Python - Day 12 (Beginner)

Topic - Scope & Mini project

- Day 12 is less about the project and more about thinking correctly in python if you understand Day 12 well your bugs reduce drastically.

(*) What is Variable Scope? (Core Idea)

Scope = where a variable is accessible (visible) in your code

Think of Scope like rooms in a house:

- variable inside a room → Only usable in that room
- variable Outside → usable everywhere (unless blocked)

Python mainly has:

(1) local Scope
(2) Global Scope

(*) Local Scope (Very Important)

A variable created inside a function belongs ONLY to that function

```
def my function():
    x=10
    print(x)
my function()
print(x) # Error
```

Why this happens:

- x is created when the function runs
- It is destroyed after function ends
- Python literally forgets it exists

Key Rule

! local variables live only during function execution

Common Beginner Mistake

```
def increase():
    score = 10
increase()
print(score) # Name Error
```

fix → return the value

```
def increase():
    score = 10
    return score
score = increase()
print(score)
```

rule:-

"Functions should Return value, not modify things"

(*) Global Scope

variable defined outside all functions

```
player_health = 100
def damage():
    print(player_health)

damage() # works
```

Global variables:

- Accessible anywhere
- Dangerous if modified carelessly

(*) Global vs local with same Name (SHADOWING)

```
Score = 0   # global
def game():
    Score = 10 # local (shadows global)
    print(Score)
game()
print(Score)
```

Output

```
10
0
```

What happened?

- Inside function → python uses local scope
- Outside → python uses global scope

This is called variable shadowing

(*) Block Scope (IMPORTANT PYTHON Difference)

Python does NOT have block Scope
This is different from C/C++

```
if True:
    x = 5
print(x) # works in python
```

why?
- if, for, while do NOT create Scope
- Only functions create Scope

! Many bugs happen because beginners assume block scope exists.

(*) Namespaces (Under the Hood Concept)
A namespace is where python stores variable names

Types:
(1) local namespace - inside function
(2) Global namespace - file level
(3) Build-in namespace - python itself (print, len, etc)

How python searches variables (*)
local → Global → Build-in
if not found → NameError

(*) Constants in python
python has NO real constants, but we follow a Convention

```
PI = 3.14159
MAX_ATTEMPTS = 10
```

Why ALL CAPS?
- Tells humans: "Don't change this"
- Python won't stop you, but good developers respect it

⇒ Think of it like
Constants are a promise, not a rule

(*)(*) Mini Project - Number Guessing Game (*)(*)
This project exists to teach Scope and Control Flow, not randomness

Core Concepts used:
- random.randint()
- while loop
- if / elif / else
- Scope discipline
- function responsibility

One line Summary
(*) Day 12 teaches you how Python "thinks" about variables - mastering this saves you from 80% of beginner bugs.