



Functie zonder parameters en zonder retourwaarde

```
# functie die een groet print
def groet():
    print("Hallo") # print een groet

### HOOFPGRAMMA MET AANROEP VAN FUNCTIE
groet() # roep de functie groet aan
groet() # roep de functie groet aan
```

Functie met parameters maar zonder retourwaarde

```
# functie die een groet print. Functie krijgt 2 parameters mee om mee te werken: naam en land
def groet( naam, land ):
    print("Hallo", naam, "uit", land + "!") # print een groet met de meegegeven naam en land

### HOOFPGRAMMA MET AANROEP VAN FUNCTIE
groet( "Sebas", "Spanje" ) # roep de functie groet aan met "Sebas" voor naam en "Spanje" voor land
groet( "John", "Engeland" )
groet( "Janneke", "Nederland" )
```

Functie met parameters en met retourwaarde

```
# functie die een groet print en het aantal keren dat er gegroet is bijhoudt
def groet( naam, teller ): # functie krijgt 2 parameters mee om mee te werken: naam en land
    print( teller, "hallo", naam ) # print een groet met de meegegeven teller en naam
    teller += 1
    return teller

### HOOFPGRAMMA MET AANROEP VAN FUNCTIE
aantal_keren = 0
aantal_keren = groet( "Sebas", aantal_keren ) # roep de functie aan met "Sebas" en aantal_keren_geroet,
aantal_keren = groet( "John", aantal_keren ) # en sla de nieuwe waarde op
aantal_keren = groet( "Janneke", aantal_keren )
print("Totaal aantal keren gegroet:", aantal_keren) # print aangepaste waarde
```

functie die gegeven kilometers omzet naar meters

```
def rekenKmNaarMeters( afstand_in_km ):
    # definitie die een parameter binnen krijgt om mee te rekenen
    afstand_in_m = afstand_in_km * 1000 # voert berekening uit
    return afstand_in_m # levert afstand_in_m op
```

HOOFPGRAMMA MET AANROEP VAN FUNCTIE

```
aantal_meters = rekenKmNaarMeters( 4 ) # wat de aanroep van rekenKmNaarMeters oplevert opslaan als aantal_meters
print( aantal_meters )
```