Beispiele für Trace-Tabellen

Gehen Sie diese while-Schleifen durch.

**While-Schleife 1**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | from time import sleep  count = 5  while count != 0:  print(count)  count = count - 1  sleep(1) |

**Vervollständigen Sie die Tabelle:**

|  |  |  |  |
| --- | --- | --- | --- |
| Leitung | Variable count | Bedingung count!=0 | Ausgabe = print |
| 2 | 5 |  |  |
| 3 |  | True |  |
| 4 |  |  | 5 |
| 5 | 4 |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |

**While-Schleife 2**

|  |  |
| --- | --- |
| 1  2  3  4 | number = 2  while number < 11:  print(number)  number = number + 2 |

**Vervollständigen Sie die Tabelle:**

|  |  |  |  |
| --- | --- | --- | --- |
| Leitung | Variable number | Bedingung: number < 11 | Ausgabe |
| 1 | 2 |  |  |
| 2 |  | True |  |
| 3 |  |  | 2 |
| 4 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 2 |  |  |  |

**While-Schleife 3**

Basierend auf Euklids Algorithmus zur Berechnung des größten gemeinsamen Faktors von zwei ganzen Zahlen.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | a = 21  b = 15  while a != b:  if a < b:  b = b-a  else:  a = a-b  print(a) |

**Vervollständigen Sie die Tabelle:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Variable | | Zustand/Bedingung | |  |
| Leitung | a | b | a != b | a < b | Ausgabe |
| 1 | 21 |  |  |  |  |
| 2 |  | 15 |  |  |  |
| 4 |  |  | True |  |  |
| 5 |  |  |  |  |  |
| 8 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 8 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 10 |  |  |  |  |  |

Erkenne und korrigiere den Fehler.

**Eine while-Schleife mit einem Fehler**

Diese while-Schleife wurde so konzipiert, dass sie die 5-er Reihe im Einmaleins bis zu 25 ausgibt (5,10,15,20,25). Verwenden Sie eine Trace-Tabelle, um zu sehen, wie der Code ausgeführt wird. Wenn Sie den Fehler entdeckt haben, schreiben Sie den Code in dem dafür vorgesehenen Feld auf der nächsten Seite um.

|  |  |
| --- | --- |
| 1  2  3  4  5 | number = 1  while number != 25:  number = number + 5  print(number) |

**Vervollständigen Sie die Tabelle:**

|  |  |  |  |
| --- | --- | --- | --- |
| Leitung | Variable number | Bedingung number != 25 | Ausgabe |
| 1 | 1 |  |  |
| 3 |  | Ture |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |

**Korrigieren Sie den Code und schreiben Sie ihn in das unten stehende Feld:**

|  |  |
| --- | --- |
| 1  2  3  4  5 |  |

**Überprüfen Sie, ob Sie richtig liegen, indem Sie die Tabelle ausfüllen:**

|  |  |  |  |
| --- | --- | --- | --- |
| Zeile | Variable number | Bedingung numbmer != 25 | Ausgabe |
| 1 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 3 |  |  |  |

Explorer-Aufgabe .

Dieses Programm zeigt die Primfaktoren für eine Zahl an. Es prüft, ob die Zahl perfekt durch den Faktor teilbar ist. Alternativ dazu prüft es, ob der Faktor die Zahl teilt.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | zahl = 15  faktor = 2  while zahl > 1:  if zahl % faktor == 0:  print(faktor)  zahl = zahl // faktor  else:  faktor = faktor + 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Variable | | Zustand | |  |
| Zeile | zahl | faktor | zahl > 1 | zahl % faktor == 0 | Ausgabe |
| 1 | 15 |  |  |  |  |
| 2 |  | 2 |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 9 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 9 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 9 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| 4 |  |  |  |  |  |

Die Ressourcen werden regelmäßig aktualisiert - die neueste Version finden Sie unter: [the-cc.io/curriculum](http://the-cc.io/curriculum).



Diese Ressource wird von der [Raspberry Pi Foundation](https://www.raspberrypi.org/) unter einer Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International-Lizenz lizenziert. Um eine Kopie dieser Lizenz zu sehen, besuchen Sie .[creativecommons.org/licenses/by-nc-sa/4.0/](https://creativecommons.org/licenses/by-nc-sa/4.0/)