

## 1. Methodes

Methodes zijn functies gedefinieerd in een klasse. Methodes worden gebruikt om operaties uit te voeren op/met de attributen van een klasse.

Een methode is een functie op een object van een klasse die het object zelf aanspreekt met het klasse-argument self.

We illustreren enkele methodes voor de klasse Cirkel voor het berekenen van de omtrek en de oppervlakte.

```
from math import *
from tj_draw import *

class Cirkel:
    def __init__(self,xcoord,ycoord,rad):
        self.xcoord = xcoord
        self.ycoord = ycoord
        self.rad = rad

    def getOmtrek(self):
        return 2*pi*self.rad

    def getOpp(self):
        return pi*self.rad**2
```

```
Python Shell 10/10
>>>#Running circle.py
>>>from circle import *
>>>c=Cirkel(1,2,3)
>>>[c.xcoord,c.ycoord,c.rad]
[1, 2, 3]
>>>c.getOmtrek()
18.84955592153876
>>>c.getOpp()
28.27433388230814
>>>|
```

Methodes kunnen ook gebruikt worden om de attributen van een object te veranderen

```
def setRadius(self,newrad):
    self.rad=newrad

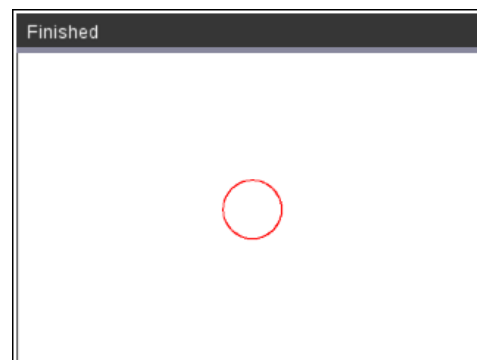
def setCenter(self,newcenter):
    self.xcoord,self.ycoord=newcenter
```

en een object grafisch voor te stellen

```
def tekenCirkel(self):
    w,h=get_screen_dim()
    set_window(-w/2,w/2,-h/2,h/2)
    set_color(255,0,0)
    draw_circle(self.xcoord,self.ycoord,self.rad)
```

```
Python Shell 10/10
>>># Veranderen attributes
>>>c.setCenter(3,5)
>>>[c.xcoord,c.ycoord]
[3, 5]
>>>c.setRadius(10)
>>>c.rad
10
>>>c.getOpp()
314.1592653589793
>>>|
```

```
Python Shell 9/9
>>>#Running circle.py
>>>from circle import *
>>>c=Cirkel(0,0,20)
>>>c.getOmtrek()
125.6637061435917
>>>c.getOpp()
1256.637061435917
>>>c.tekenCirkel()
>>>|
```



## 2. Magic methodes

Voor Python-classes kunnen speciale methodes, magic methodes, geïmplementeerd worden, gebruikmakend van speciale syntax. We maakten al kennis met `__init__()`, de methode die de attributen van de klasse definieert.

We bekijken nog twee andere magic methodes:

- `__str__()` string-representatie van een object
- `__len__()` geeft de (user defined) lengte van een object

We herschrijven de klasse Cirkel als volgt:

```
from math import *
class Cirkel:
    def __init__(self,xcoord,ycoord,rad):
        self.xcoord=xcoord
        self.ycoord=ycoord
        self.rad=rad
        self.omtrek=2*pi*rad
        self.opp=pi*rad**2
```

En we definiëren de volgende `__str__()` en `__len__` magic methodes voor de klasse Cirkel:

```
def __str__(self):
    return "Een cirkel met straal {} en middelpunt ({},{})".format(self.rad,self.xcoord,self.ycoord)

def __len__(self):
    return self.omtrek
```

Deze methodes voeren we als volgt uit op een object:

```
Python Shell 10/10
>>>#Running cirkel.py
>>>from cirkel import *
>>>c=Cirkel(0,0,10)
>>>print(c)
Een cirkel met straal 10 en middelpunt (0,0)
>>>str(c)
'Een cirkel met straal 10 en middelpunt (0,0)'
>>>len(c)
62.83185307179586
>>>|
```

```
Python Shell 11/11
>>>c=Cirkel(-2,2,1)
>>>print(c)
Een cirkel met straal 1 en middelpunt (-2,2)
>>>len(c)
6.283185307179586
>>>c=Cirkel(-2,2,0.5)
>>>print(c)
Een cirkel met straal 0.5 en middelpunt (-2,2)
>>>len(c)
3.141592653589793
>>>|
```