

## Functioneel Programmeren

### Doelgroep Cursus Functioneel Programmeren

De cursus Functioneel programmeren is bedoeld voor personen die de principes van functioneel programmeren willen leren en die deze kennis in talen met functionele aspecten willen toepassen.

### Vorkennis Cursus Functioneel Programmeren

Om deel te nemen aan deze cursus dient men te beschikken over basis programmeer vaardigheden en is ervaring met software ontwikkeling vereist.

### Uitvoering Training Functioneel Programmeren

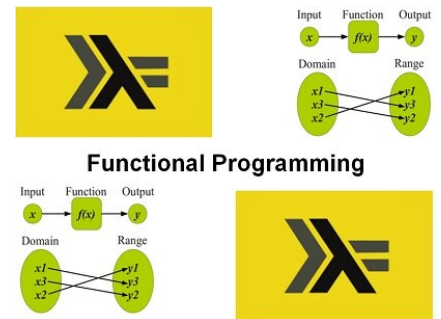
De theorie wordt besproken aan de hand van presentaties. Demo's worden gebruikt om de concepten te verduidelijken. Voorbeelden in verschillende talen worden gebruikt om de concepten te illustreren. De theorie wordt afgewisseld met oefeningen.

### Certificering Functioneel Programmeren

De deelnemers krijgen na het goed doorlopen van de cursus een officieel certificaat Functioneel Programmeren.

Duur: 1 dag

Prijs: € 749

[Open Rooster](#)


## Inhoud Cursus Functioneel Programmeren

In de cursus Functioneel Programmeren worden de basisprincipes van functioneel programmeren besproken en geïllustreerd met voorbeelden. Functioneel programmeren is naast object georiënteerd programmeren een belangrijke stroming in de ontwikkeling van de informatica.

### Intro Functional Programming

De cursus gaat van start met een bespreking van op de achtergronden van functioneel programmeren in de lambda calculus. In het functioneel programmeer paradigma wordt een programma opgebouwd uit functies die operaties uitvoeren op de argumenten.

### Functional Characteristics

Moderne talen zoals Java, Python, [Scala](#) en JavaScript zijn uitgerust met functionele kenmerken maar zijn er niet geheel op gebaseerd.

### Pure Functional Languages

Programmeertalen die in zijn geheel op dit paradigma zijn gebaseerd zijn Erlang, F#, [Haskell](#), Lisp en ML.

### Side Effects

Aan de orde komt hoe in puur functionele programmeertalen functies geen neveneffecten veroorzaken. Dit zijn effecten die invloed hebben op meer dan het resultaat van de functie, zoals het veranderen van een globale variabele. Variabelen die dienen als accumulator, counter of controle variabele horen niet thuis in een functionele taal.

### Map en Reduce

Voorts wordt aandacht besteed aan kenmerkende functionele begrippen zoals immutable data, closures en map en reduce algoritmes.

### Functions en Style

Ook komen in de cursus Functioneel Programmeren de diverse soorten functies zoals first class functies, higher order functies en lambda functies aan bod.

### Other Functional Concepts

Tenslotte staat de bespreking van andere functionele begrippen als Generators en Iterators, Fold Expressions, Memoization en Monads op het programma.

## Modules Cursus Functioneel Programmeren

Module 1 : Intro Functional Programming	Module 2 : Functions and Style	Module 3 : Map and Reduce
What is Functional Programming? Paradigm Origins Lambda Calculus Absence of side effects Immutable Data and Keeping State Maintenance Benefits Functional Languages Erlang an Clojure Haskell and JavaScript Support in Python and Java	Functional Style Imperative Programming Declarative Programming First Class Functions Functions as Parameters Functions as Return Values Pure Functions Anonymous Functions Higher Order Functions Recursion	Replace Iteration Map Parameters Function and Collection Anonymous Functions Lambda's Reduce to Single Value Reduce Parameters Benefits Map and Reduce Filter and Find Function All and Any Function
Module 4 : Pipelines	Module 5 : Other Functional Concepts	
Combining Functions Parallelism Composition Efficiency Issues Referential Transparency Tail Call Optimization Currying Lazy Evaluation Determinism	Generators and Iterators Fold Expressions Memoization Monads Strict Evaluation Type Systems Non-strict Evaluation Evil of Shared State Thread Safeness	