

Ruby Programming

Audience Course Ruby Programming

The course Ruby Programming is intended for experienced developers who want to learn the Ruby scripting language.

Prerequisites Course Ruby Programming

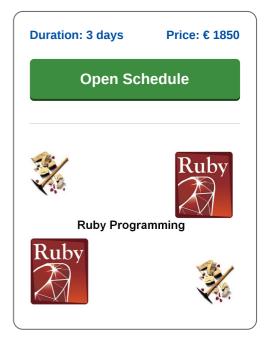
Knowledge and experience with programming in another programming language, such as C, C++, C#, Visual Basic, Java or Perl is required.

Realization Training Ruby Programming

The theory is discussed on the basis of presentation slides and is interspersed with practical exercises. Illustrative demos provide further clarification of the concepts. The course material is in English.

Certification Ruby Programming

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



Content Course Ruby Programming

In the course Ruby Programming participants learn to program in Ruby using the syntax and language constructs of that language.

Ruby Syntax

After an introduction about the background of Ruby, the installation and the way in which code can run, the variables, data types and control flow are discussed.

Classes en Objects

Attention is also paid to methods and parameter passing in Ruby and to object oriented aspects such as classes, objects, inheritance and polymorphism.

Exception Handling

The handling of errors through exception handling is discussed as well.

Modules and Mixins

The course continues with a discussion of partitioning code into modules, modules as namespaces and mixins as the use of modules from the standard library.

Blocks and Procs

Also closures such as blocks, lambdas and Procs are discussed and Ruby meta programming is treated with introspection, Open Classes and dynamic method invocation.

Ruby Gems

Finally writing RubyGems is on the course schedule and an overview of the capabilities of the Standard Library is given.

51 Zwolle. Onlir



Modules Course Ruby Programming

Module 1 : Ruby Intro	Module 2 : Variables and Data Types	Module 3 : Control Flow
What is Ruby?	Numbers and Big Numbers	Statements
Ruby Timeline	Strings and String Literals	Assignment operators
Object Orientation	String Interpolation	Conditionals
Installation	Arrays	Multiple Selection
Interactive Ruby	Hash	while and until Loop
Ruby Execution	Range	for Loop
Loading Ruby Code	Struct	each Iteration
Naming Conventions	Types of Variables	Arithmetic operators
Executing External Programs	Naming Conventions	Comparison operators
Ruby Blocks	Constants	Ruby truth
Resources	Pre-defined Variables	Logical Operators
Module 4 : Methods and Parameters	Module 5 : Classes and Objects	Module 6 : Exception Handling
Method Definitions	Classes and Objects in Ruby	Error Handling
Invoking Methods	Object Initialization	Exception Handling
Methods and Parenthesis	Attribute Accessors	Raising Exceptions
Return values	Current Object	Handling Exceptions
Default value argument	Class Variables and Methods	Exception Class Hierarchy
Variable Argument List	Method Visibility	Typed Exception Handling
Array Argument	Singleton Methods	Ensure Block
Hash Argument	Inheritance	Retry Command
Methods with Code Block	Overriding	Throw and Catch
Method with Bang	Method Lookup	Raising Exceptions
Aliasing Methods	Duck Typing	User Defined Exceptions
Module 7 : Modules	Module 8 : Closures	Module 9 : Meta Programming
Modules	Benefits of Closures	What is Introspection?
Module Importing	Lambdas and Procs	Introspection Code
Files without Namespacing	Lambdas with Parameters	Classes are Open
Modules for Namespacing	Procs versus Lambdas	Class Definition are Executable
Namespaces	Proc as Argument and Return	Receivers
Mixins	Proc Objects as Closure	Classes are Objects
Mixin Example	What are Blocks?	Dynamic Method Invocation
Include versus Extend	Representing Blocks	Method Missing
Mixins and Inheritance Chain	Calling Blocks with Yield	Invoking method_missing
Modules versus Classes	Passing Arguments	define_method
Comparable Module	Ampersand Operator	Evaluating Code
Enumerable Module	From Proc to Block	Dynamic Typing
Module 10 : Ruby Gems	Marketa 44 Duke Otamaland Hilanam	
Module 10 . Ruby Gellis	Module 11 : Ruby Standard Library	
What are Gems?	Standard Library Overview	
	•	
What are Gems?	Standard Library Overview	
What are Gems? Creating Gems	Standard Library Overview Files	
What are Gems? Creating Gems gemspec file	Standard Library Overview Files Accessing Files	
What are Gems? Creating Gems gemspec file Installing and Using the Gem	Standard Library Overview Files Accessing Files File Open Modes	
What are Gems? Creating Gems gemspec file Installing and Using the Gem Publish the Gem	Standard Library Overview Files Accessing Files File Open Modes Reading and Writing	
What are Gems? Creating Gems gemspec file Installing and Using the Gem Publish the Gem Grabbing the Gem	Standard Library Overview Files Accessing Files File Open Modes Reading and Writing Directories	
What are Gems? Creating Gems gemspec file Installing and Using the Gem Publish the Gem Grabbing the Gem Rake	Standard Library Overview Files Accessing Files File Open Modes Reading and Writing Directories Date and Time	
What are Gems? Creating Gems gemspec file Installing and Using the Gem Publish the Gem Grabbing the Gem Rake Writing Tests	Standard Library Overview Files Accessing Files File Open Modes Reading and Writing Directories Date and Time XML Access	

3995 AA Houten