

Repetition1

Dr. Günter Kolousek

15. November 2015

Lege wiederum ein Verzeichnis an. Nennes es 08_repetition1! In diesem Verzeichnis sollen alle Dateien der jeweiligen Einheit abgelegt werden.

1 Wiederholung 1

1. Berechne: $11_2 + 22_3 \cdot 33_4 - 44_5$
2. Berechne: $(11_2 + 22_3)(33_4 - 44_5)$
3. Berechne das 12-fache der Summe von 345 und 678.
4. Berechne die Differenz von 5 und der Differenz von 4 und 3.
5. Berechne alle Zweierpotenzen bis zur Hochzahl 10 (also von 0 bis 10) und lerne diese auswendig.
6. Lerne alle Hexadezimalziffern auswendig.
7. Berechne: 2^{2^3}
8. Berechne: $(2^2)^3$ Was fällt dir im Vergleich zur vorhergehenden Angabe auf?
9. Berechne die Geschwindigkeit eines Motorrades

Ein Motorrad legt innerhalb von 2 Stunden und 22 Minuten 252km zurück. Es liegt eine gleichförmige Bewegung zugrunde, d.h. das Motorrad beschleunigt nicht (und bremst daher auch nicht). Berechne die Geschwindigkeit.

10. Berechne die zurückgelegte Strecke eines Lastwagens

Ein Lastwagen fährt mit einer durchschnittlichen Geschwindigkeit von 95km/h. Welche Strecke legt dieser Lastwagen in 2 Stunden und 22 Minuten und 22 Sekunden zurück?

11. Schreibe das Programm `salary1.py`, das den Benutzer nach einer Anzahl von Jahren der Firmenzugehörigkeit fragt und danach das Gehalt in diesem Jahr ausgibt.

Zugrundeliegend ist, dass das Anfangsgehalt 1800€ beträt und eine jährliche Steigerung von 20€ vereinbart ist.

D.h. es handelt sich bei unserem Gehaltsmodell um eine lineare Funktion. Im Programm sollen die Werte `x`, `k` und `d` am Anfang als Variable definiert werden und danach das Ergebnis ausgegeben.
12. Schreibe ein Program `salary2.py`, das den Benutzer nach einem Zielgehalt fragt und dem Benutzer ausrechnet nach wievielen Jahren er dieses erreichen wird.

Wir gehen wiederum von 1800€ Anfangsgehalt und einer jährlichen Steigerung von 20€ aus.
13. Berechne den Nettogehalt

Mit Ihrem Arbeitgeber wurde ein Gehalt vereinbart. Leider wird dieser Betrag nicht auf das Bankkonto überwiesen, da sowohl Sozialversicherungsbeiträge (Pension, Krankenversicherung,...) abgezogen werden als auch die Einkommenssteuer an das Finanzamt zu entrichten ist.

Nehmen wir an, dass mit dem Arbeitgeber ein Bruttogehalt von 1800€ vereinbart haben. Um es einfach zu halten, gehen wir davon aus, dass die Sozialversicherungsbeiträge einen Umfang von 20% ausmachen. D.h. zuerst müssen diese von 1800€ abgezogen werden. Vom verbliebenen Betrag muss noch die Einkommenssteuer abgezogen werden. Wiederum der Einfachheit halber gehen wir davon aus, dass diese ein Drittel von dem schon verminderten Betrag ausmachen.
14. Schreibe ein Programm `umfang.py`, das $4 * 3.1415 =$ und anschließend das Ergebnis ausgibt.

Diese Formel ist doch ähnlich der Formel zur Berechnung des Umfang eines Kreises, nicht wahr? Welchen Radius hat der Kreis?

Baue in einem weiteren Schritt das Programm so um, dass die Konstante `pi` aus dem Modul `math` verwendet wird!
15. Schreibe ein Programm `countdown.py`, dass alle Zahlen von 10 bis 0 (jeweils in einer eigenen Zeile) auf der Konsole ausgibt. Verwende eine Schleife!
16. Schreibe ein Programm `hello.py`, das Er sagte: "Chez l'ami Roger" ausgibt.
17. Schreibe ein Programm, das den Benutzer nach einer Seitenlänge eines Rechteckes fragt, danach nach dem Umfang dieses Rechteckes. In weiterer Folge soll die verbleibende Seitenlänge und der Flächeninhalt ausgegeben werden.
18. Wann ist eine binäre Zahl ungerade? Wann gerade? Warum?

19. Schreibe ein Programm `triangle.py`, das den Benutzer nach einer Kathete und der Hypotenuse fragt und danach die andere Kathete, den Umfang und die Fläche ausgibt.
20. Schreibe ein Programm `quader.py`, das den Benutzer nach einer Raumdiagonale sowie zwei Seiten fragt und danach die dritte Seite und auch das Volumen berechnet und ausgibt.
21. Schreibe ein Programm `kmh2ms.py`, das nach einer Geschwindigkeit in km/h fragt, diese in m/s umrechnet und danach ausgibt.

Das sollte so aussehen::

```
Bitte die Geschwindigkeit in km/h eingeben: 100
100km/h = 27.7777777778
```

22. Programm `linfunc.py`: Bestimme alle Ergebnisse der linearen Funktion $f_1(x) = 2x$ für die Zahlen von 0 bis 10.
 - a) Definiere eine Python-Funktion `f1(x)`, die den Funktionswert berechnet und zurückliefert.
 - b) Schreibe das Hauptprogramm so, dass dieses mit einer `for`-Schleife die Funktion entsprechend aufruft und die Ergebnisse ausgibt.

Zeichne das Schaubild der Funktion (am Papier)!

23. Erweitere das Programm `linfunc.py`:
 - a) Definiere eine weitere Funktion `f2(x)`, die die mathematische Funktion $f_2(x) = 3x$ berechnet und zurückliefert.
 - b) Schreibe das Hauptprogramm so um, dass der Benutzer gefragt wird, ob er die Lösung mit der Variante A oder mit der Variante B sehen will. Das könnte so aussehen:


```
(A) Funktion f1 für die Zahlen von 0 bis 10
(B) Funktion f2 für die Zahlen von -5 bis 5
```

Gibt der Benutzer weder A noch B ein, dann beendet sich das Programm.
 - c) Gibt der Benutzer A ein, dann soll das Programm wie im vorhergehenden Beispiel funktionieren. D.h. die Funktion `f1` für die Zahlen von 0 bis 10 berechnen und ausgeben.
 - d) Gibt der Benutzer B ein, dann soll das Programm die Funktionswerte der Funktion `f2` für die Zahlen von -5 bis 5 berechnen und ausgeben.

Zeichne das Schaubild der Funktion (am Papier)!

- e) Erweitere das Programm weiters um eine Variante **C**, für die eine Funktion $f_3(x)$ definiert. Diese Funktion ist folgendermaßen definiert: $f_3(x) = 2x + 1$. Dafür sollen die Werte von -7 bis +7 ausgegeben werden.

Zeichne das Schaubild der Funktion (am Papier)!

Baue das Programm in das Menü ein!

- f) Verändere jetzt das Programm so, dass auch Kleinbuchstaben für die Menüauswahl zulässig sind.

24. Schreibe ein Programm `minimum.py`, das 4 Zahlen abfragt und die kleinste Zahl ausgibt. Das Programm soll ausschließlich `if`, `elif` und `else` Anweisungen verwenden.

Schreibe das Programm in weiterer Folge so um, dass die Bestimmung des Minimums in eine Funktion `minimum` verlagert wird. Verwende dazu Defaultwerte. 0 wäre dafür nicht schlecht, oder?

25. Schreibe ein Programm `diagonal1.py`, das

- a) eine Funktion `quadrat(length)` enthält, das ein Quadrat an der gegebenen Position mit der Seitenlänge `length` mit der Turtle zeichnet. Dies soll mit dem Uhrzeigersinn durchgeführt werden.
- b) eine Funktion `diagonal(length)` enthält, das eine Diagonale der Länge von `length` Quadraten ausgehend von Koordinatenursprung nach rechts oben (also im Winkel von 45 Grad) zeichnet. Dazu soll die Funktion `quadrat` jeweils mit einer Seitenlänge von 30 aufgerufen werden.
- c) Der Benutzer soll nach einer Länge der Diagonale gefragt werden und die entsprechende Zeichnung angefertigt werden.

26. Schreibe ein Programm `diagonal2.py`, das es erlaubt auch gefärbte Diagonalen (wie im vorhergehenden Beispiel) zu zeichnen.

Gehe folgendermaßen vor:

- a) Schreibe ein Modul `diagonallib`, das vorerst die Funktion `Quadrat` enthält. Diese soll prinzipiell wieder wie im vorhergehenden Beispiel funktionieren, allerdings mit der Erweiterung, dass nicht nur die Länge sondern auch die Stiftfarbe mitgegeben werden kann.

Die Funktion soll also folgende Signatur haben::

`quadrat(length, pencolor)`

Teste die Funktion in der IPI!

- b) Schreibe die Funktion `quadrat` so um, dass als Defaultwert für die Stiftfarbe schwarz verwendet wird.

- c) Schreibe eine Funktion `filled_quadrat` im gleichen Modul, die als Parameter die Seitenlänge, die Stiftfarbe und die Füllfarbe bekommt und zum Füllen die Funktion `quadrat` verwendet.

Lege als Defaultwert für die Stiftfarbe schwarz und für das Füllen die Farbe rot fest.
 - d) Schreibe weiters eine Funktion `diagonal`, die im Prinzip wie im vorhergehenden Beispiel funktioniert, aber eben die Funktion `filled_quadrat` verwendet und ebenfalls Parameter mit Default-Werten für die Stiftfarbe und die Füllfarbe hat.
 - e) Schreibe zum Abschluss das Hauptprogramm `diagonal2.py`, das den Benutzer nach einer Länge, einer Stiftfarbe und einer Füllfarbe fragt und danach die Diagonale zeichnet!
27. Schreibe nun ein Programm `mod.py`. In diesem Programm definiere eine Funktion `mod`, die den Rest der beiden übergebenen Zahlen ermittelt und zurückliefert.
- Allerdings darf weder der `%` noch der `//` Operator verwendet werden. `int()` darf schon verwendet werden.
- Der Benutzer soll nach zwei Zahlen gefragt werden und der Rest der Division soll ausgegeben werden.
28. Schreibe nun eine Funktion `prod` in ein Modul `prod.py`, der bis zu 5 Parameter übergeben werden können und die das Produkt der Zahlen zurückliefert. Verwende dazu Parameter mit Default-Werten (d.h. Standard-Werten). 1 wäre als Defaultwert nicht so schlecht, oder?
- Teste die Funktion im interaktiven Interpreter (d.h. in der Python-Shell).