Kingdom of Saudi Arabia
المملكة العربية السعودية
Ministry of Education
وزارة التعليم
Umm AlQura University
جامعة أم القرى
Adham University College
الكلية الجامعية بأضم
Computer Science Department
قسم الحاسب الآلي

Programming Languages Course, Chapter Five

Summer Semester of 2018 Academic Year

Global and Local Variables in Python

- Global variables are the one that are defined and declared outside a function and we need to use them inside a function.

```
# This function uses global variable s
def f():
    print s

# Global scope
s = "I love Geeksforgeeks"
f()
```

The Output:

```
I love Geeksforgeeks
```

- If a variable with same name is defined inside the scope of function as well then it will print the value given inside the function only and not the global value.

```
# This function has a variable with
# name same as s.
def f():
    s = "Me too."
    print s

# Global scope
s = "I love Geeksforgeeks"
f()
print s
```

The Output:

```
Mee too.

I love Geeksforgeeks
```

The variable s is defined as the string “I love Geeksforgeeks”, before we call the function f(). The only statement in f() is the “print s” statement. As there is no local s, the value from the global s will be used.

- The question is, what will happen, if we change the value of s inside of the function f()? Will it affect the global s as well? We test it in the following piece of code:

```
def f():
    print s

    # This program will NOT show error
    # if we comment below line.
    s = "Me too."

    print s

# Global scope
s = "I love Geeksforgeeks"
f()
print s
```

The Output:

```
Line 2: undefined: Error: local variable 's' referenced before assignment
```

- To make the above program work, we need to use "global" keyword. We only need to use global keyword in a function if we want to do assignments / change them. global is not needed for printing and accessing. Why? Python "assumes" that we want a local variable due to the assignment to s inside of f(), so the first print statement throws this error message. Any variable which is changed or created inside of a function is local, if it hasn't been declared as a global variable. To tell Python, that we want to use the global variable, we have to use the keyword "global", as can be seen in the following example:

```
# This function modifies global variable 's'
def f():
    global s
    print s
    s = "Look for Geeksforgeeks Python Section"
    print s

# Global Scope
s = "Python is great!"
f()
print s
```

Now, there's no ambiguity.

The Output:

```
Python is great!

Look for Geeksforgeeks Python Section.

Look for Geeksforgeeks Python Section.
```

- A good Example:

```
a = 1

# Uses global because there is no local 'a'
def f():
    print 'Inside f() : ', a

# Variable 'a' is redefined as a local
def g():
    a = 2
    print 'Inside g() : ',a

# Uses global keyword to modify global 'a'
def h():
    global a
    a = 3
    print 'Inside h() : ',a

# Global scope
print 'global : ',a
f()
print 'global : ',a
g()
print 'global : ',a
h()
print 'global : ',a
```

The Output:

```
global :  1

Inside f() :  1

global :  1

Inside g() :  2

global :  1

Inside h() :  3

global :  3
```

Reference:

https://www.geeksforgeeks.org/global-local-variables-python/

Remember, *"Success is 1% inspiration and 99% perspiration"* 😉

If you have any questions, feel free to ask me through my email
T.Mariah Sami Ahmed Khayat
Teacher Assistant @ Adam University College
mskhayat@uqu.edu.sa