

TECHGENX

Python course + Selenium Automation + AWS + Docker Implementation

Please Note: All below course content will be covered in practical scenarios and regular assignments will be shared. All sessions will be recorded and shared with student for future reference (free of cost).

Please Note: GIT and GIT Hub will be an integral part of the course and will be covered

Core Python:

- Python Introduction
- Python history
- Functions declaration and function calling
- Compiler VS Interpreter
- Operators (arithmetic, Membership, identity, logical, Augmented Assignment, Comparison)
- Expression vs Statements
- Conditional statement (if-elif-else)
- Loop structure (for and While)
- mini-programs in python
- How to import library/Module in python
- Usage of pip command
- some generic libraries like random, os, sys, etc.
- Practical programs on Fibonacci series, Prime Num, Even/Odd
- Develop a “guess a number” game using python
- Truthy vs Falsey values
- List functions:
 - List declaration
 - Fetching value from List
 - append()
 - insert()
 - remove()
 - extend()

- Slicing of List
- Sort()
- list()
- Iterating list using For Loop
- pop()
- count ()
- Clear()
- Reverse()

- Tuples functions:
 - Creating tuples
 - Slicing of Tuple
 - Mutable vs immutable
 - Len() , max() , Count(), Min()
 - Index()
- Dictionary Functions:
 - Declaring a Dictionary
 - Assignment in Dictionary
 - Extracting values from Dictionary
 - Inserting list in Dictionary
 - get()
 - copy()
 - setdefault()
 - fromkeys()
 - pop()
 - keys()
 - values()
 - items()
 - Iterating dictionary using for loop
 - dict()
 - clear()
- Any()
- All()
- Exception Handling
- File Handling

- Os Module
- get current working directory
- change dir
- make dir
- statistics of file
- get Size of file in Bytes
- Absolute vs relative path
- Remove dir
- Create a dynamic file path
- Open A file
- Write a File
- Read A File
- Append A File

Advance Python:

- OOPS:
 - Polymorphism
 - Overriding
 - Overloading
 - Inheritance
 - Simple
 - Multiple
 - Multilevel
 - Hybrid
 - Abstraction
 - Encapsulation
 - Class structure
 - Constructor

Selenium + Python

- 1. Installation
 - 1.1. Introduction
 - 1.2. Downloading Python bindings for Selenium
 - 1.3. Drivers

- 1.4. Detailed instructions for Windows users
 - 1.5. Downloading Selenium server
- 2. Getting Started
 - 2.1. Simple Usage
 - 2.2. Example Explained
 - 2.3. Using Selenium to write tests
 - 2.4. Walk through of the example
 - 2.5. Using Selenium with remote WebDriver
- 3. Navigating
 - 3.1. Interacting with the page
 - 3.2. Filling in forms
 - 3.3. Drag and drop
 - 3.4. Moving between windows and frames
 - 3.5. Popup dialogs
 - 3.6. Navigation: history and location
 - 3.7. Cookies
- 4. Locating Elements
 - 4.1. Locating by Id
 - 4.2. Locating by Name
 - 4.3. Locating by XPath
 - 4.4. Locating Hyperlinks by Link Text
 - 4.5. Locating Elements by Tag Name
 - 4.6. Locating Elements by Class Name
 - 4.7. Locating Elements by CSS Selectors
- 5. Waits
 - 5.1. Explicit Waits
 - 5.2. Implicit Waits
- 6. Page Objects
 - 6.1. Test case
 - 6.2. Page object classes
 - 6.3. Page elements
 - 6.4. Locators
- 7. WebDriver API
 - 7.1. Exceptions
 - 7.2. Action Chains

- 7.3. Alerts
- 7.4. Special Keys
- 7.5. Locate elements By
- 7.6. Desired Capabilities
- 7.7. Touch Actions
- 7.8. Proxy
- 7.9. Utilities
- 7.10. Service
- 7.11. Application Cache
- 7.12. Firefox WebDriver
- 7.13. Firefox WebDriver Options
- 7.14. Firefox WebDriver Profile
- 7.15. Firefox WebDriver Binary
- 7.16. Firefox WebDriver Extension Connection
- 7.17. Chrome WebDriver
- 7.18. Chrome WebDriver Options
- 7.19. Chrome WebDriver Service
- 7.20. Remote WebDriver
- 7.21. Remote WebDriver WebElement
- 7.22. Remote WebDriver Command
- 7.23. Remote WebDriver Error Handler
- 7.24. Remote WebDriver Mobile
- 7.25. Remote WebDriver Remote Connection
- 7.26. Remote WebDriver Utils
- 7.27. Internet Explorer WebDriver
- 7.28. Android WebDriver
- 7.29. Opera WebDriver
- 7.30. PhantomJS WebDriver
- 7.31. PhantomJS WebDriver Service
- 7.32. Safari WebDriver
- 7.33. Safari WebDriver Service
- 7.34. Select Support
- 7.35. Wait Support
- 7.36. Color Support
- 7.37. Event Firing WebDriver Support

- 7.38. Abstract Event Listener Support
- 7.39. Expected conditions Support
-
- 8. Appendix: Frequently Asked Questions
 - 8.1. How to use ChromeDriver ?
 - 8.2. Does Selenium 2 support XPath 2.0 ?
 - 8.3. How to scroll down to the bottom of a page ?
 - 8.4. How to auto save files using custom Firefox profile ?
 - 8.5. How to upload files into file inputs ?
 - 8.6. How to use firebug with Firefox ?
 - 8.7. How to take screenshot of the current window ?
- 9. DataDriven Testing
 - Fetching data from excel using openpyxl, xlwrld, xlrd , etc.

Pytest Framework

- Installation
- How to use Pytest framework
- Generic commands
- Use of Markers
- Use of Pytest.ini file
- Ordering of Test Scripts using Pytest
- What is Fixtures
- How to use Fixtures
- Flake 8
- Flake Report

Docker:

- What is Docker?
- Installation Steps
- Windows ,Mac and Linux Installer
- Troubleshooting with Docker
- Docker Terminologies
- Useful Docker commands
- Pulling an Docker image

- Image Name Format
- Creating a container
- Basic Linux commands
- Port Mapping
- Docker Stop Command
- Volume mapping
- Docker Network
- Creating own Docker image
- Passing Environment variable to Docker container
- Pushing an image to Docker Hub
- Docker compose
- Running test inside Docker container

Docker + Maven:

- What is Maven
- Maven CL
- Maven plugins
- Maven Package
- Running test using Jar file
- Creating image
- Creating dockerfile with dependencies
- Building docker image with maven
- Running test by container
- Architecture of Docker compose files
- Docker compose run with healthcheck script
- Docker compose file for multiple test suites
- Zelenium – Docker components.

Jenkins:

- Introduction
- CI/CD
- Installing Jenkins via Docker
- Plugins
- Master vs Slave concept
- Agent connection
- Launch agent without Master
- Adding Docker with Jenkins
- Jenkinsfile
- Git vs GitHub
- Creating Github account
- Jenkins job with Docker

- Creating Jenkins job for Docker image
- Run test suite

AWS

- AWS terminologies
- AWS Spot Instance Advisor
- AWS Create Account
- AWS-Jenkins requirement
- AWS – Create VPC and subnets
- AWS- creating AMI
- AWS- creating users for Jenkins
- AWS- EC2 plugin install
- AWS- Adding credentials in Jenkins
- AWS-Spot fleet configuration in Jenkins
- AWS- Running our test and auto scaling
- AWS- Copy AM to other regions
- AWS- Cleaning up resources