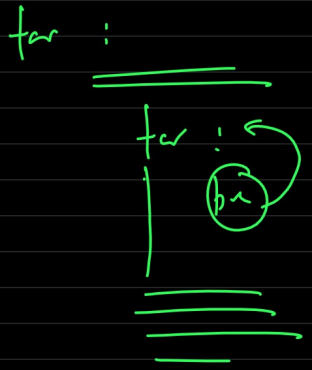
25-01-2026

Agenda:

- String methods
- while loop
- Functions
- Function arguments



String methods

```
"Hello".strip()
       .lstrip()
       .rstrip()
```

"Monal"
0 1 2 3 4

M → 0, -5
l → -1/4
nal → 2

-1 → not found

develope → standard → find('M') → 0, -5

loops:

for

→ iterate over a sequence of elements

ex. `for i in [1,2,3]:` → 3 iterations

→ we already know how many iteration loops is going to take.

while

→ repeat a block of code until certain condition is True

ex. `while condition:`

condition → understand python

```
while condition:
    jump to next
    topic
```

Syntax: while condition : → True

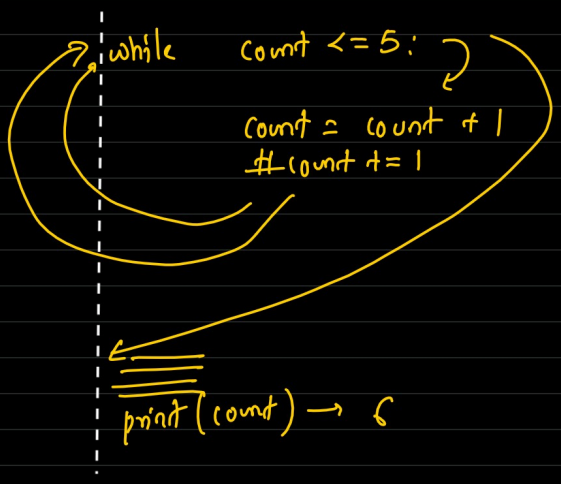
do something

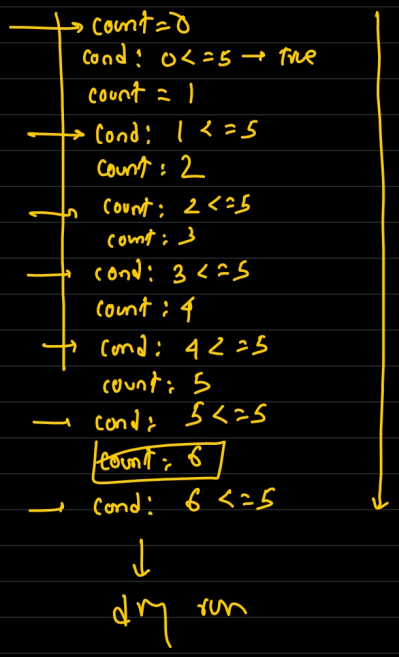
Example:

count = 0

keep adding +1 to the count until it is smaller than or equal to 5.

count <= 5



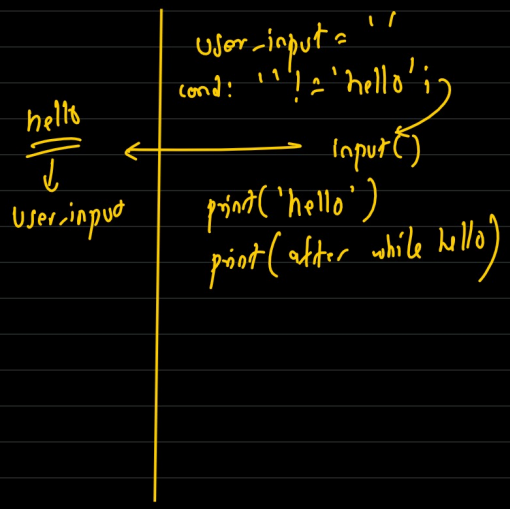


↓

dry run

hello != hello → false

```
user_input = ""
while user_input!="hello":
    user_input = input("Enter your input")
    print(user_input)
print("after while loop")
```



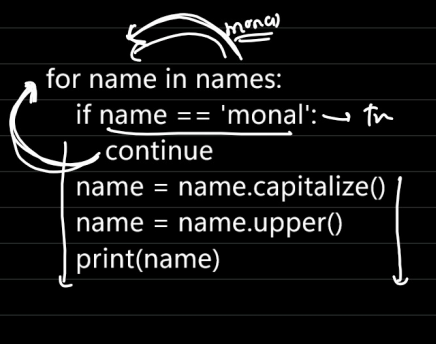
Control flow :

'mona'

100 values

for condition :

→ 100 times



'monal' ⟷

if 'a' == 'monal': → continue

→ something
→ something
→ something

```
for name in names:
    if name == 'monal':
        break
    name = name.capitalize()
    name = name.upper()
    print(name)
```

break → Terminates the loop

continue → skips the current iteration

'a' == 'monal' → run below

'b' == 'monal' → — " —

'monal' == 'monal' → True → break → come out of loop

Function :— is a named, reusable block of code,

→ used for a specific task.

Examples : print() (named)

: type()

print()
print()
} reusable

pre-defined function

print the output in terminal / screen.

User-defined function:

print X

Syntax:

def name_of_function(): ← parenthesis

do something

def print():

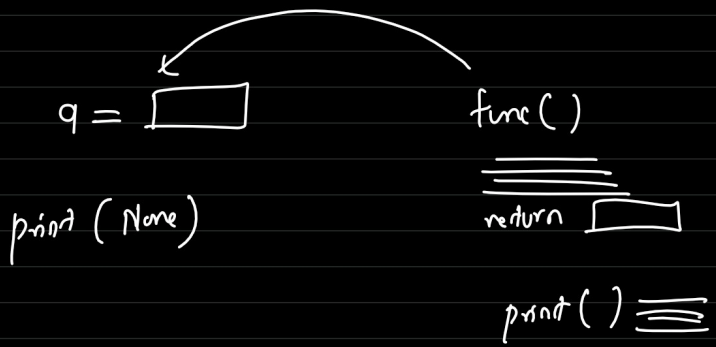
→ predefined function

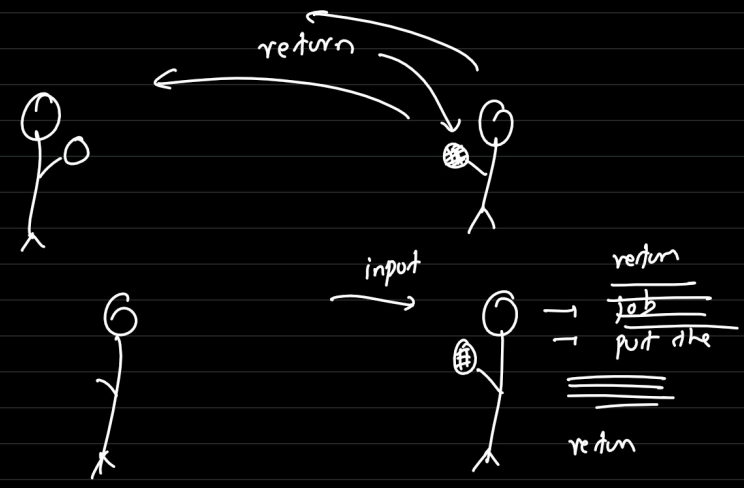
function → our_custom_string_cleaner → doing some job

print → clean version of string

→ cleaning the string inside it

p.) we want to get cleaned string.





print(3) → OS → 3

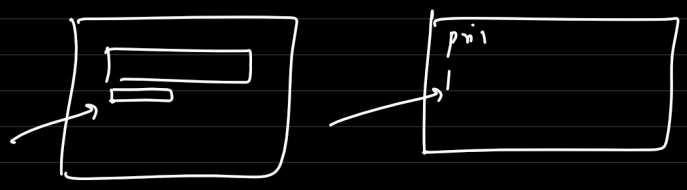
a = 3

a += 1

print(a) → 4

} see the output

Terminal



l = [200, 100, 2, 3]

sorted(l) → sort the given list & store the new sorted list in a variable

→ ~~return the new variable.~~

l = [200, 100, 2, 3]

l = sorted(l) None

l = [2, 3, 100, 200]

function → job
→ return None
→ return value

when we dont want updated value

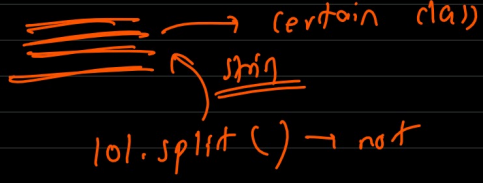
when we want update value

```
def add_dummy():
    name = input("Enter your name : ")
    print("hello", name, ", Greeting from krish naik academy.")

return_value = add_dummy()
print(return_value)
```

input "Hello ---- Ir

return value ← None ← add_dummy()



lol.split() → not

function → returns something → nothing → None
→ value →

P.1 we want to create reusable code that will take input from user a) a number & subtract. Once calculate return that value.

↓

Throw

↙

```
def subtract():
    a = float(input())
    b = float(input())
    c = a-b
```

val = subtract() → a=1, b=2
c = 1-2 = -1

↓

None

```
def subtract():
    a = float(input())
    b = float(input())
    c = a-b
    return c
```

val = subtract() → a=1, b=2
c = 1-2 = -1
→ return c

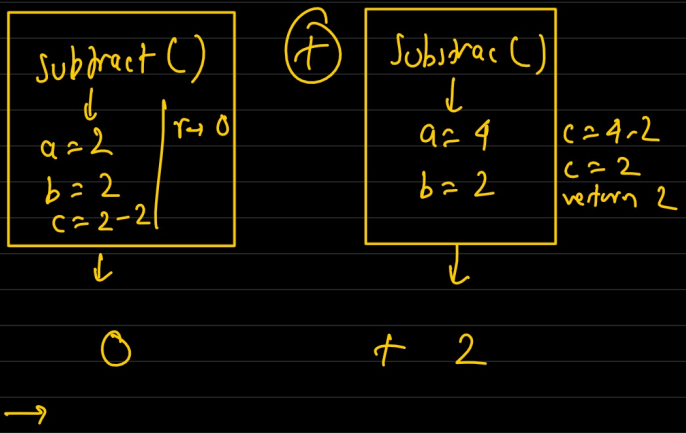
↓

-1

subtract()

top to bottom

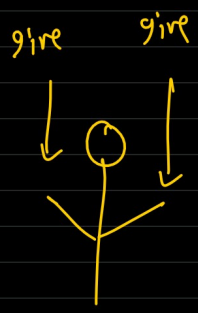
left to right →

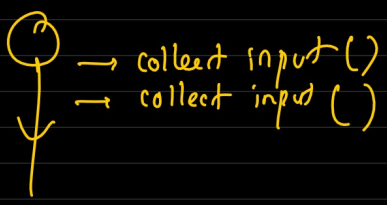


↓ ↓

0 + 2

→



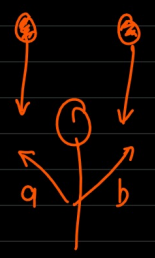



```
add(1,2)          def add(a,b):
                      print(a+b)
```

None

a=1
b=2

a+b

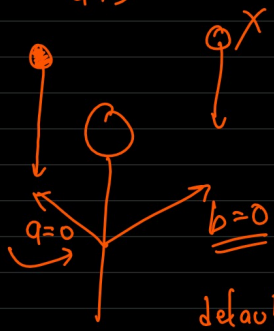


a, b → arguments of a function

a+b

( ) → arguments

subtract(1)



default arguments

default-argu
non-d

add(a=0,b) →

add(a, b=0)

add(a=0, b)

add(1)

add(a, b=0)

add(1)