

## Functional Programming

### Audience Functional Programming Course

The course Functional Programming is designed for individuals who want to learn the principles of functional programming and to apply this knowledge in languages with functional aspects.

### Prerequisites Course Functional Programming

To participate in this course experience with software development and basic programming skills are required.

### Realization Training Functional Programming

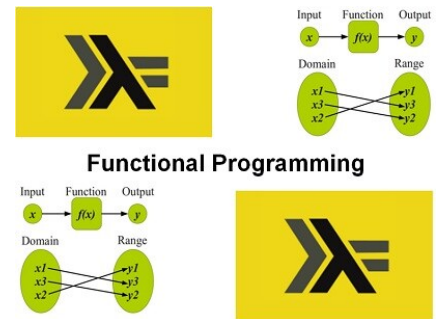
The theory is discussed on the basis of presentations. Demos are used to clarify the concepts. Examples in different languages are used to illustrate the concepts. The theory is interspersed with exercises.

### Certification Functional Programming

Participants receive an official certificate Functional Programming after successful completion of the course.

Duration: 1 day

Price: € 749

[Open Schedule](#)


## Content Course Functional Programming

In the course Functional Programming the basic principles of functional programming are discussed and illustrated with examples. In addition to object-oriented programming, functional programming is an important trend in the development of computer science.

### Intro Functional Programming

The course starts with a discussion of the background of functional programming in the lambda calculus. In the functional programming paradigm a program is built from functions that perform operations on the arguments.

### Functional Characteristics

Modern languages such as Java, Python, [Scala](#) and JavaScript are equipped with functional features but are not entirely based on them.

### Pure Functional Languages

Programming languages based entirely on this paradigm are Erlang, F#, [Haskell](#), Lisp and ML.

### Side Effects

The course covers how functions do not cause side effects in purely functional programming languages. These are effects that affect more than the result of the function, such as changing a global variable. Variables that serve as an accumulator, counter or control variable have no place in a functional language.

### Map and Reduce

Also attention is paid to characteristic functional concepts such as immutable data, closures and map and reduce algorithms.

### Functions and Style

Additionally the course Functional Programming covers the various types of functions such as first class functions, higher order functions and lambda functions.

### Other Functional Concepts

Finally the discussion of other functional terms such as Generators and Iterators, Fold Expressions, Memoization and Monads is on the agenda.

## Modules Course Functional Programming

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	