

Python Scripting with Ansible

Overview

This Python Scripting program provides the foundation for the participants to understand the Python scripting.

This program takes someone who has worked on any operating system & exposure to programming to the next level where the learner is exposed to a very powerful cross platform scripting language.

At the end of the program, the participants will be able to understand Python scripts written by other programmers as well as write new Python scripts to automate tasks with extensive hands-on experience.

The program also introduces the learner to some very important and fundamental concepts which are very essential for a person developing applications using cross platform programming languages.

Prerequisites

This course is for people who have worked on any operating system and an editor. The learner should have explored programming concepts & would like to program using the Python Scripting language.

Duration: 5 Days

Topics Covered

Day 1

- **4 stages of application development**
- **4 Generations of Programming Languages**
- **Cross platform development**
 - **Associating an extension with an application... the windows way**
 - **The `#!/usr/bin/python3`**
- **Introduction to Python**
 - **Purely object oriented**
 - **No limits programming language**
- **Friends in the new land (Getting Help)**
 - **`dir`**
 - **`type`**
 - **`help`**
 - **`help()`**
- **Running Python Statements**
- **Writing Python Scripts**
- **Built in functions**
 - **Introduction to a few basic built in functions**

- **Control Flow Statements**
 - **General Syntax - Indentation**
 - **Expressions**
 - **Unary Arithmetic operators**
 - **Binary Arithmetic operators**
 - **Conditionals**
 - **if.. elif .. else**
 - **Loops**
 - **while**
 - **Loop Control Statements**
 - **break**
 - **continue**
 - **Loops**
 - **for loop**
 - **Examples of for loop using lists, tuples & dictionaries**
- **Data Structures**
 - **Lists**
 - **Quick introduction to Objects and Classes**
 - **Using Lists**
 - **Tuple**
 - **Using Tuples**
 - **Tuples and the print statement**

Day 2

- **Dictionary**
 - **Using Dictionaries**
- **String**
- **Sets**
- **I/O Handling**
 - **Files**
 - **open**
 - read mode
 - write mode
 - append mode
 - read write mode
 - write read mode
 - append read mode
 - **read**
 - read
 - readline
 - readlines
 - **write**

- write
 - writelines
 - close
 - seek
 - tell
- **Writing Functions**
 - Defining Function
 - Processing parameters
 - Local Variables
 - Default argument values
 - Keyword Arguments
 - The return statement
 - DocString
 - Lambda
 - Recursive functions
 - Changing the recursive depth
 - Call by value
 - Call by reference

Day 3

- **Modules**
 - Introduction
 - Using the sys module
 - Byte-compiled .pyc files
 - The from..import statement
 - A module's __name__
 - Using a module's __name__
 - Making your own Modules
 - Creating your own Modules
 - from..import
 - The dir() function
 - Using the dir function
- **Packages**
 - The __init__.py
 - Sub Packages
 - Sub Modules

Day 4

- **Object-Oriented Programming**
 - Introduction
 - The self
 - Classes
 - Creating a Class
 - object Methods
 - Using Object Methods
 - The __init__ method
 - Using the __init__ method

- Class and Object Variables
 - Using Class and Object Variables
- Inheritance
 - Using Inheritance
- Polymorphism
- **Errors and Exceptions**
 - Errors
 - Try..Except
 - Handling Exceptions
 - Raising Exceptions
 - How To Raise Exceptions
 - Try..Finally
 - Using Finally
 - User Defined Exceptions

Day 5

- **Debugging using pdb**
 - Setting break points
 - Printing values of variables
 - Getting help
 - Listing the code
 - Stepping into the function
 - Stepping over a function
 - Continue till the next break point
 - Setting temporary break points
 - Listing break points
 - Associating conditions to the break point
 - Enable / disable break points
 - Trace back
 - Move between frames
- **Testing Modules**
 - doctest
 - Pytest
 - Unittest
- **Working with Ansible**
 - Introduction to Ansible
 - Introduction to YAML
 - Ansible Architecture
 - Ansible Inventory
 - The Playbook Grammar
 - Working with Ansible Modules
 - Working with Ansible Variables
 - Ansible Loops
 - Ansible Roles