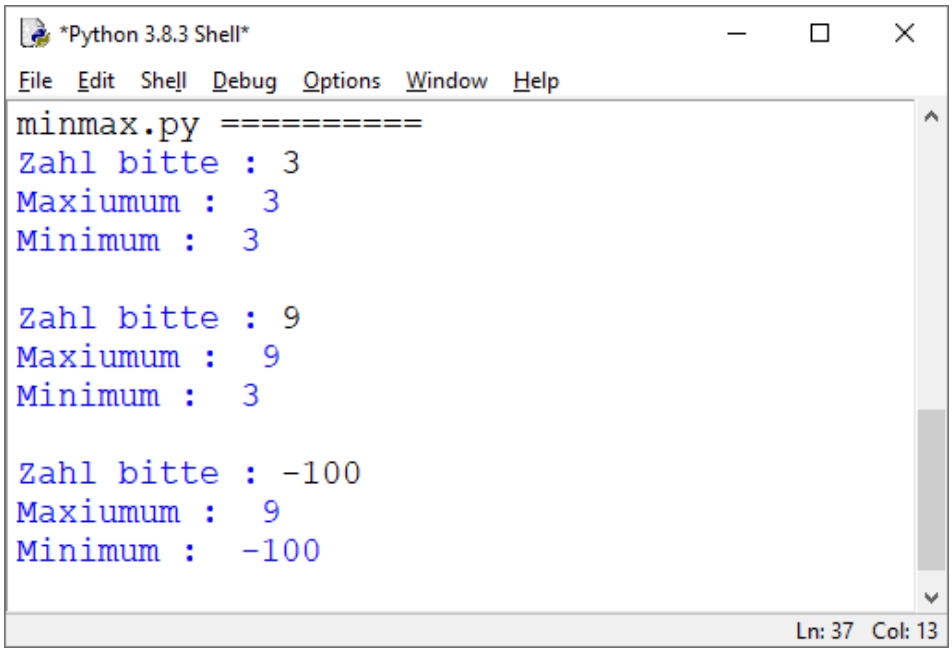
Min-Max

Es soll eine beliebige Anzahl von Zahlen eingelesen und fortlaufend geprüft werden. Es wird dabei das Maximum und das Minimum der eingegebenen Zahlen ermittelt, und ausgegeben. Das Programm läuft ohne Ende.

Beispiel :



Lösungsvorschlag

Praxis

Das Ganze soll endlos laufen.

Also eine Schleife, und zwar mit ner Bedingung, die immer “True” ist.

Man könnte a=1 setzen, und dann “while a == 1 : ” prüfen.

Einfacher ist es, man schreibt das Ergebnis der Prüfung geleich direkt hin : “ while True : ”

Der Rest ist simpel, natürlich werden die Über- und Unterschreitung des aktuellen Maximums oder Minimums mit if-Konstrukten geprüft :

```
min = 100000000
max = -100000000
while True :
   a = int(input("Zahl bitte : "))
   if a < min :
      min=a
   if a > max :
      max=a
   print("Maxiumum : ",max)
   print("Minimum : ",min)
   print("")
```

Durchschnitt

Es soll eine beliebige Anzahl von Zahlen eingelesen, und forlaufend der Durchschnitt berechnet werden. Bei Eingabe von “stop” endet das Programm.

Also z.b. :
1 rein -> Durchschnitt = 1
4 rein -> Durchschnitt = 2.5
4 rein -> Durchschnitt = 3
usw.

Wenn die Zahlen Meßwerte wären, hätte man damit schon einen einfachen digitalen Filter, ähnlich einem Tiefpaß.

Praxis

Das Ganze soll laufen, bis “stop” eingegeben wird.

Das zwickt sich ein bisschen mit der Rechnung, für die man ein Zahlenformat braucht (ich habe Integer gewählt).

Ohne besondere Maßnahmen läuft das zwar mit der Abfrage auf “stop”, macht aber eine sehr unschöne Fehlermeldung :

```
sum=0
zahl=0
rein=0
while rein != "stop" :
   rein = input("Zahl bitte : ")
   b = int(rein)
   sum =  sum + b
   zahl = zahl + 1
   durch = sum / zahl
   print("Anzahl : ",zahl, "Summe : ",sum ," Schnitt : ",durch)
   print("")
```

Beheben läßt sich das durch das sehr praktische “try/except”:

Wenn im Ablauf Probleme erwartet werden, “probiert” man die kritische Operation quasi aus (= try), und falls es knirscht (= except), reagiert man kontrolliert drauf. (Fachjargon : “das Programm wirft eine Exception”).

```
sum=0
zahl=0
rein=0
while rein != "stop" :
   rein = input("Zahl bitte : ")
   zahl = zahl + 1
   try :
      b = int(rein)
      sum =  sum + b
      durch = sum / zahl
   except :
      print("das wars !")

   print("Anzahl : ",zahl, "Summe : ",sum ," Schnitt : ",durch)
   print("")
```

Summe

Sie sollen zwei Zahlen einlesen, sagen wir n und m.

Errechnen sie die Summe aller Zahlen zwischen n und m,

die beiden eingeschlossen.

n=9, m=14 -> Summe = 69

n=200, m=350 -> Summe = 41525

Lösungsvorschlag

Dazu gibts wenig zu sagen, klar muß es eine Schleife sein ...

```
a=int(input("erste zahl : "))
b=int(input("zweite zahl : "))
summe=0
while a < b+1 :
   summe=summe+a
   a=a+1
print("Summe : ",summe)
```

Denksport

Praxis

Sie sollen zwei Zahlen tauschen.

Ablauf : Sie lesen zwei Zahlen ein, also z.b. a=3 und b=5

Und jetzt soll getauscht werden, und die beiden werden ausgegeben, also a wird 5 und b wird 3.

a) Programmieren sie das in Python.

b) Sie haben sicher eine Hilfsvariable benutzt, in die sie einen Wert gesichert haben, bevor er vom anderen überschrieben wurde.

Die Aufgabe lautet jetzt aber so : tauschen sie a und b, ohne (!!) eine weitere Variable benutzen zu müssen

a)

```
z1 = int(input("erste Zahl bitte : "))
z2 = int(input("zweite Zahl bitte : "))
zwischen = z2
z2 = z1
z1 = zwischen
print("erste jetzt : ",z1)
print("zweite jetzt : ",z2)
```

b)

```
z1 = int(input("erste Zahl bitte : "))
z2 = int(input("zweite Zahl bitte : "))
z1 = z1 + z2
z2 = z1 - z2
z1 = z1 - z2
print("erste jetzt : ",z1)
print("zweite jetzt : ",z2)
```

Das alles geht natürlich auch viel schöner mit Arrays, indem man die Indizes einfach tauscht.