# Functies

Voor extra info zie ook :

1/ ELO map Python => pdf pythonboek van Pieter Spronck Hoofdstuk 8 Functies

2/ <https://www.w3schools.com/python/python_functions.asp>

Wat is een functie?

Er zijn 3 verschillende types functies in Python

1/ build in , bv print()

2/gebruiker functies

3/anonieme functies ook wel lambda functies genoemd ( zie later )

Een functie bevat een lijn of blok code die wordt uitgevoerd indien de functie wordt aangeroepen, en een functie vervult een specifieke taak.

Een reden om een functie te gebruiken kan zijn dat je die blok code regelmatig aanroept vanop verschillende plaatsen, of om je complexe code te verdelen in verschillende blokken die bij elkaar horen en een specifieke taak uitvoeren.

Tijdens het aanroepen kan je parameters meegeven naar de functie maar dit is geen verplichting, een functie kan ook iets uitvoeren zonder parameters te krijgen.

Een functie kan parameters teruggeven aan de aanroeper via het keyword return.

Een functie herken je aan het keyword def gevolgd door de functie naam met ronde haken, waarbij tussen de ronde haken eventueel parameters voorkomen.

Een functie wordt later 1 of meerdere keren aangeroepen in je code met de functienaam + ronde haken en eventuele parameters, dus zonder def!

Soms spreekt men ook over een method, dit is niet meer dan een functie die onderdeel uitmaakt van een klasse. ( zie klasse en objecten RP-2)

Argumenten worden meegegeven tijdens de aanroep en komen terecht bij de parameters van de functie

Hoe kan je een functie maken?

'

1/ gebruik def en een functienaam om de functie te declareren

2/ plaats de ronde haken met de eventuele parameters, beëindig de lijn met :

3/ voeg de code toe die je wil uitvoeren indien de functie wordt aangeroepen

4/ beëindig de functie met return indien je iets wil retourneren naar de aanroeper van de functie

* 1. De functie klaar maken ( definiëren)

def functienaam():

Ingesprongen je blok code

1 of meerdere lijnen

* 1. De functie aanroepen

functienaam()

* 1. Parameters meegeven tijdesn de aanrope van de functie

def functienaam(par1,par2):

#statements

functienaam(arg1,arg2)

* 1. Default parameters zorgen ervoor dat een parameter niet per se moet meegeven worden tijdens aanroep

def functienaam(par1=10,par2=20):

#statements

* 1. List parameter zie later
  2. Return Value, python kan 1 of meerdere resultaten terugsturen

def functienaam(a,b,):

statements

return a+b

var = functienaam(arg1,arg2)

* 1. Keyword Arguments zie later

om de orde van parameters tijdens de aanroep te kunnen kiezen in gelijk welke volgorde.

vb:

fctnaam(par1naam , par3naam=”value”, par2naam=”value”)

oefeningen met functies

# oef 1 geef naam en leeftijd mee als parameters met de functie hallo.

indien l <= 18 groet met "hi"+ naam anders "aangenaam" + naam

def hallo(naam,leeftijd):

if leeftijd <= 18:

print("Hi",naam)

else:

print("aangenaam",naam)

n=input("Geef naam: ")

x=int(input("Geef leeftijd: "))

hallo(n,x)

# oef 2 wat gebeurt er precies in onderstaande?

def bewerkingen(a,b):

return a+b,a-b,a\*b,a/b

print(bewerkingen(10,2))

q,x,y,z=bewerkingen(100,40)

# oef 3 default parameters

Test wat er gebeurt hieronder en verklaar!

def som(a,b):

return a+b

def som2(a=0,b):

return a+b

def som3(a,b=0):

return a+b

def som4(a=5,b=10):

return a+b

x1=som(1,2)

x2=som2(2)

x3=som3(1)

x4=som4(1,2)

x5=som4(2)

x6=som4()