

LECTURE

Thirteen



Python Variable Scope

In Python, we can declare variables in three different scopes. we can classify Python variables into three types:

1. Local Variables
2. Global Variables
3. The global Keyword

Python Local Variables

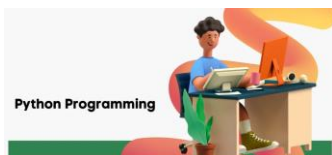
- When we declare variables inside a function, these variables will have a local scope (within the function).
- We cannot access them outside the function.

Python Local Variables
<pre>def greet(): message = 'Hello' print('Local', message) greet() print(message)</pre>

Output

Local Hello

NameError: name 'message' is not defined



Python Global Variables

- In Python, a variable declared outside of the function or in global scope is known as a global variable.
- This means that a global variable can be accessed inside or outside of the function.

Python Global Variables

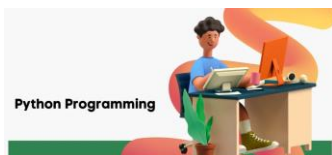
```
message = 'Hello'
def greet():
    print('Local', message)

greet()
print('Global', message)
```

Output

Local Hello

Global Hello



The global Keyword

- We only need to use the global keyword in a function if we want to do assignments or change the global variable.

The global Keyword
<pre>def f(): global s s += ' BIC' print(s) s = "BIT" print(s) s = "Python is great!" f() print(s)</pre>

Output

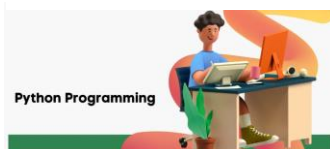
Python is great! BIC

BIT

BIT

Local Variable Vs. Global Variables

Comparison Basis	Global Variable	Local Variable
Definition	declared outside the functions	declared within the functions
Lifetime	They are created the execution of the program begins and are lost when the program is ended	They are created when the function starts its execution and are lost when the function ends
Data Sharing	Offers Data Sharing	It doesn't offers Data Sharing
Scope	Can be access throughout the code	Can access only inside the function
Parameters needed	parameter passing is not necessary	parameter passing is necessary
Storage	A fixed location selected by the compiler	They are kept on the stack
Value	Once the value changes it is reflected throughout the code	once changed the variable don't affect other functions of the program



Python Modules

- Python Module is Py file that contains define functions, classes, and variables.
- There are many Python modules, each with its specific work.
- Grouping related code into a module makes the code easier to understand and use. It also makes the code logically organized.
- There are several built-in modules in Python, which you can import whenever you like.
- Math module has been explained in first semester of Programming Fundamentals (I).
- To create a Python module, write the desired code and save that in a file with **.py** extension.
- Let's create a simple **calc.py** in which we define two functions, one **add** and another **subtract**.

A simple module, calc.py

```
def add(x, y):  
    return (x+y)  
  
def subtract(x, y):  
    return (x-y)
```

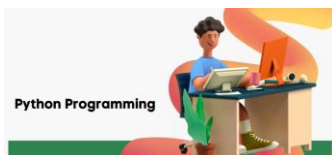
Now, we are importing the **calc** that we created earlier to perform add operation.

Syntax to Import Module in Python

```
import calc  
  
print(calc.add(10, 2))
```

Output

12



Python Built-in Module (Python Calendar module)



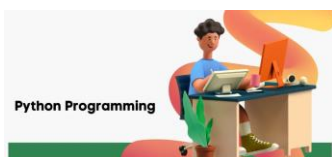
- Python has a built-in Python Calendar module to work with date-related tasks.
- Using Calendar module, we can display a particular month as well as the whole calendar of a year.
- These calendars have Monday as the first day of the week, and Sunday as the last (as per the European convention).

Example: Write a Python program to read year and displays the calendar of the year based on Calendar module.

```
import calendar
yy = int(input("year= "))
print(calendar.calendar(yy))
```

Example: Write a Python program to read year and month and displays the calendar for that month of the year based on Calendar module.

```
import calendar
yy = int(input("year= "))
mm = int(input("month= "))
print(calendar.month(yy, mm))
```



Example: Write a Python program to read year, month and day, displays the day of week based on Calendar module.

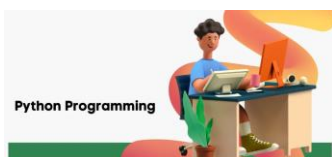
```
import calendar
dw = {
    0: 'Monday',
    1: 'Tuesday',
    2: 'Wednesday',
    3: 'Thursday',
    4: 'Friday',
    5: 'Saturday',
    6: 'Sunday' }
y = int(input("year= "))
m = int(input("month= "))
d = int(input("day= "))
v=calendar.weekday(y,m,d)
print(dw[v])
```

Python Built-in Module (Python Random module)

Python Random Module



- Python Random module generates random numbers in Python. These are pseudo-random numbers means they are not truly random.
- Random module can be used to perform random actions such as generating random numbers, printing random a value for a list or string, etc.



Example: Write a Python program to generate random floats between 0.0 to 1.

```
import random
r1 = random.random()
print(r1)
```

Example: Write a Python program to generate random integers within specific ranges num1 to num2.

```
import random
x=int(input("num1= "))
y=int(input("num2= "))
r1 = random.randint(x, y)
print(r1)
```

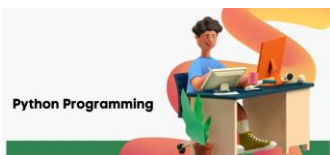
Example: Write a Python program to read list and displays random element from the list based on random module.

```
import random
z= []
n = int(input("Enter length of list:"))
for i in range(n):
    it = int(input("Enter item:"))
    z.append(it)
print(random.choice(z))
```

Example: Write a Python program to creates passwords of a desired length, consisting of lowercase characters a to z.

```
import random
def generate_password(length):
    characters = 'abcdefghijklmnopqrstuvwxyz'
    password = "".join(random.choices(characters, k=length))
    return password

desired_length = 8
password = generate_password(desired_length)
print(f"Generated password: {password}")
```



Example: Write a Python program to represents the game of Rock, Paper, Scissors, you can create a function that randomly selects one of the three choices (rock, paper, or scissors) for each turn. This function can be called **play_rps**, and it will allow a player to play the game against the computer. You could take an input for the player's choice and then randomly select the computer's choice. The function would then determine the winner based on the classic rules of Rock, Paper, Scissors.

```
import random
def play_rps(player_choice):
    choices = ['rock', 'paper', 'scissors']
    if player_choice not in choices:
        return "Invalid choice."
    computer_choice = random.choice(choices)

    if player_choice == computer_choice:
        return f"It's a tie! Both chose {player_choice}."
    elif (player_choice == 'rock' and computer_choice == 'scissors') or
         (player_choice == 'paper' and computer_choice == 'rock') or
         (player_choice == 'scissors' and computer_choice == 'paper'):
        return f"You win! You chose {player_choice} and the computer
chose {computer_choice}."
    else:
        return f"You lose! You chose {player_choice} and the computer
chose {computer_choice}."

player_choice = input("Please choose 'rock', 'paper', or 'scissors' ")
result = play_rps(player_choice)
print(result)
```

