Department of Computer Science and Engineering
PES University, Bangalore, India

Lecture Notes
Python for Computational Problem Solving
UE23CS151A

Lecture #51
Functions: Local and Global variables

By,
Prof. Sindhu R Pai,
Anchor, PCPS - 2023
Assistant Professor
Dept. of CSE, PESU

Verified by,
PCPS Team - 2023

More on Functions

There are two types of variables:

Global Variables: Variables created outside all the functions and accessible to all the functions. Scope of Global variable is within the file and outside that file too.

Local variables: Variables created inside the function and accessible only to that function. Scope of the local variable is within the function.

Example code 1:

```
x=10
print("global first outside",x)
def f1():
        x=11
        #local variable x.
        #Life and scope is only within this function
        print("inside",x)
f1()
print("global second outside",x)
```

```
C:\Users\Dell>python notes_functions.py
global first outside 10
inside 11
global second outside 10

C:\Users\Dell>▪
```

Example code 2:

```
x=10    # Global variable.
        #Can be used anywhere inside this file/code
def f1():
        print("inside",x)
f1()
print("outside",x)
```

```
C:\Users\Dell>python notes_functions.py
inside 10
outside 10

C:\Users\Dell>
```

Example code 3: Global variable is readonly inside a function. We cannot modify the value of that variable inside the function directly

```
x=10
print(x)
def f1():
        x=x+1  #UnboundLocalError.
        print("inside",x)
f1()
print("outside",x)
```

```
C:\Users\Dell>python notes_functions.py
10
Traceback (most recent call last):
  File "C:\Users\Dell\notes_functions.py", line 264, in <module>
    f1()
  File "C:\Users\Dell\notes_functions.py", line 262, in f1
    x=x+1          #UnboundLocalError. Global variable is readonly inside a function. We cannot m
odify the value of that variable inside the function directly
    ^
UnboundLocalError: cannot access local variable 'x' where it is not associated with a value
```

Note: If you want to modify the global variable inside the function, then use the keyword global with the variable name

Example code 4: Usage of global keyword

```
x=10
print("global first outside",x)
def f1():
        global x
        x=x+1
        print("inside",x)
f1()
print("global second outside",x)
#If you interchange the first two statements inside the function, it results in Error.
```

```
C:\Users\Dell>python notes_functions.py
global first outside 10
inside 11
global second outside 11
```

Example code 5: There is no block scope in python.

```
for i in [23,55,77]:
        print(i)
print(i)
```

```
C:\Users\Dell>python notes_functions.py
23
55
77
77
```

Example code 6: To make changes to a global variable, global keyword not required inside a block.

```
i = 78
print(i)
for i in [23,55,77]:
        print(i)
print(i)
```

```
C:\Users\Dell>python notes_functions.py
78
23
55
77
77
```

Try it!

- **Create a global list variable. Try making changes inside the function and observe the output**

- **Create a global tuple variable. Have a list inside tuple. Try modifying the list inside a tuple using the function.**

- **Can you create a global variable inside a function and make it available to all other functions in the same file?**

-END-