skala_nach)

elif skala_von == "Celsius":
    if skala_nach == "Fahrenheit":
```

```
        # Berechnung nach Formel.
        temp_f = temp * (9/5) + 32

        # Ergebnis in einen String umwandeln und mit erklärendem Text ausgeben.
        print("Das sind exakt " + str(temp_f) + "° Fahrenheit")
    else:
        # Wenn die Zielskala weder Celsius noch Fahrenheit ist, liegt ein␣
↪Fehler vor.
        print("Unbekannte Skala: " + skala_nach)

elif skala_von == "Fahrenheit":
    if skala_nach == "Celsius":
        # Berechnung nach Formel.
        temp_c = temp - 32 * (5/9)

        # Ergebnis in einen String umwandeln und mit erklärendem Text ausgeben.
        print("Das sind exakt " + str(temp_c) + "° Celsius")
    else:
        # Wenn die Zielskala weder Celsius noch Fahrenheit ist, liegt ein␣
↪Fehler vor.
        print("Fehler: Unbekannte Skala: " + skala_nach)
else:
    # Wenn die Ausgangsskala weder Celsius noch Fahrenheit ist, liegt ein␣
↪Fehler vor.
    print("Fehler: Unbekannte Skala: " + skala_von)
```

```
Willkommen beim Beispieltemperaturkonverter!

Von welcher Skala wollen Sie konvertieren? Celsius
```

Ihre Beschreibung und Erläuterungen hier! (Doppelklick zum Editieren, Shift + Enter um die Zelle auszuführen)