SpriralTrain BV
Standerdmolen 10, 2e verdieping.
3995 AA Houten

spiraltrain@gmail.com
www.spiraltrain.nl
Tel.: (0)30 737 0661
Fax: (0)30 737 1050

KvK nr. : 61618039
BTW nr. : NL854415798B01
IBAN : NL58RABO0156271540
BIC : RABONL2U

PRG400: Cursus Python Programmeren

Code: PRG400 **Duur:** 4 dagen **Prijs:** € 1750

Doelgroep voor de Opleiding Python Programmeren

Deze cursus Python programmeren is voor developers en systeembeheerders die willen leren programmeren in Python en andere personen die Python code willen begrijpen.

Voorkennis voor de Training Python Programmeren

Kennis en ervaring met programmeren is niet strikt noodzakelijk om deel te nemen aan deze cursus. Ervaring in C, C#, Java, Perl of Visual Basic is bevorderlijk voor een goede begripsvorming.

Uitvoering Training Python

De theorie in de cursus Python wordt behandeld aan de hand van presentatie slides. Illustratieve demo's verduidelijken de concepten. De theorie wordt afgewisseld met oefeningen. De cursustijden zijn van 9.30 tot 16.30.

Certificaat Python Programmeren

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

Inhoud van de Python Cursus

In deze cursus leren de deelnemers te programmeren in de object georiënteerde scripttaal Python. Python is een taal die vaak wordt gebruikt voor installatie scripts en voor prototypes van grote applicaties. Na een inleiding over de installatie en de verschillende manieren om scripts uit te voeren, worden de basisbegrippen zoals declaraties, variabelen en control flow structures besproken. Ook wordt aandacht besteed aan de collection structures, zoals Lists, Tuples en Dictionaries. Vervolgens wordt de focus gericht op het gebruik van functies met de verschillende methoden van parameter passing, zoals by value en by reference. Ook de scope van variabelen en lambda functies worden hierbij besproken. Vervolgens wordt aandacht besteed aan de opdeling van Python software in modules en komt het gebruik van namespaces en packages aan de orde. Voorts wordt object georiënteerd programmeren met classes en objects in detail behandeld. In dit opzicht worden concepten als encapsulation, inheritance en polymorphism belicht. Ook de afhandeling van fouten in scripts met behulp van exception handling komt aan de orde. Eveneens staat de functionaliteit van diverse Python library functies voor het benaderen van files op het programma. Voorts komen iterators aan de orde die lazy evaluation mogelijk maken waarbij een object pas wordt gegenereerd als het nodig is, evenals generators en coroutines waarmee concurrent geprogrammeerd kan worden. Ook wordt ingegaan op decorators waarmee functionaliteit zoals caching en proxying, aan bestaande functies en classes kan worden toegevoegd. Tevens wordt aandacht besteed aan database access met de Python Database API. Het gebruik van diverse libraries voor de interactie met XML data wordt behandeld en tenslotte wordt aandacht besteed aan verschillende libraries voor Reguliere Expressies, networking, unit testing en date and time. De verdieping op deze cursus is de cursus Advanced Python Programmeren.

Module 1 : Python Intro	Module 2 : Variables and Data Types	Module 3 : Data Structures
What is Python? Python Features History of Python Getting Started Setting up PATH Environment Variables Running Python Command Line Options Interactive Mode Script Mode Identifiers Reserved Words Comments Lines and Indentation Multi Line Statements Quotes	Variables Multiple Assignment Data Types Python Numbers Numerical Types Number Type Conversions Conversion Functions Built-in Number Functions Python Strings String Operators and Operations Escape Characters String Formatting Triple Quotes Raw and Unicode Strings Built-in String Functions	Sequences and Lists Accessing and Updating Lists Multidimensional Lists List Operations List Functions and Methods Tuples Accessing Values in Tuples Usage of Tuples Tuple Functions Bytes and Byte Arrays Sets and Dictionaries Dictionary Characteristics Accessing Values in Dictionaries Updating Dictionaries Properties of Dictionary Keys Non Mutable Keys Dictionary Methods
Module 4 : Control Flow and Operators	**Module 5 : Functions**	**Module 6 : Modules**
Control Flow Constructs if Statement else Statement elif Statement Nested if while Loop Infinite while Loop for Loop Iterating by Sequence Index break Statement	Functions Function Syntax Calling Functions Pass by Value Pass by Reference Overwriting References Function Arguments Keyword Arguments Default Arguments Variable Length Arguments	Modules import Statement from…import Statement Locating Modules Creating and Using Modules dir Function Python Packages Explict Import Modules Implicit Import Modules Namespaces and Scoping

continue Statement Loop with else Combination pass Statement Python Operators Operator Precedence	Anonymous Functions Syntax Lambda Functions return Statement Scope of Variables	globals and locals Functions reload Function Namespaces and Scoping Test Harnass
Module 7 : Classes and Objects	**Module 8 : Excepting Handling**	**Module 9 : Python IO**
Object Orientation Creating Classes Class Members Creating and Using Objects Accessing Attributes Property Syntax Built-in Class Attributes Constructors and Destructors Encapsulation Inheritance Overriding Methods Class Methods Operator Overloading Polymorphism	Unexpected Errors Exception Handling Typed Exception Handling Exception Handling with Else except Clause Multiple Exceptions Standard Exceptions try-finally Clause Example try-finally Exception Arguments Raising Exceptions Example raising Exceptions User Defined Exceptions	Input and Output IO Module Opening Files File Open Modes Result of Calling open File Object Attributes Reading Binary Files Writing Binary Files Reading Text Files Writing Text Files File Positions Renaming and Deleting Files Directory Methods Creating Directories
Module 10 : Comprehensions	**Module 11 : Generators**	**Module 12 : Decorators**
What are comprehensions? Lambda Operator Filter Reduce and Map Functional Programming Generator comprehensions List comprehensions Dictionary comprehensions Set comprehensions	What are Iterators? Lazy evaluation yielding versus returning itertools module What are Generators? Generator expressions Bidirectional communication Chaining generators Coroutines	What are Decorators? Tweaking original object Replacing original object Decorators on classes Decorators on functions Copying the docstring Examples in library Deprecation of functions while-loop removing decorator Plugin registration system
Module 13 : Python Database Access	**Module 14 : Python and XML**	**Module 15 : Python Libraries**
Python DB API DB API Concepts Check Database Version Using with Inserting Data Prepared Statements Last inserted row id Retrieving Data Fetching Rows Dictionary Cursor Parameterized Queries Metadata Transactions	XML Parsing XML Technologies XML Processing Options Python XML Libraries The XML DOM Building a DOM tree Node Interface Model DOM Navigation DOM Manipulation Simple API for XML SAX Operation SAX Callbacks XML Parsing Models Pull versus Push Parsing XPath XPath expressions XPath axes	Regular Expressions match Function Matching versus Searching Search and Replace Network Layering TCP/IP Layering Connectionless Services Connection Oriented Services Socket utility functions Asynchronous Servers Unit Testing Unit Test Example Date and Time Handling Time Tuple struct-time time Module Calendar Functions