Infrastructure as Code- Walkthrough

Creating a project

Contents

A **walkthrough** is intended to bring you through a technical exercise. A walkthrough shows you how I completed a task in a particular context, on specific systems, at a point in time. The document is a good basic guide, but you should always confirm that the information is current and if a more recent best practice exists, you should make yourself aware of it.

Introduction

This walkthrough is provided to demonstrate how to get a project structure in Python.

Deliverables

You will be required to keep a list of commands with notes as to why you are using them as well as results of any testing you have done. You may be asked to present these notes as evidence of work completed or as part of a lab book. Check with your lecturer!

Prerequisites

1. I will carry out my exercises using a Windows 11 desktop. You can carry out this work on almost any modern platform.
2. You should have already installed Python as per my notes.
3. You should have already installed VSC as per my notes.
4. You have created a directory to keep example files in and you are ready to code.
    a. This could be on your OneDrive, in these examples, I am using **OneDrive\Python\Exercises_12**

Getting Started

First, I'm going to create a basic directory structure for the project. My root directory is called **Exercises_12**. I'm going to store any packages in a directory called **source**. I write a simple batch file to create the necessary directories.

I'm calling my project...**Project**

```
@echo off
cls
echo "***********************************************"
echo This batch file will create a project directory
echo This is for demonstration purposes only.
echo "***********************************************"
echo *** press [ctrl][c] to exit or any key to continue ***
pause
set /p NAME=Enter the name of the project, then press [return]
echo Creating %NAME%
mkdir %NAME%
cd %NAME%
mkdir Documentation
mkdir Tests
mkdir Examples
mkdir Source
cls
dir
echo "***********************************************"
echo Finished creating project %NAME%
echo "***********************************************"
cd ..
```

Next, I steal some code from Walkthrough 7, to determine what operating system I'm working with. I save this under **Project\Source**

```
"""
directory_utilities.py
By: JOR
Date: 01OCT22
"""

import os, platform

# Define global variables
current_working_directory = None

def detect_os()->str:
   # Detect the OS in use
   return platform.system()

def detect_working_directory()->str:
   # Returns the directory this script was run from
   return os.getcwd()

if (__name__ == '__main__'):
   print(f"This module executes as a standalone script")

   # Check the OS in use, lower case
   my_os = detect_os()
   my_os = my_os.lower()

   # Parse the response, only check for Windows and Linux
   if my_os == "windows":
      print("Your system is Windows")
   elif my_os == "linux":
      print("Your system is Linux")
   else:
      print(f"Cannot continue, unidentified system = {my_os}")
      sys.exit()

   # Get the current working directory
   current_working_directory = detect_working_directory()
   print(f"You are coding in: {current_working_directory}")

else:
   print(f"This module is called {__name__} and is being called by another script")
```

I create a file **main.py** under the Project directory.

```
"""
main.py
By: JOR
Date: 20OCT22
"""

from Source.diretory_utilities import detect_os, detect_working_directory
print(detect_os())
print(detect_working_directory())
```

This should let us check to see that the program structure is generally working.