**Globale Verständnisfragen**

Zum Abschluss dieses Lernmoduls wollen wir noch einmal die wichtigsten Erkenntnisse und Informationen, die Sie aus diesem Lernmodul mitnehmen sollen, mit einigen zusammenfassenden Fragen verschiedenen Typs abfragen und wiederholen. Sie können diese Fragen als "inverted classroom" verstehen, und damit als komplementär zu den vorgeschlagenen größeren Übungen im nächsten Abschnitt. Die Idee ist, dass Sie sich hier überlegen können, ob Sie die gestellten Fragen selbst beantworten können. Falls es noch Unklarheiten gibt, können Sie die entsprechenden Antworten nochmals direkt in den passenden Abschnitten des Lernmoduls rekapitulieren. Dementsprechend sind die Antworten auch absichtlich skizzenhaft. Bitte zögern Sie nicht, bei konkreten Fragen einen der Kanäle zur Hilfe zur Selbsthilfe zu nutzen.

**Vokabeltest**

 **Was sind die Komponenten der von Neumann Architektur?**

 Speicher, Bus und Prozessor, wobei letztere aufgeteilt ist in eine Kontrolleinheit und eine Recheneinheit, s. Abschnitt Computerarchitektur. Wichtig ist, dass Sie verstanden haben, dass die von Neumann Architektur ein mentales Modell ist, und von konkreten Realisierungen in der Hardware komplett abstrahiert.

 **Was sind Quelltext-Kommentare?**

 Quelltext-Kommentare dienen der Dokumentation von Programmen, siehe diesen Abschnitt für Details. In Python können wir mit # ganze Zeilen oder den Rest einer Zeile auskommentieren, so dass dieser auskommentierte Teil unseres Codes schlicht und einfach ignoriert wird.

 **Was ist eine Variable?**

 Eine Variable ist im Programm ein Platzhalter für einen konkreten Wert. Eine Variable hat immer einen konkreten Namen, und bekommt einen konkreten Wert zugewiesen. Im von Neumann Modell (und in Python) kann eine Variable verstanden werden als ein "Zeiger" auf eine konkrete Stelle im Speicher, an dem der Wert der Variable abgelegt ist. Details finden sich im Abschnitt über Variablen.

 **Was ist eine Zuweisung?**

 Eine Zuweisung setzt eine Variable auf einen von uns vorgegebenen Wert. Konkret bedeutet dies, dass im Speicher unter dem Namen der Variable dann der konkrete Wert abgelegt ist. Mehr Informationen finden Sie im Abschnitt über Variablen und Zuweisungen.

 **Wie kann man eine Variable einlesen über die Tastatur, zur Laufzeit des Programms?**

 Hierfür gibt es den Befehl input().

 **Wofür sind while Schleifen da?**

 While Schleifen ermöglichen die wiederholte Ausführung von Teilen unserer Programme. Die Art der Wiederholung ist dabei flexibel, und wird in der Regel durch eine Variable gesteuert. Eine Einführung in diese Art von Schleifen finden Sie im entsprechenden Abschnitt.

 **Was ist eine Endlosschleife?**

 Eine Endlosschleife ist eine Schleife, die nicht terminiert: Die Aktualisierung der Schleifenvariable stellt nie sicher, dass die Abbruchbedingung der Schleife greift.

 **Was ist ein Block in Python?**

 Ein Block ist ein zentrales Strukturierungs-Hilfsmittel für Python-Programme. Ein Block ist definiert als eine um die gleiche Anzahl Leerzeichen eingerückte Sequenz von Codezeilen. In Verbindung mit Kontrollstrukturen wie der while Schleife definieren Sie, welche Teile des Codes Bestandteil der Schleife sind, d.h. wiederholt ausgeführt werden. Details finden Sie hier.

**Syntaxfragen**

 **Wie formatieren Sie die Ausgabe einer ganzen Zahl rechtsbündig auf 5 Spalten Breite?**

 i=42

 print("{:5}".format(i))

 **Gegeben drei beliebige Zeichenketten mit jeweils Maximallänge 10. Wie geben Sie diese in einer Zeile folgendermaßen aus?** "s1 rechtsbündig | s2 zentriert | s3 linksbündig"

 print("{:<10}{:^10}{:>10}".format(s1,s2,s3))

 **Was ist der allgemeine Syntax einer while Schleife?**

 while condition:

 block

 **Was bewirkt die Multiplikation einer Zeichenkette mit einer Zahl?**

 Die Zeichenkette wird wiederholt, und zwar so oft, wie es die Zahl vorgibt.

 **Was bewirkt die Addition zweier Zeichenketten?**

 Die Zeichenketten werden konkateniert.

**Einfache Verständnisfragen**

 **Übersetzen Sie die Anweisungskette a = 1; b = 2; c = a\*b in die Sprechweise der von Neumann Abstraktion. Benutzen Sie insbesondere die Begriffe Speicher, Bus, ALU, Control Unit.**

 Das sollte gemäß des entsprechenden Abschnitts klar sein.

 **Warum ist dieser Code erlaubt?** a = 7; print(a); a = "sieben"; print(a)

 In Python sind Variablen dynamisch typisiert, d.h. ihre Namen (hier a) dürfen unterwegs den Typ (hier Zahl und Zeichenkette) ändern. Empfehlenswert ist das allerdings nicht.

 **Wie lesen Sie eine ganze Zahl über die Tastatur ein zur Laufzeit?**

 var = input("enter something")

 Wichtig ist die Typkonvertierung: var = int(var)

 **Wie berechnen Sie den Tangens von pi/4? Wie von pi/2?**

 Hier benötigen wir das Paket math.

 math.tan(math.pi/4), math.tan(math.pi/2)

 Details finden sich im Abschnitt über elementares Rechnen

 **Schreiben Sie eine while Schleife, die die Zweierpotenzen bis 2^20 ausgibt.**

 i=1

 while i<20:

 print(2\*\*i)

 i = i+1

**Transferfragen**

 **Wie lesen Sie mit einem input Befehl zwei Zahlen gleichzeitig ein?**

 Eine kurze Recherche mit der Suchmachinen-Abfrage "python input two values" liefert bspw. diese Antwort. Offensichtlich verstehen wir sie nicht. Trotzdem ist diese Frage didaktisch sehr sinnvoll: Wir gewinnen die Erkenntnis, dass diese Funktionalität realisierbar ist, aber eben noch nicht mit unseren recht rudimentären Kenntnissen zum aktuellen Zeitpunkt des Kurses. Später werden wir ohne weiteres Nachdenken die beiden vorgeschlagenen Lösungsmöglichkeiten einsetzen können. Bis dahin gilt, dass wir zwei Werte über zwei input() Befehle einlesen müssen.

 **Schreiben Sie eine while Schleife, die nur über die Quadratzahlen von 1 bis 100 iteriert.**

 i = 1

 while i\*\*2 <= 100:

 print(i\*\*2)

 **Schreiben Sie eine while Schleife, die die geometrische Summe für k Summanden berechnet.**

 k = 10

 i = 0

 sum = 0

 while i<=k:

 sum = sum + k\*\*i

 i = i+1

 **Schreiben Sie eine while Schleife, die die Summe der ersten k Quadratzahlen berechnet.**

 k = 10

 i = 0

 sum = 0

 while i<k:

 sum = sum + i\*\*2

 i = i+1

 **Überprüfen Sie die Korrektheit der letzten beiden Programme mit der entsprechenden Formel.** Hinweis: https://de.wikipedia.org/wiki/Formelsammlung\_Arithmetik#Summenformeln

 Das ist einfach, wir müssen nur die Formeln abtippen.