

EINFÜHRUNG IN DIE PROGRAMMIERUNG MIT PYTHON

Fingerübungen

Woche 04 – 1. Juni 2016

Selbst definierte Funktionen

... mit Parametern; Strukturierung u. Vermeidung von Redundanz

(1) Maximum

Schreiben Sie eine Funktion `max(a, b, c)`, die das Maximum von drei Zahlen ermittelt und auf der Konsole (= Terminal/*Python Shell*) ausgibt (mittels `print(...)`). Die Zahlen werden als *Parameter* (a, b & c) übergeben.

(2) ASCII-Zeichenkunst: Rechtecke beliebiger Größe

Schreiben Sie eine Funktion `printRectangle(width, height)`, die ein Rechteck aus Sternen der Breite `width` und Höhe `height` auf der Konsole ausgibt. Die Ausgaben folgender Aufrufe wären bspw.:

<code>printRectangle(3, 5)</code>	<code>printRectangle(7, 3)</code>	<code>printRectangle(4, 4)</code>
*** *** *** *** ***	***** ***** *****	**** **** **** ****

(3) [*Zusatzaufgabe*] Versuchen Sie sich auch an **Dreiecken** beliebiger Größe. Beispiele hierzu auf dem (alten) Blatt zum „PRAXISTEIL“ aus Woche 03.

Selbst definierte Funktionen

... mit Rückgabewerten; Strukturierung u. Vermeidung von Redundanz

(4) Boolesche Zufallsfunktion

Schreiben Sie eine Funktion namens `randbool`, die einen zufälligen booleschen Wert (also *True* oder *False*) zurückgibt!

Tipp: Nutzen Sie hierzu die (bereits gegebene) Funktion `randint(a, b)` aus dem Modul `random`, welche eine zufällige Zahl zwischen `a` und `b` zurückgibt (jeweils inklusive).

Erinnerung: Um auf Funktionen aus einem Modul zuzugreifen, müssen Sie dieses zunächst importieren (`import random`). Der anschließende Aufruf wäre bspw. `random.randint(0, 1)`. Tipp: `bool()`

(5) Produkt

Schreiben Sie eine Funktion `product(a, b, c)`, die das Produkt der drei (als *Parameter* übergebenen) Gleitkommazahlen (*float*) berechnet und als Ganzzahl (*int*) zurückgibt. Sie können das Ergebnis dazu mit der Funktion `round(...)` mathematisch korrekt auf eine ganze Zahl (*int*) runden.

Selbst definierte Funktionen

Strukturierung und Vermeidung von Redundanz, *Zusatzaufgaben*

(6) [*Zusatzaufgabe*] Primzahl-Prüfer

Tipp: Mit Modulo-Operator (%) Rest auf 0 prüfen.

Schreiben Sie eine Funktion, die prüft, ob eine (als Parameter übergebene) Zahl eine Primzahl ist (\rightarrow `return True`) oder nicht (\rightarrow `return False`).

Hinweis: Eine Primzahl ist eine natürliche Zahl größer als 1, die keine (positiven) Teiler außer 1 und sich selbst (also 2, 3, 5, 7, 11, 13, ...) hat.

(7) [*Zusatzaufgabe*] Fakultät (*rekursiv*)

Aufgabe (7) wird **nicht** besprochen. Lösungen auf Moodle.

Def.: *Rekursiv* wird eine Funktion bezeichnet, die sich selbst aufruft. Eine rekursive Funktion benötigt in jedem Fall eine Abbruchbedingung (*Anker*), d. h. ein Ende (sonst: endlos viele rekursive Aufrufe \rightarrow `RecursionError`).

Ihre Aufgabe ist es nun, sich eine solche rekursive Funktion zu überlegen, die die Fakultät einer (als Parameter übergebenen) Zahl berechnet. Der Gedanke dabei ist der folgende (am Beispiel der Fakultät (*fak*) von 4):

$$\begin{array}{l} \text{wobei} \quad fak(4) = 4 * 3 * 2 * 1 = 24 = 4 * fak(3) \\ \text{wobei} \quad fak(3) = 3 * 2 * 1 = 6 = 3 * fak(2) \\ \text{wobei} \quad fak(2) = 2 * 1 = 2 = 2 * fak(1) \\ \text{und} \quad fak(1) = 1 \end{array} \left. \vphantom{\begin{array}{l} fak(4) \\ fak(3) \\ fak(2) \\ fak(1) \end{array}} \right\} \begin{array}{l} \text{rekursive Aufrufe} \\ \text{Rekursionsanker (Abbruch)} \end{array}$$

Selbst definierte Funktionen

... mit Rückgabewerten; weitere Zusatzaufgaben (alt)

(8) [*Zusatzaufgabe*] Zufälliger Buchstabe

Schreiben Sie eine Funktion namens **randchar**, die einen zufälligen Buchstaben zurückgibt (im Bereich A-Z & a-z)! Sie können hierzu z. B. `randint()` (siehe (4)) und die Funktion `chr(i)`, welche das zu einem ASCII-Code (*i*) gehörende Zeichen zurückgibt, verwenden.

ASCII-Code	Zeichen
65, 66, ..., 90	A, B, ..., Z
97, 98, ..., 122	a, b, ..., z

Selbst definierte Funktionen

... mit Rückgabewerten; weitere Zusatzaufgaben (alt)

(9) [*Zusatzaufgabe*] Zufällige Bestätigungscode (Szenario)

Hintergrund: In einem Webshop sollen Personen (Käufer) Artikel bestellen können, ohne sich extra registrieren zu müssen. Für eventuelle Rückfragen oder Änderungen muss der/die Käufer/in jedoch seine/ihre E-Mail-Adresse hinterlegen. Der Webshop möchte sich gegen falsche Eingaben absichern, indem die eingegebene Mail-Adresse vor dem Kauf verifiziert werden muss. Dazu wird ein Bestätigungscode generiert und per E-Mail an die entsprechende Adresse versandt. Der Käufer muss diesen Code anschließend online eingeben, bevor die Bestellung aufgegeben wird.

Ihre Aufgabe ist es nun, eine Funktion zu schreiben, die einen solchen Bestätigungscode **zufällig** generiert (und per `return` zurückgibt). Der Code soll dabei aus insgesamt **fünf Zeichen** bestehen. Die Zeichen können Zahlen, Klein- oder Großbuchstaben (keine Sonderzeichen o. ä.!) sein. Nutzen Sie (falls nötig) die Funktionen aus den Aufgaben (4) und (5) zu Rückgabewerten.

Beispielcodes: da5G1, ukwwb, HG62D, 651Px, 02912