

Clojure Programmeren

Doelgroep Cursus Clojure Programmeren

De cursus Clojure Programmeren is bedoeld voor een ieder die wil leren programmeren in de functionele programmeer taal Clojure.

Voorkennis Cursus Clojure Programmeren

Om aan deze cursus te kunnen deelnemen is basiskennis van programmeren in een andere programmeertaal bevorderlijk voor de begripsvorming maar niet vereist.

Uitvoering Training Clojure Programmeren

De theorie wordt behandeld op basis van presentatie slides. De theorie wordt verduidelijkt door middel van demo's. Na bespreking van een module is er de mogelijkheid te oefenen. De cursustijden zijn van 9.30 tot 16.30.

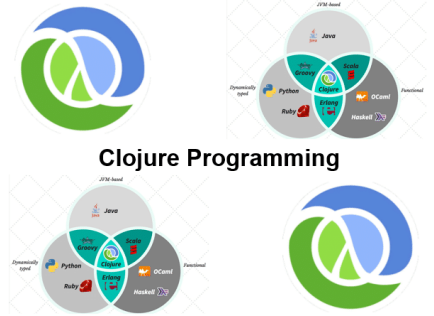
Certificering Cursus Clojure Programmeren

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

Duur: 3 dagen

Prijs: € 1999

Open Rooster



Inhoud Cursus Clojure Programmeren

In de cursus Clojure Programmeren leren de deelnemers programmeren in de dynamische en functionele programmeer taal Clojure. Clojure heeft zijn roots in de Lisp programmeertaal en er bestaan compilers voor zowel Java als .NET zodat Clojure applicaties in beide run time omgevingen kunnen draaien.

Clojure Intro

De cursus gaat van start met een overzicht van de kenmerken van functioneel programmeren en hoe deze door Clojure worden ondersteund. Aandacht is er voor expression evaluation, immutability en concurrency support. Ook wordt gedemonstreerd hoe de REPL loop in Clojure kan worden gebruikt om code te runnen.

Clojure Syntax

Vervolgens wordt ingegaan op de variabelen en data types van de Clojure taal en komen de control flow constructies in Clojure aan de orde. Ook exception handling wordt dan behandeld.

Functies

Functies zijn vanzelfsprekend een centraal element in de functionele taal Clojure en er wordt ingegaan op hoe in Clojure met functies wordt gewerkt. Naast standaard functies worden ook anonymous functies, variadic functies en higher order functies zoals map and reduce behandeld. En ook het doorgeven van functies als parameters en teruggeven van functies als return value komt aan de orde.

Data Structures

Data Structures in Clojure worden eveneens behandeld waaronder sequences, lists, sets, vectors, maps en hashmaps. Hierbij wordt ook ingegaan op de operaties op data structures zoals set unions en intersections en het sorteren van maps.

Advanced Types

En ook advanced types zoals predicates, atoms, agents en watchers komen aan bod.

Multiple Threads

Tenslotte is het tijd voor concurrency met multiple threads in Clojure. Dan passeren ook object locking, transactions, agents en asynchronous calls de revue.

Modules Cursus Clojure Programmeren

| Module 1 : Clojure Intro | Module 2 : Language Syntax | Module 3 : Functions |
|---|--|---|
| Functional Programming Clojure Roots in Lisp Expression Evaluation Clojure Compilers Running in JVM Running in .NET Immutability State Management Concurrency Support Installing Clojure Leiningen Tool CounterClockwise Plugin REPL Loop | Data Types Numbers and Strings char and Boolean nil and Atom Symbols Variables Require Statement Operators for and while doseq and dotimes if and if-do case and cond Exception Handling | Function Definition defn Macro Anonymous Functions Functions with Parameters Passing Arguments Variadic Functions Ampersand Symbol Higher Order Functions Passing Functions Returning Functions filter Function map and reduce iterate and remove |
| Module 4 : Data Structures | Module 5 : Advanced Types | Module 6 : Clojure Concurrency |
| Sequences seq Keyword Creating Lists ISeq Interface List Access Sets Set Intersection Set Union Vectors Maps Key Value Pairs HashMaps SortedMaps | Predicates Metadata vary-meta Atoms swap! Method compare-and-set! StructMaps Agents Watchers add-watch Macros macro-expand Date Class | Threads in Clojure Sharing Data References in Clojure Isolation in Vars Mutable State Object Locking Synchronization with Refs Transactions dosync Blocks Asynchronous Changes Agent State send and send-off await and await-for |