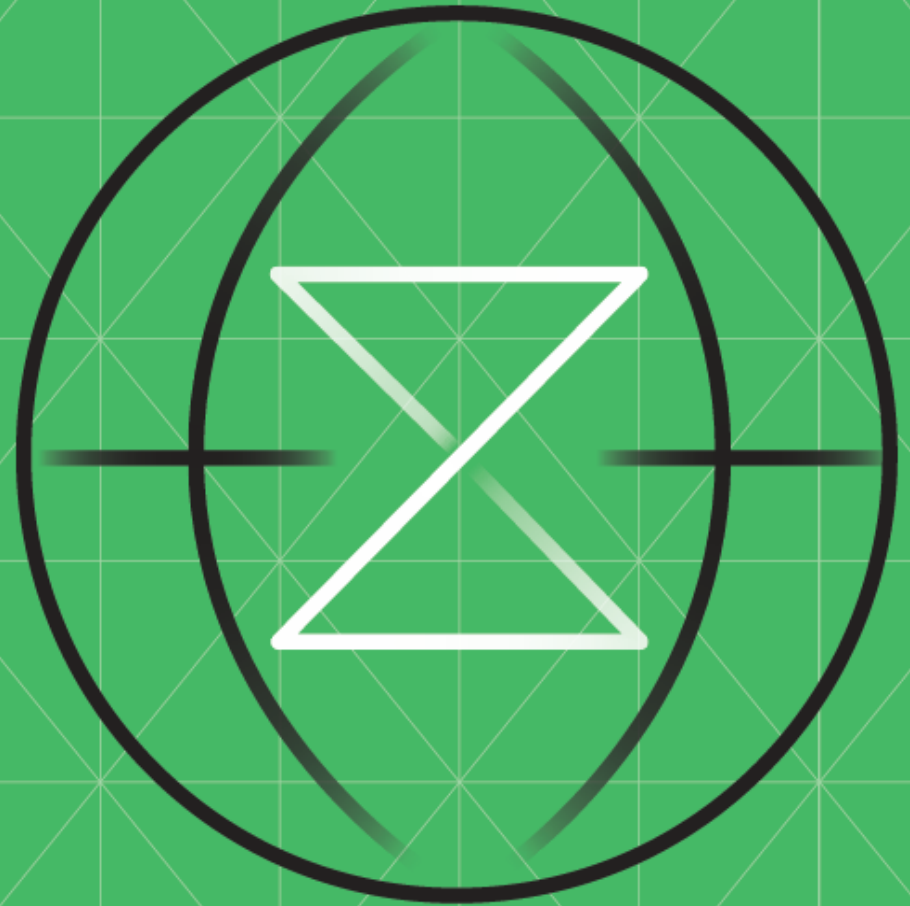


# PYT1

## More Python on z/OS

- [USING PYTHON ON Z/OS](#)
- [1 GETTING THE CODE](#)
- [2 CREATE THE FILE](#)
- [3 WRITING THE CODE](#)
- [4 KICKING IT OFF!](#)
- [5 SUMMARY](#)



# USING PYTHON ON Z/OS

Using Python work with data in USS and MVS

## The Challenge

During this challenge we will be using Python code and execute this in z/OS. You will be able to do some basic Dataset management as well as setting variables.

## Before You Begin

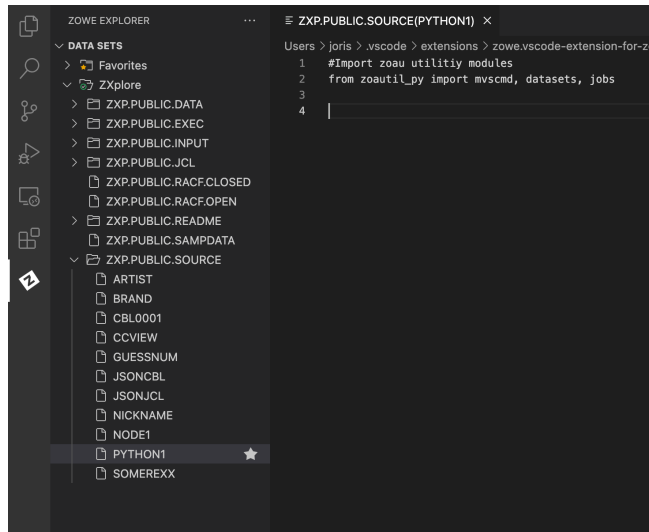
Make sure you have a basic understanding of using a TSO interface, Dataset management, USS and SSH

## Investment

Steps	Duration
4	45 minutes

# 1 GETTING THE CODE

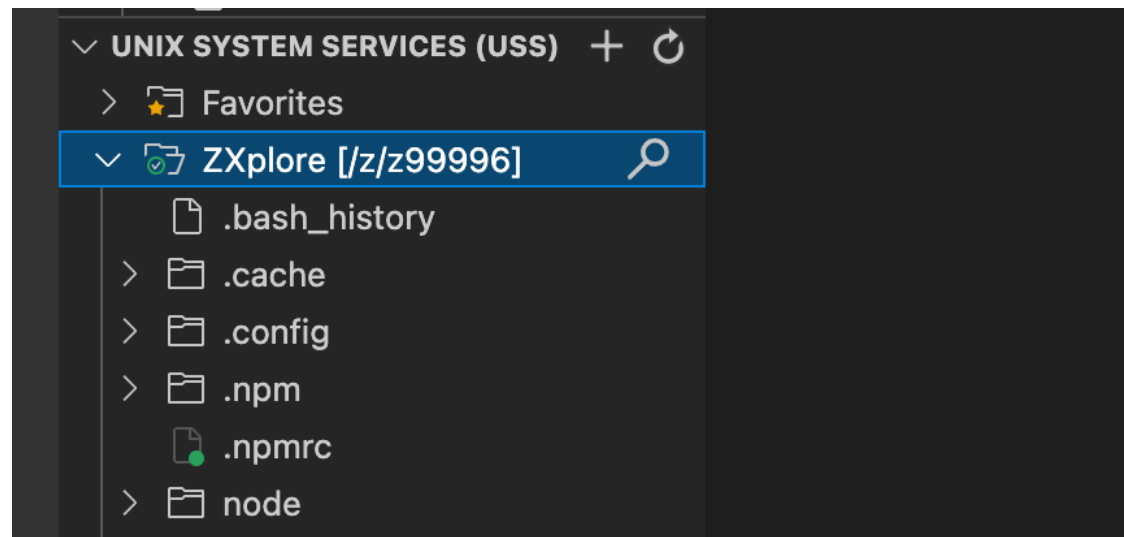
Use VSCode to logon to the zXplore platform. set a filter on the datasets using "ZXP.PUBLIC". Locate the Dataset "ZXP.PUBLIC.SOURCE" and open the Dataset member called "PYTHON1". Select all the text and copy this with Ctrl+c or cmd+c, depending on your workstation operating system.



PYTHON1

Now use the USS interface in VSCode and open your own user directory. You can do this by setting a filter "/z/zxxxxx" where you replace the zxxxxx with your userid.

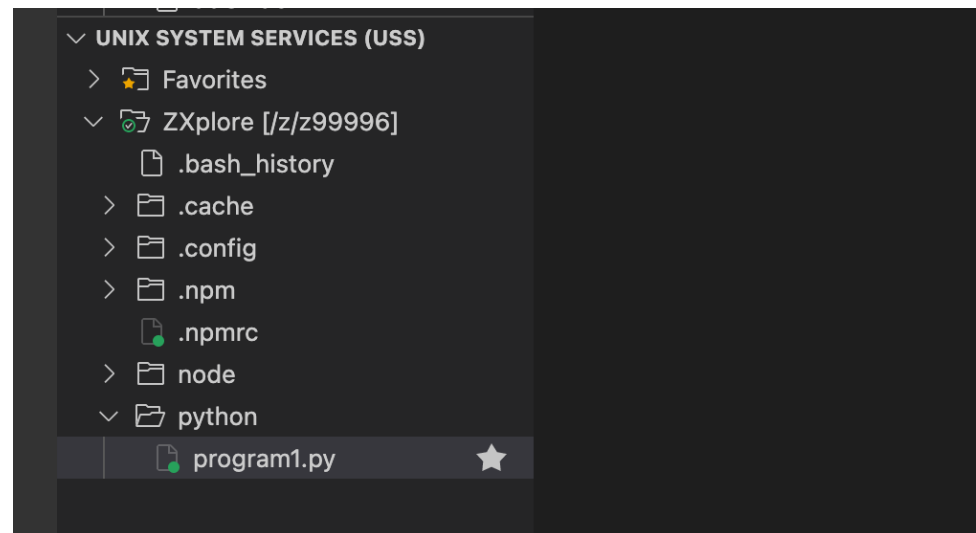
PYTHON1



## 2 CREATE THE FILE

The next step is to create an empty file which we will use to write our program code. Optionally, you can create a folder first to store your file in to keep your directory structured. Right-click on your home folder and select “reate folder or Create file”. Give the folder and file any name you like as long as the filename ends with .py to give it the Python extension.

You should end up with something similar like this.



PYTHON1

Now paste the codelines we copied from the “ZXP.PUBLIC.SOURCE(PYTHON1)” member into this file.

```
1 #Import zoau utilitiy modules
2 from zoautl_py import mvscmd, datasets, jobs
3 import os
4
5
```

PYTHON1

Great! you just entered your first and very important code lines. What this code does is import the Python modules installed on the zXplore system which allows the code to understand specific code statements.

The module called `zoautl_py` enabled us to use code to interact with z/OS while the `os` module enables us to get information from the operating system itself.

PT11|23022-1428

### 3 WRITING THE CODE

Ok, lets see what we can do. First we want to set a HLQ ( remember High Level Qualifier) to make sure that everything we do will end up in your own addressspace where you have "write" access.

Enter the code below and make sure you replace the userid with your own id.

```
#set user ID as the High Level Qualifier. ENTER YOUR OWN USERID HERE.  
HLQ="z99996"
```

```
#Import zoau utilitiy modules  
from zoautl_py import mvscmd, datasets, jobs  
import os  
  
# set user ID as the High Level Qualifier. ENTER YOUR OWN USERID HERE.  
HLQ="z99996"
```

PYTHON1

Now we are going to have the code create a dataset with data inside it. Copy/ Paste the code below where you will create a sequential dataset called "PYTHON.DATA" and there the HLQ will be inserted in front.

```
# create a sequential dataset  
datasets.create("%s.PYTHON.DATA" % HLQ, type="seq", record_length = 80, record_format="FB",  
primary_space="100k")
```

Use the code below to store data in this dataset. the data is in our case just some text.

```
# write your data in the dataset
datasets.write("%s.PYTHON.DATA" % HLQ, content='THIS IS JUST CONTENT CREATED BY USING PYTHON')
```

```
#Import zoau utility modules
from zoautl.py import mvscmd, datasets, jobs
import os

# set user ID as the High Level Qualifier. ENTER YOUR OWN USERID HERE.
HLQ="z99996"

# create a sequential dataset (type=seq)
datasets.create("%s.PYTHON.DATA" % HLQ, type="seq", record_length = 80, record_format="FB", primary_space="100k")

# write your data in the dataset
datasets.write("%s.PYTHON.DATA" % HLQ, content='THIS IS JUST CONTENT CREATED BY USING PYTHON')
```

PYTHON1

That is it. if we kick off this code it will create a dataset and entering the specific data. But before we do that let's add a small extra. In order to see what data is inserted we can enter some code which will show the data inserted. In addition we can also read the user who did enter the data and display that as a return code.

We can insert the code as below to read and display the data entered in the dataset

```
# check if the data is in the dataset
print(datasets.read("%s.PYTHON.DATA" % HLQ))
```

In addition we can also read the user id who is executing the script by setting the user as a variable.

```
# get your username from the system and store it in user
user=os.environ.get("USER")
```



Now that we have the username we can do several things with this. For now we just include this in our return message.

```
#print user
print('this program is executed by' +user)
```

Your code should look something like this now

```
1  #Import zoau utility modules
2  from zoautl.py import mvscmd, datasets, jobs
3  import os
4
5  # set user ID as the High Level Qualifier. ENTER YOUR OWN USERID HERE.
6  HLQ="z99996"
7
8  # create a sequential dataset (type=seq)
9  datasets.create("%s.PYTHON.DATA" % HLQ, type="seq", record_length = 80, record_format="FB", primary_space="100k")
10
11 # write your data in the dataset
12 datasets.write("%s.PYTHON.DATA" % HLQ, content='THIS IS JUST CONTENT CREATED BY USING PYTHON')
13
14
15 # check if the data is in the dataset
16 print(datasets.read("%s.PYTHON.DATA" % HLQ))
17
18 # get your username from the system and store it in user
19 user=os.environ.get("USER")
20 #print user
21
22 print('this program is executed by' +user)
```

PYTHON1

PT1123022-1428

## 4 KICKING IT OFF!

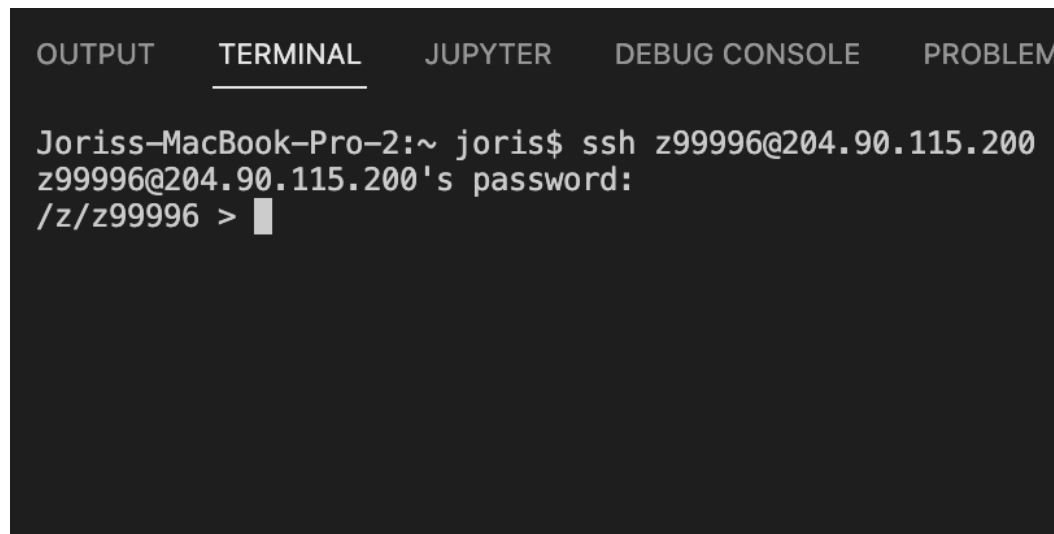
Let's run this code!

In VSCode open a new terminal by using the menu option "terminal, new terminal".

We need to make a SSH connection towards the zXplore environment by entering the following on the terminal command line:

`ssh userid@204.90.115.200` - replace the word *userid* with your zXplore ID.

It will ask for a confirmation to store the connection fingerprint the first time. Type "yes" to acknowledge. It will now ask you for your password. Enter your zXplore password here. (**Note** the cursor will not move or display your password.)



The screenshot shows a VS Code terminal window with the 'TERMINAL' tab selected. The terminal output is as follows:

```
Joriss-MacBook-Pro-2:~ joris$ ssh z99996@204.90.115.200
z99996@204.90.115.200's password:
/z/z99996 > █
```

PYTHON1

Ok we are in; we now can navigate into the folder where we stored our Python script or, display the directory by using the `ls` command

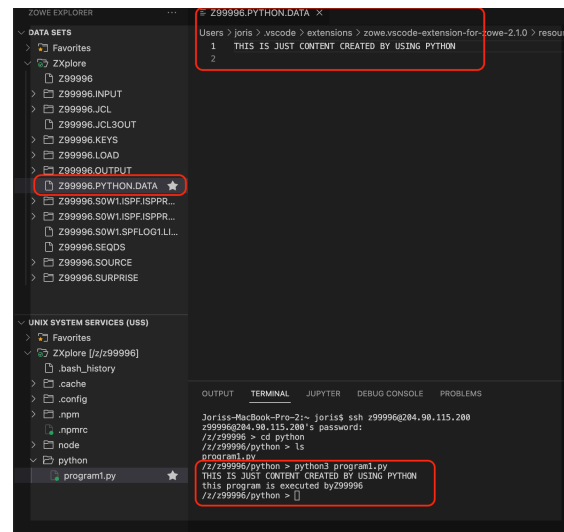
```
Joriss-MacBook-Pro-2:~ joris$ ssh z99996@204.90.115.200
z99996@204.90.115.200's password:
/z/z99996 > cd python
/z/z99996/python > ls
program1.py
/z/z99996/python > █
```

## PYTHON1

Execute the command with the “python3” command

```
python3 program1.py
```

Observe what happens. We expect that a dataset called “ZXXXXX.PYTHON.DATA” will be created in your MVS name space. You will also see the string of data in your return code as well as your userID.



## PYTHON1

## 5 SUMMARY

Congratulations!

You just executed your Python script on z/OS which:

- Created a dataset
- entered data in the dataset
- read the data from that dataset and displayed it
- read and displayed your user id.

There are many more functions you can do with Python on z/OS.

Feel free to explore more Python commands.