Python - Day - 10 (Beginner)

Big picture of Day 10

Till Day 9:

- you used functions
- you passed inputs
- you printed outputs

Day 10 changes mindset:
Function should RETURN values, not just print them

(1) Why print() is NOT enough

Problem with print()
when you use print() inside a function

- The value is shown on screen
- BUT you cannot reuse it

Example

```
def add(a, b):
    print(a+b)

result = add(2, 3)
print(result)
```

// This looks fine, but:

Output

```
5
None
```

why None?
Because print() does not give anything back to the Caller

(2) what return actually does (VERY IMPORTANT)
what return means
return:
- Sends a value back to where the fun is called
- Ends the function immediately

Example:

```
def add(a,b):
    return a+b

result = add(3,2)
print(result)
```

Output
5

return = "Hey, take this value and use it Outside"

(3) Difference Between print and return

print()	return
• Shows output	• Sends Output back
• Cannot be reused	• Can be Stored & reused
• for debugging	• for logic
• User sees it	• Program uses it

{ Use print() for humans
Use return for programs }

(4) Storing Returned values

Returned values can be:

- Stored in variables
- passed into other functions
- Used in conditions

Example:

```
def Square(n):
    return n*n

x = Square(4)
y = Square(x)
print(y)
```

Output

256

(5) Multiple return Statments

A function can have more than one return, but only one runs.

Example:

```
def check_number(n):
    if n>0:
        return "positive"
    elif n<0:
        return "Negative"
    else:
        return "zero"
```

Once a return runs:
- function stops
- Code below it is ignored

Anything after return is dead code

(*)(*) Mini project - Calculator project (*)(*)
At the end of Day 10 you build a text based Calculator that:
- Asks the user for two numbers
- lets them choose an operation (+, -, *, /).
- Uses function return values to compute results
- Offers the option to continue calculations with the result or start fresh

Key takeaways from Day 10
- Returning values makes your functions much more useful
- Mapping symbols to functions shows the power of python dictionaries.
- you now know how to build a small interactive program that keeps running until the user chooses to quit.