Activity – access values in data types

In a script called **A10-access.py** do the following - Follow Q&A as in you did in bash

a Define these variables

```
s='Python Programming!'
L=[1,2,3,4,5,6,[7,8,9]]
T=("Hello","Ciao","Hola")
D={'pop':3, 'jazz':10, 'blues':6}
```

b Access the last element of s, and print it to screen

c Access s and use slice notation to extract the substring `Python` and print it to screen

d Reverse the string s and print it to screen.

e Slice s from element at index 2 to element at index 10 in step of 2. Print it to screen.

f Use slice notation to extract from L the sub-list made up by the first 4 elements.
Assign the new list to variable L1 and print it to screen.

Activity – access values in data types

In **A10-access.py**

g Use L, and double indexing [ ] [ ] to extract the number 9 from L, and print it to screen

h Access values in T, and print Ciao to screen

i Access the value 10 of dictionary D, and print it to screen

j Use one print function to print to screen string s, and dictionary D.

Run the script, and the output should be:

```
!
Python
!gnimmargorP nohtyP
to rg
[1, 2, 3, 4]
9
Ciao
10
Python Programming! {'pop': 3, 'jazz': 10, 'blues': 6}
```

Activity – create data types

In a script called **A10-create.py** do the following - Follow Q&A as in you did in bash

a Define these variables

```
L=['yellow','orange','blue']
number=[2,4,7,9]
fruits=['apple','orange','pear','banana']

s='color'
ss='c o l o r'

D={1:'a', 2:'b', 3:'c'}
T=('Hello', 2, 100)
```

b Create this string out of list L. Use a string method. Assign the new string to variable s1 and print it to screen.

```
yellow*orange*blue
```

c Create this string out of list L. Use a string method and slice notation.
Assign the new string to variable s2 and print it to screen. Notice items of list L are reversed.

```
blue*orange*yellow
```

Activity – create data types

In **A10-create.py**

d Create this list out of string s and print it to screen. Use a function.

```
['c', 'o', 'l', 'o', 'r']
```

e Create this list out of string ss and print it to screen. Use a method.

```
['c', 'o', 'l', 'o', 'r']
```

f Create list of values from dictionary D and print it to screen. Use a function and a method.

```
['a', 'b', 'c']
```

g Unpack the tuple T, and use one of the new variables to print to screen:

```
Hello
```

Activity – create data types

In **A10-create.py**

h Create a tuple (T1) out of list L. print T1 to screen.

```
('yellow', 'orange', 'blue')
```

i Use dict(zip()) to create this dictionary out of lists fruits and number. Print the dictionary to screen.

```
{'apple': 2, 'orange': 4, 'pear': 7, 'banana': 9}
```

Run the script, and the output should be”

```
yellow*orange*blue
blue*orange*yellow
['c', 'o', 'l', 'o', 'r']
['c', 'o', 'l', 'o', 'r']
['a', 'b', 'c']
Hello
('yellow', 'orange', 'blue')
{'apple': 2, 'orange': 4, 'pear': 7, 'banana': 9}
```

dictionary of Morse Code

Download the script **A10-morse.py** from the Canvas scripts folder. The script contains a dictionary called `morse`

Translate your first name or last name, or simply the word HELLO in Morse code by using the dictionary `morse` where each key is an English letter, and its value is the corresponding Morse code.

Edit the script to print out your name in morse code, or simply the word HELLO
You should access each value in dictionary morse and print it to screen

HELLO in morse code should be:

```
.... . .-.. .-.. ---
```

Activity - argv from sys module

Make a script called **A10-argv.py** and in it:

Make a script that takes two numbers separated by a space from the command line, adds those two numbers, and prints the arithmetic result to the screen.

Example:
If you pass 100 200 from the command line, this should be printed to the screen
300

Activity - nested types

In a script called **A10-nested.py** do the following

a. Define this dictionary

```
# If you have a long line of code that you want to break up among multiple lines, you can use \

D={'s1':'GACTC', \
's2':'134,256',\
'd1':{'tuple1':(1,2,3),'L1':[1,2,3,4,5,6]},\
'L2':['a','b','c','d'],\
'L3':[{'d3':'you made it'},'bravo']}
```

b. Access values in D and print to screen

```
bravo you made it
```

Submit to A10

- **A10-access.py**
- **A10-create.py**
- **A10-morse.py**

- **A10-argv.py - optional**
- **A10-nested.py - optional**