

## Ruby Programmeren

### Doelgroep Cursus Ruby Programmeren

De cursus Ruby Programmeren is bedoeld voor ervaren developers die de scripttaal Ruby willen leren.

### Voorkennis Cursus Ruby Programmeren

Kennis en ervaring met programmeren in een andere programmeertaal, zoals C, C++, C#, Visual Basic, Java of Perl is gewenst.

### Uitvoering Training Ruby Programmeren

De theorie wordt behandeld aan de hand van presentatie slides en wordt afgewisseld met praktische oefeningen. Illustratieve demo's zorgen voor een verdere verduidelijking van de concepten. De cursustijden zijn van 9.30 tot 16.30.

### Certificering Ruby Programmeren

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

Duur: 3 dagen

Prijs: € 1999

[Open Rooster](#)



Ruby Programming



## Inhoud Cursus Ruby Programmeren

In de cursus Ruby Programmeren leren de deelnemers programmeren in Ruby aan de hand van de syntax en taalconstructies van deze scripttaal.

### Ruby Syntax

Na een introductie over de achtergronden van Ruby, de installatie en de wijze waarop Ruby code kan worden uitgevoerd, komen de variabelen, data types en control flow aan de orde.

### Classes en Objects

Ook is er aandacht voor methods en parameter overdracht en voor de object georiënteerde aspecten zoals classes, objects, inheritance en polymorfisme.

### Exception Handling

Daarnaast komt ook het afhandelen van fouten door middel van exception handling ter sprake.

### Modules en Mixins

Vervolgens wordt ingegaan op het indelen van code in modules, modules als namespaces en mixins evenals op het gebruik van modules uit de standard library.

### Blocks en Procs

Ook closures zoals blocks, lambda's en Procs komen aan de orde en er wordt ingegaan op Meta Programming met introspection, Open Classes en dynamic method invocation.

### Ruby Gems

Tenslotte komt het schrijven van Ruby Gems aan bod en wordt een overzicht gegeven van de mogelijkheden van de Standard library.

## Modules Cursus Ruby Programmeren

<b>Module 1 : Ruby Intro</b>	<b>Module 2 : Variables and Data Types</b>	<b>Module 3 : Control Flow</b>
What is Ruby? Ruby Timeline Object Orientation Installation Interactive Ruby Ruby Execution Loading Ruby Code Naming Conventions Executing External Programs Ruby Blocks Resources	Numbers and Big Numbers Strings and String Literals String Interpolation Arrays Hash Range Struct Types of Variables Naming Conventions Constants Pre-defined Variables	Statements Assignment operators Conditionals Multiple Selection while and until Loop for Loop each Iteration Arithmetic operators Comparison operators Ruby truth Logical Operators
<b>Module 4 : Methods and Parameters</b>	<b>Module 5 : Classes and Objects</b>	<b>Module 6 : Exception Handling</b>
Method Definitions Invoking Methods Methods and Parenthesis Return values Default value argument Variable Argument List Array Argument Hash Argument Methods with Code Block Method with Bang Aliasing Methods	Classes and Objects in Ruby Object Initialization Attribute Accessors Current Object Class Variables and Methods Method Visibility Singleton Methods Inheritance Overriding Method Lookup Duck Typing	Error Handling Exception Handling Raising Exceptions Handling Exceptions Exception Class Hierarchy Typed Exception Handling Ensure Block Retry Command Throw and Catch Raising Exceptions User Defined Exceptions
<b>Module 7 : Modules</b>	<b>Module 8 : Closures</b>	<b>Module 9 : Meta Programming</b>
Modules Module Importing Files without Namespacing Modules for Namespacing Namespaces Mixins Mixin Example Include versus Extend Mixins and Inheritance Chain Modules versus Classes Comparable Module Enumerable Module	Benefits of Closures Lambdas and Procs Lambdas with Parameters Procs versus Lambdas Proc as Argument and Return Proc Objects as Closure What are Blocks? Representing Blocks Calling Blocks with Yield Passing Arguments Ampersand Operator From Proc to Block	What is Introspection? Introspection Code Classes are Open Class Definition are Executable Receivers Classes are Objects Dynamic Method Invocation Method Missing Invoking method_missing define_method Evaluating Code Dynamic Typing
<b>Module 10 : Ruby Gems</b>	<b>Module 11 : Ruby Standard Library</b>	
What are Gems? Creating Gems gemspec file Installing and Using the Gem Publish the Gem Grabbing the Gem Rake Writing Tests Documenting Code Using Rdoc Using Bundler	Standard Library Overview Files Accessing Files File Open Modes Reading and Writing Directories Date and Time XML Access DOM Parsing SAX Parsing MultiThreading	