

#KLOOIENMETCOMPUTERS

ARNOUT VAN KEMPEN OVER ROMMELEN IN EEN DIGITALE WERELD

Toen ik de vorige keer een programmeeropdracht gaf, had ik die zelf nog niet uitgevoerd. Wie aan het worstelen is gegaan, zal snel een bijzondere eigenschap van Python hebben ontdekt: het type van variabelen wordt niet vooraf gedeclareerd, waardoor je vrij makkelijk over het hoofd ziet dat een type fout is. En de `input()` functie die we de vorige keer zagen, levert standaard een string. Het navolgende programma bevat daardoor geen fouten op regel-niveau en toch gaat het fout:

```
a=input()
b=input()
print(a/b)
```

De oplossing is simpel, als je bedenkt dat `input` geen commando is, maar een functie. Functies kan je samenvoegen in de vorm `f(g(x))`. Eerst wordt `g` uitgevoerd en dan `f` op de uitkomst van `g`. Als je dus een getal wilt invoeren, waarmee kan worden gerekend, moet je van de `input()` nog een integer maken, in Python met de functie `int()`.

Een mogelijke uitwerking van de programmeeropgave van vorige keer is daarmee:

```
# programma voor het vereenvoudigen van een breuk

teller=int(input("Teller? "))
noemer=int(input("Noemer? "))
rest=teller%noemer
aliquote=teller//noemer

if rest == 0:
    print(aliquote)
else:
    print(aliquote, rest, "/", noemer)
```

De regel met `#` aan het begin is niet nodig, maar het is wel handig om toelichtingen toe te voegen waarmee op een later moment nog valt te begrijpen wat je precies gedaan hebt.

Voor degenen die zelf op zoek zijn gegaan en de `int()`-functie hebben gevonden: gefeliciteerd, dit is wat mij betreft nou klooiën met computers. Het gaat fout en je worstelt tot je het oplost.

En voor degenen die zich door mij op het verkeerde spoor voelen gezet: klopt, maar weet dat ik zelf ook op het verkeerde spoor zat. Zoals aan het begin van deze serie aangegeven: ik ben zelf ook Python aan het leren en doe dat ook door klooiën, tot het lukt. ←