Day 29

Local vs Global Variables in Python:

What is a Variable?

A variable is a name that stores a value in memory.

x = 10

What is Scope?

Scope defines where a variable can be accessed in your program.

Python mainly uses:

- Local scope
- Global scope

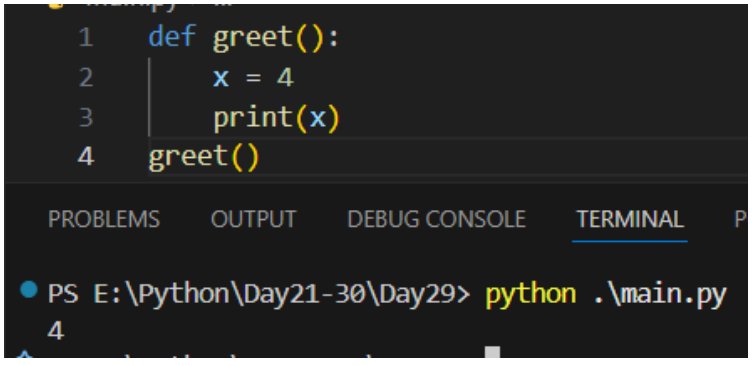
Local Variables: A local variable is created inside a function and can be used only within that function.

```
def my_function():
    x = 10   # local variable
    print(x)
my_function()
```

Global Variables: A global variable is created outside all functions and can be accessed anywhere in the program.

```
x = 20   # global variable
def my_function():
    print(x)
my_function()
print(x)
```

Example: local variable.



Example: global variable.

```
1   x = 4
2   def greet():
3       print(x)
4   greet()
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Python\Day21-30\Day29> python .\main.py
4
```

Example: local variable get preference over the global variable if there name be same.

```
1   x = 4
2   def greet():
3       x= 5
4       print(x)
5   greet()
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PO

PS E:\Python\Day21-30\Day29> python .\main.py
5
```

\

Example: using 'global' keyword

```
7   x = 10
8   def change():
9       global x
10      x = 20
11      print(x)
12  change()
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Python\Day21-30\Day29> python .\main.py
20
```

Why Avoid Too Many Global Variables?

Using many global variables can:

- Make code hard to debug
- Cause unexpected changes
- Reduce readability

Feature	Local Variable	Global Variable
Defined	Inside function	Outside function
Scope	Only inside function	Entire program
Memory	Created when function runs	Created at program start
Best practice	✅ Recommended	⚠️ Use carefully

--The End--