

Variablen, Funktionen, Kontrollfluss

1 Variablen

- Starten Sie den Python Interpreter, und machen Sie sich mit ihm vertraut
- Speichern Sie ihren Namen in einer Variable. Achten Sie darauf sinnvolle (vorzugsweise englische) Variablennamen zu verwenden!
- Speichern Sie ihr Alter in einer Variablen.
- Geben Sie folgenden Text auf der Konsole aus: „-Name- ist -Alter- Jahre alt“. Lesen Sie die Werte -Name- und -Alter- aus den entsprechenden Variablen aus, die Sie bereits angelegt haben.
- Entscheiden Sie für die folgenden Bezeichner, ob Sie gültige Variablennamen in Python sind: size, circle_radius, test-variable, testVariable, correct?, attention!, for
- Sehen Sie sich folgendes Code Beispiel an:

```
a = 42
b = '42'
c = 42.0
```

Wo liegt der Unterschied zwischen den Variablen a, b und c? Welche Typen haben Sie? Überlegen Sie sich zuerst ihre Antwort, und überprüfen Sie sie dann mittels type().

- Speichern Sie den Radius eines Kreises in einer Variablen. Berechnen Sie Umfang und Flächeninhalt dieses Kreises und geben Sie beides auf der Konsole aus. *Tipp*: Sie brauchen dazu natürlich π – Sie finden mehr darüber in der Dokumentation des Math Moduls (<http://docs.python.org/2/library/math.html>)!

2 Funktionen

- Schreiben Sie eine Funktion incr(x), die $x + 1$ zurückgibt.
- Implementieren Sie eine Funktion say(text), die den übergebenen Text auf der Konsole ausgibt. Der Parameter text ist optional, sollte er nicht übergeben werden gibt say „Hello Kitty!“ aus.
- Implementieren Sie eine Funktion leapYear(year), die berechnet ob es sich beim gegebenen Jahr um ein Schaltjahr handelt. Geben Sie entsprechend True oder False zurück. *Tipp*: Ein Schaltjahr ist restlos durch 4 teilbar, jedoch ist ein volles Jahrhundert nur dann ein Schaltjahr, wenn es auch durch 400 teilbar ist. Beispiel: 1900 ist kein Schaltjahr, 2000 aber schon.

3 Kontrollfluss

- Schreiben Sie eine Funktion cmp(a, b), die zwei gegebene Zahlen a und b vergleicht. Ist $a < b$, gibt die Funktion -1 aus. Ist $a > b$, gibt die Funktion 1 aus. Sind beide Zahlen gleich groß, gibt die Funktion 0 aus.
- Geben Sie die Zahlen von 25 bis 1 rückwärts aus (25, 24, ..., 1).
- Geben Sie die ersten n Fibonacci Zahlen aus. Es gilt $F_n = F_{n-1} + F_{n-2}$, $F_1 = 1$ und $F_0 = 0$

4 Zusatzübungen

Sollten Sie bereits mit allen Übungen fertig sein, können Sie versuchen folgende Zusatzprobleme zu lösen:

- Programmieren Sie folgendes kleines Spiel: Der Computer „denkt“ sich eine geheime Zahl aus. Der Benutzer muss versuchen, diese Zahl in mehreren Versuchen zu erraten. Nach jedem Versuch gibt das Spiel aus, ob der Benutzer zu niedrig, zu hoch, oder richtig geraten hat. Die maximale Anzahl der Versuche, und den Wertebereich der Zufallszahlen, können vom Benutzer gewählt werden. Das Spiel ist beendet, wenn die maximale Anzahl an Versuchen überschritten wurde, oder aber der Spieler die geheime Zahl errät. *Tipp:* Sie können mit `userInput = int(raw_input())` eine Zahl von der Kommandozeile einlesen. Verwenden Sie das Modul `random` um Zufallszahlen zu erzeugen (<http://docs.python.org/2/library/random.html>).