

## Lisp Programmeren

### Doelgroep Cursus Lisp Programmeren

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

### Voorkennis Cursus Lisp 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 Lisp 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 Lisp Programmeren

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

Duur: 3 dagen

Prijs: € 1999

Open Rooster



Lisp Programming



## Inhoud Cursus Lisp Programmeren

In de cursus Lisp Programmeren leren de deelnemers applicaties te ontwikkelen in de functionele programmeer taal Lisp. Lisp is één van de oudste programmeer talen, maar is in de loop van zijn bestaan in belangrijke mate gemoderniseerd. Lisp is een machine onafhankelijke taal en is bij uitstek geschikt van Artificial Intelligence software, omdat Lisp symbolische informatie effectief kan verwerken. De meest gebruikte Lisp dialecten zijn heden ten dage [Common Lisp](#) en [Scheme](#).

### Lisp Intro

De cursus gaat van start met een bespreking van de fundamentals van functioneel programmeren en expressie evaluatie. Vervolgens wordt ingegaan de programma structuur van Lisp applicaties, symbolische expressies en het macro systeem. Ook is er aandacht atoms, list en strings.

### Lisp Syntax

Lisp kent zoals iedere programmeer taal variabelen, types en diverse control flow constructies. Deze worden besproken in de module language syntax.

### Functions

En ook functies in Lisp komen aan de orde met het defun macro, met optionele, rest en keyword parameters en met return values. Hierbij is er tevens aandacht voor typische functionele functies zoals lambda's en mapping functies.

### Data Structures

Vervolgens wordt ingegaan op de diverse data structures die Lisp te bieden heeft zoals sequences, list en arrays, record structures, property lists, trees, hashtables en sets. De operaties op deze data structures zoals unions en intersections tussen sets komen daarbij ook aan bod.

### Lisp I/O

En eveneens wordt input en output in Lisp applicaties besproken. Hierbij wordt ingegaan op Stream I/O, input en output functions en het lezen en schrijven van en naar files.

### Classes and Structures

De cursus wordt tenslotte afgesloten met een bespreking van classes en structures in Lisp. Ook de bundeling van code in packages en het hergebruik van code door middel van inheritance passeren daarbij de revue.

## Modules Cursus Lisp Programmeren

<b>Module 1 : Lisp Intro</b>	<b>Module 2 : Language Syntax</b>	<b>Module 3 : Functions</b>
Functional Programming Lisp Roots Lisp Dialects Common Lisp and Scheme Machine Independence Expression Evaluation Macro System Lisp Executer CLISP Compiler Program Structure Symbolic Expressions Atoms, lists and strings	List Forms Naming Conventions Scalar types Numbers and Characters Symbols typep Predicate type-of Function Variables setq and defvar Operators and Control Flow cond, case, when loop for, dotimes, dolist	Defining Functions defun Macro Parameter Passing Optional Parameters Rest Parameters &rest Symbol Keyword Parameters Return Values return-from Operator Lambda Functions lambda Expression Mapping Functions
<b>Module 4 : Data Structures</b>	<b>Module 5 : Lisp I/O</b>	<b>Module 6 : Structures and Classes</b>
Arrays and Indexing Strings Sequences Sequence Functions Lists cons Record Structure Symbols Property Lists Vectors Fill Pointer Sets Unions and Intersections Trees and Hashtables	I/O Streams Reading Input Input Functions Output Functions Formatted Output File I/O Opening Files :element-type Keyword :external-format Argument with-open-file :direction Keyword :output Keyword Reading and Writing	Defining Structures defstruct Macro Access Functions Constructors Predicates Copier Function Package as Namespace Creating Packages Using Packages *package* Variable Creating Classes defmethod Macro Inheritance