

Robotic Framework

Day 1,2,3

An Overview of Python

- What is Python?
- Interpreted languages
- Advantages and disadvantages
- Downloading and installing
- Which version of Python
- Where to find documentation

Running Python Scripts

- Structure of a Python script
- Using the interpreter interactively
- Running standalone scripts under Unix and Windows

Getting Started

- Using variables
- String types: normal, raw and Unicode
- String operators and expressions
- Math operators and expressions
- Writing to the screen
- Command line parameters
- Reading from the keyboard

Flow Control

- About flow control
- Indenting is significant
- The if and elif statements
- while loops
- Using lists
- Using the for statement
- The range() function

Sequence Data

- list operations
- list methods
- Strings are special kinds of lists
- tuples
- sets
- Dictionaries

Defining Functions

- Syntax of function definition
- Formal parameters
- Global versus local variables
- Passing parameters and returning values

Working with Files

- Text file I/O overview
- Opening a text file
- Reading text files

- Raw (binary) data
- Using the pickle module
- Writing to a text file

Dictionaries and Sets

- Dictionary overview
- Creating dictionaries
- Dictionary functions
- Fetching keys or values
- Testing for existence of elements
- Deleting elements

Errors and Exception Handling

- Dealing with syntax errors
- Exceptions
- Handling exceptions with try/except
- Cleaning up with finally

Using Modules

- What is a module?
- The import statement
- Function aliases
- Packages

Regular Expressions

- RE Objects
- Pattern matching
- Parsing data
- Subexpressions
- Complex substitutions
- RE tips and tricks

Highlights of the Standard Library

- Working with the operating system
- Grabbing web pages
- Sending email
- Using glob for filename wildcards
- math and random
- Accessing dates and times with datetime
- Working with compressed files

An Introduction to Python Classes

- About o-o programming
- Defining classes
- Constructors
- Instance methods
- Instance data
- Class methods and data
- Destructors
- Conclusion

Day 4-5

1. Robot framework

- a. Introduction

- b. Installation and uninstallation
- c. Demonstrations

2. Creating test data using python and jython

- a. Test data syntax
- b. Creating test cases
- c. Creating test suites
- d. Using test libraries
- e. Variables
- f. Creating user keywords
- g. Resource and variable files
- h. Teardown concept
- i. Advanced features
- j. synchronization, parameterization.
- k. handle web table, frame, alerts

3. Executing test cases using python and jython

- a. Basic usage
- b. Test execution
- c. Post-processing outputs
- d. Configuring execution
- e. Created outputs

4. Robot Framework test automation libraries

- a. Selenium Webdriver
- b. API
- c. Database
- d. XML
- e. Files
- f. Creating custom Robot Framework libraries
- g. AutoIT
- h. log4j
- i. TestNG
- j. etc

5. Robot Framework automated tests in many different ways,

- a. command window,
- b. batch files,
- c. Jenkins CI
- d. Windows task scheduler

6. Collaboration using Git version control

7. Extending Robot Framework

- a. Creating test libraries
- b. Remote library interface
- c. Using listener interface
- d. Extending the Robot Framework Jar
- e. Extend Robot Framework with Java
- f. Hybrid and Keyword Data driven test cases

8. Supporting Tools

- a. Library documentation tool (libdoc)
- b. Test data documentation tool (testdoc)
- c. Test data clean-up tool (tidy)
- d. Other tools distributed with Robot Framework
- e. External tools

9. Appendices

- a. All available settings in test data
- b. All command line options
- c. Test data templates
- d. Documentation formatting
- e. Time format
- f. Internal API