

21-01-2026

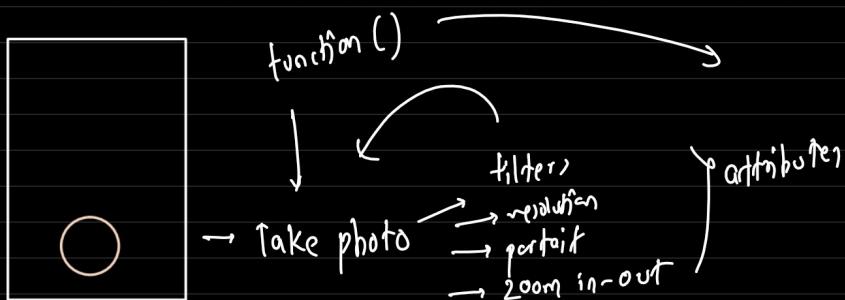
Agenda :

- input •
- output •
- control flow (advance) •
- loops

Output :

`print()` → used to display information on the console.

→ arguments



`print("hello_world")` → hello_world
`print("hello_world")` (\'\n')

'\n'
↓
new line

`print("hello_world")` → `print("hello_world", end='\\n')`

`print ("Hello")`
`print ("Hello")`



```
print("Hello", end=' ')
print("Hello")
```

HelloHello

```
print("Hello", end='..')
print("Hello")
```

Hello..Hello

input :

First name	<input type="text" value="Enter your name"/>
Final	<input type="text"/>
DOB	<input type="text"/> <input type="text"/>

name = → wait for user to type
 it does not respond to mouse click

name = "monal" string → it only responds to keyboard
 → stops taking input when ↵ pressed.

Enter your name_:

control flow (advance) (nested)

```
if      if      if      if      if:
else    if      elif    else    if
        if      elif    if     if
        else    if      else   else
            if
            else
```

if user_authenticated = True / False

if user :

User_role =

1/3 if user_role == "admin":
print("you have access to edit, view & delete")

2/3 elif user_role == "editor":
print("you have --- views & edit")

else: elif user_role == "viewer":
print("you have only view access")

Loop:-

- avoid repetitive writing
- efficient code
- works until a condition is met.

$$2 \times 1 = 2 \rightarrow \text{print}("2-", "x-", "1-", "=" - , "2")$$

$$2 \times 2 = 4$$

$$2 \times 3 = 6$$

$$\vdots$$

$$2 \times 10 = 2$$

for iterable :

grocery shop

apple → 1

iterate → repeat
something

bag(object) →



0

Task : count all thing in bag

⋮

cont one by one

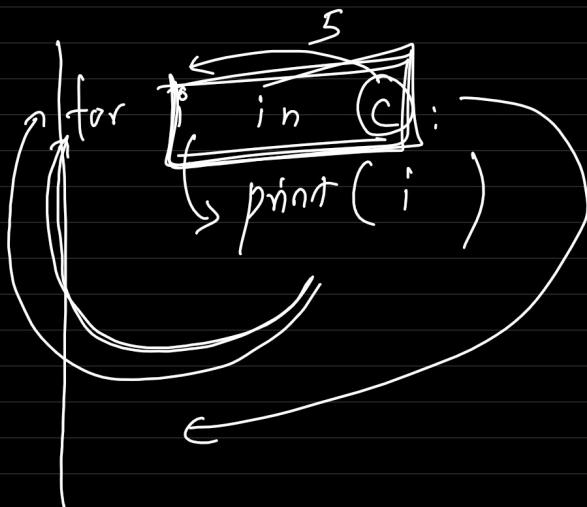
$$a=1, b=10^0, c = [1, 2, 3, 4]$$

$\downarrow \quad \downarrow \quad \downarrow$

X X iterable

$$c = [1, 2, 3, 4, 5]$$

1
2
3
4



i=1
 $\text{print}(1) \rightarrow 1$

i=2
 $\text{print}(2) \rightarrow 2$

i=5
 $\text{print}(5) \rightarrow 5$

`range(1, 100)` → numbers from 1 - 99

`range(1, 101)` → numbers from 1 - 100

`step_size = 1`

$[1, 2, 3, 4, \dots, 10]$

\nearrow
`step_size`

Diagram showing a list of integers from 1 to 10. Above each integer, there is a curved arrow pointing to the next integer, labeled '1'. Below the list, an upward-pointing arrow is labeled 'step_size'.

1, 3, 5, 7
↓
step-size = 2

range()
list, tuple, dict

for iterable:


```
for day in days_of_week:  
    if day == "wednesday":  
        print("send an email for booster_class")
```