



---

# FUNCTIONEEL PROGRAMMEREN

---

ATP – Opdracht 3



14 APRIL 2024

Naam: Berke Özmuk  
Studentennummer: 1762463  
Docent: Nick Goris  
Vak: ATP

## Inhoudsopgave

Functioneel programmeren .....	2
Pure functies .....	2
Hogere orde functies .....	2

# Functioneel programmeren

De student heeft drie concepten uit FP in zijn project geïmplementeerd:

- Pure functies
- Hogere orde functies
- First class functions

Deze concepten worden in dit document uitgewerkt en toegelicht.

## Pure functies

De pure functies zijn functies die geen bijwerkingen hebben en dezelfde output produceren voor dezelfde input, maakt niet uit wat de context is. Dit zorgt ervoor dat de code voorspelbaar is en makkelijk te begrijpen is. Hieronder zie je een voorbeeld van een pure functie die de student gaat implementeren in zijn project:

```
float fahrenheitToCelsius(float fahrenheit){  
    return (fahrenheit - 32.0f) * (5.0f/9.0f);  
};
```

Zoals je kan zien krijgt de functie een input en wordt de conversie van Fahrenheit naar Celsius de output, ongeacht wat de context is. Dit is een perfect voorbeeld van een pure functie. De functie krijgt een input, geeft een voorspelbare output, is niet afhankelijk van externe toestanden en bevat geen bijwerkingen.

Nadelen van het niet implementeren van een pure functie is, dat de code onvoorspelbaar wordt, het wordt moeilijker om de code te testen en te begrijpen en onderhouden. Dit zijn allemaal nadelen die je niet wilt hebben in je code.

## Hogere orde functies

Hogere orde functies zijn functies die een of meerdere functies als parameter mee krijgen en/of een functie teruggeeft als output. Hieronder zie je een voorbeeld van een hogere orde functie die de student gaat implementeren in zijn project.

```
def calc_runtime(f):  
    def wrapper(*args, **kwargs):  
        start_timer = datetime.now()  
        x = f(*args, **kwargs)  
        print("{} runtime: {} seconds".format(f.__name__,  
datetime.now() - start_timer))  
        return x  
    return wrapper
```

Zoals je kan zien is dit een decorator. De decorator krijgt een functie mee als parameter, maakt een nieuwe functie aan die de runtime berekent van de meegegeven functie en retourneert vervolgens de nieuwe functie die runtime heeft berekent.

Dit laat goed zien dat er gebruikt wordt gemaakt van een hogere orde functie. De decorator krijgt een functie als parameter en retourneert een functie.

Nadelen van het niet implementeren van hogere orde functies is, dat de code minder modulair wordt, kans op herhalende code en code wordt complexer.