

PRG500 : Ruby Programmeren

Code : PRG500 **Duur :** 3 dagen

Categorie : Scripting

Doelgroep :

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

Voorkennis :

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

Uitvoering :

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



Ruby Programming



Inhoud :

In deze cursus leren de deelnemers programmeren in Ruby aan de hand van de syntax en taalconstructies van deze scripttaal. 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. Ook is er aandacht voor methods en parameter overdracht in Ruby en voor de object georienteerde aspecten zoals classes, objects, inheritance en polymorfisme. Daarnaast komt ook het afhandelen van fouten door middel van exception handling ter sprake. Vervolgens wordt ingegaan op het indelen van code in modules evenals op het gebruik van modules uit de standard library. Ook het gebruik van code blocks, closures in Ruby en introspection passeren de revue. Tenslotte komt het schrijven van Ruby applicaties met een grafische user interface aan de orde en wordt een inleiding gegeven op Ruby Web Applications met Ruby on Rails.

Module 1 : Ruby Intro

- What is Ruby?
- Ruby Timeline
- Object Orientation
- Installation
- Interactive Ruby
- Ruby Execution
- Loading Ruby Code
- Naming Conventions
- Executing External Programs
- Ruby Blocks
- Resources

Module 2 : Variables and Data Types

- Numbers
- Strings
- String Literals
- Arrays
- Hash
- Range
- RegExp
- Struct
- Types of Variables
- Naming Conventions
- Constants
- Local Variables
- Instance Variables
- Class Variables
- Global Variables
- Pre-defined Variables

Module 3 : Control Flow

- Statements
- Assignment
- Assignment operators
- Conditionals
- Multiple Selection
- while Loop
- until Loop
- for Loop
- each Iteration
- Arithmetic operators
- Comparison operators
- Logical Operators
- Ruby truth
- Equality

Module 4 : Methods and Parameters

- Method Definitions
- Invoking Methods
- Method Characteristics
- Return values
- Default value argument
- Variable Argument List
- Array Argument
- Hash Argument
- Methods with Code Block
- Method with Bang
- Method Operators
- Class Methods
- Singletons Methods
- Method Visibility
- private Methods

Module 5 : Classes and Objects

- Classes and Objects
- Objects in Ruby
- Ruby Classes
- Class Syntax
- Object Instantiation
- Getters and Setters
- attr_accessor
- Current Object
- Ruby class
- Inheritance
- Class Methods
- Method Visibility
- Duck Typing

Module 6 : Exception Handling

- Errors and Error Codes
- Need for Exception Handling
- Exception Class
- Exception Class Hierarchy
- Handling Exceptions
- Exception Handling
- User Defined Exceptions
- Raising Exceptions
- Catch and Throw
- List of Exceptions

Module 7 : Modules

- Modules
- Using Include
- Inheritance Chain
- Namespaces
- Modules versus Classes
- Mixins
- Comparable
- EnumerableModules
- Using Include
- Inheritance Chain
- Namespaces
- Modules versus Classes
- Mixins
- Comparable
- Enumerable

Module 8 : Blocks

- Blocks
- What is a block?
- How does a block look like?
- How does block get passed
- How does block get executed?
- Proc object
- Create Proc object
- Execute Proc object
- Proc Object as Argument
- & (Ampersand) operator
- lambda
- Where do blocks get used?
- Closure
- What is a closure?

Module 9 : Introspection

- Introspection
- Dynamic Method Dispatch
- Open Classes
- Method Aliases
- Hooks
- Evaluating Code
- Defining Methods Dynamically

Module 10 : Standard Library

Standard Library
Files
Documenting Code
Rake
Ruby Gems
Unit Testing
Test Driven Development
Assertions
Test Suites

Module 11 : TK GUI

GUI Toolkits
Event Driven Programs
Hello World in TK
Geometry Managers
Allocation Rectangle
Event Handling Example

Module 12 : Ruby on Rails Intro

Ruby on Rails
Model View Controller
ActiveRecord
Scaffold
Assertions