



Datum:  
Punten

Naam:

Nummer:

Klas:

Snummer:

4 Zelf gedefinieerde functies

## 1 Nut en gebruik

/0,5

In hoofdstuk twee heb je geleerd wat conversiefuncties zijn.

✍ Noteer er drie: \_\_\_\_\_

Je kan ook je **eigen functies creëren**.

Dat is nodig

- wanneer je **code** wil **bundelen** omdat **code te omvangrijk** wordt om op te nemen in het hoofdprogramma
- als dezelfde **code** **meermaals voorkomt**.

Werkwijze

- Je **definieert** een functie **één keer** met een unieke naam (met of zonder parameters tussen de haakjes) net **voor het hoofdprogramma**.
- Vanuit het hoofdprogramma roep je **de functie op** = call telkens je ze nodig hebt.
- De aangeroepen **functie wordt uitgevoerd**. Zodra ze **eindigt**, gaat de **computer verder** met de instructies na de oproep of call in het hoofdprogramma.

## 2 Programmavoorbeeld

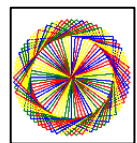
/5

Met de module **turtle** maak je **tekeningen**, zoals in bijgevoegd programma. De schildpad kan vooruit en achteruit stappen (in pixels), naar rechts of links draaien (in graden). Je kan de schildpad optillen en neerzetten. De schildpad start in het midden van het scherm op positie (0,0) en kijkt naar rechts.

```
#Modules importeren
import turtle
#Tekenen
def teken(lengte):
    for teller in range(4):
        turtle.forward(lengte)
        turtle.right(90)
#Programma
turtle.pendown()
turtle.speed(100)
turtle.pensize(3)
turtle.pencolor("blue")
teken(150)
turtle.done()
```

- ✍ Noteer de **naam** van de zelf gedefinieerde **functie**: \_\_\_\_\_
- ✍ Onderstreep de **oproep** van de functie = de **call** in het hoofdprogramma.
- ✍ De functie heeft één **parameter** tussen de haakjes. Noteer de naam: \_\_\_\_\_
- ✍ Noteer de waarde van het **argument** dat bij de call aan de functie wordt doorgegeven: \_\_\_\_\_
- ✍ Wat tekent het programma? \_\_\_\_\_

- 💻 Ontwerp een programma zoals in de figuur rechts, dat bovenstaande tekening 100 keer weergeeft telkens 5° naar rechts gedraaid en in een andere kleur. De kleur wordt willekeurig bepaald met de opdracht `random.choice(kleurenlijst)`. De lijst wordt vooraf in het programma gedefinieerd met de instructie `kleurenlijst = ["red", "green", "yellow", "blue"]`. Op <https://trinket.io/docs/colors> vind je de namen van de kleuren door op een kleurvak te klikken.



Laat controleren door de leerkracht.

## 3 Het return-commando

/2,5

**Argumenten** in de **call-instructie** geven de **informatie** door **aan de functie**. Maar omgekeerd kunnen **functies** net zoals `int()`, `float()` en `str()` **informatie** aan het programma **teruggeven** met het **return-commando** dat tevens de functie beëindigt. Python gaat dan verder met de code na de call.

```
from math import sqrt
def pythagoras( a, b ):
    return sqrt( a * a + b * b )
c=pythagoras( 3, 4 )
print( c )
```

- ✎ Wat is de **naam** van de zelf gedefinieerde **functie**? \_\_\_\_\_
- ✎ Noteer de **namen** van de **parameters**: \_\_\_\_\_
- ✎ Onderstreep de call in het hoofdprogramma.
- ✎ Noteer de waarde van de argumenten die bij de call aan de functie worden doorgegeven: \_\_\_\_\_
- ✎ Noteer de waarde die terugkeert naar het hoofdprogramma: \_\_\_\_\_

## 4 Opdracht Even

/2

Definieer een functie **isEven** die bepaalt of een getal even is. De functie retourneert de waarde True als het getal even is, False als het getal oneven is.

Druk in het hoofdprogramma van de getallen 0 t.e.m. 10 de getallen af die even zijn.

```
0 is even
2 is even
4 is even
6 is even
8 is even
10 is even
```

Laat controleren door de leerkracht.

## 5 Opdracht: Palindroom

/5

### 5.1 Probleemstelling

Ontwerp het programma **Palindroom.py** dat een woord invoert. Een functie **Palindroom** controleert of dat woord een spiegelwoord is. De functie retourneert de waarde True als het woord een palindroom is, False als het geen palindroom is.

### 5.2 Probleemanalyse: bewerkingen met strings

Ieder teken in een string heeft een index (zoals bij lijsten) die de positie in de string bepaalt. De index begint bij 0. Je kan ook een negatieve index gebruiken die start bij -1, de laatste letter van de string.

woord	p	y	t	h	o	n
index	0	1	2	3	4	5
index	-6	-5	-4	-3	-2	-1

Je kan ook

- een deelstring bepalen, bv. woord[0:2] → py
- de lengte van een string berekenen, bv. len(woord) → 6.

- ✎ Noteer de juiste index en gebruik de len()-functie om het laatste teken in het woord Python te bepalen:  
print(woord[ \_\_\_\_\_ ]) → n

### 5.3 Testen

Test het programma met verschillende waarden:

```
Geef een woord: lepel
Het woord is een palindroom!
```

```
Geef een woord: zon
Het woord is geen palindroom.
```

Laat controleren door de leerkracht.