Variablen, Funktionen, Kontrollfluss

1 Variablen

- Starten Sie den Python Interpreter, und machen Sie sich mit ihm vertraut
- Speichern Sie ihren Namen in einer Variable. Worauf sollten Sie bei der Benennung von Variablen achten?
- 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: 1a, size, circle_radius, test-variable, testVariable, correct?, attention!, for, xqwoi Sind diese Bezeichner auch sinnvoll?
- 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 Kontrollfluss

- Geben Sie die Zahlen von 25 bis 1 rückwärts aus (25, 24, ..., 1).
- Geben Sie alle Zahlen von 1 bis inklusive 100 aus, die entweder durch 3 oder durch 7 teilbar sind.
- Geben Sie alle Zahlen von 1 bis inklusive 100 aus die durch 8, aber nicht durch 24, teilbar sind.
- Geben Sie die ersten n Fibonacci Zahlen aus. Es gilt $F_n = F_{n-1} + F_{n-2}$, $F_1 = 1$ und $F_0 = 0$. Die Variable n wird vorher beliebig gesetzt (Beispiel: n = 42).
- Geben Sie alle Vielfachen der Zahlen 2 bis inklusive 10 aus, die kleiner als 100 sind. Beginnen Sie mit den Vielfachen von 2: 2, 4, ..., 98. Dann kommen alle Vielfachen von 3 (3, 6, ..., 99), und so weiter.

3 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.
- 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.
- Betrachten Sie die beiden folgenden Codesschnipsel, die die selbe Ausgabe erzeugen. Welche Vor- oder Nachteile könnte die eine oder andere Version haben?

```
>>> def function(a):
...     return a * 2
...
>>> print function(4)
8
```

```
>>> def function(a):
...     print a * 2
...
>>> function(4)
8
```

- 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 furch 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. Würden Sie in einem Projekt auf eine existierende Lösung zurückgreifen (Library), oder die Funktion wie in dieser Übung implementieren?

4 Zusatzübungen

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

- Programmieren Sie das folgende kleine 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 nieder, 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`).
- (Achtung schwer!) Überlegen Sie sich, wie Sie mit Zufallszahlen die Zahl Pi approximieren können. Tipp: Stellen Sie sich vor Sie haben ein Quadrat, das einen Kreis umschließt. Erzeugen Sie dann eine große Zahl zufälliger Punkte innerhalb dieses Quadrates und berechnen Sie die Wahrscheinlichkeit, dass ein Punkt innerhalb des Kreises liegt.