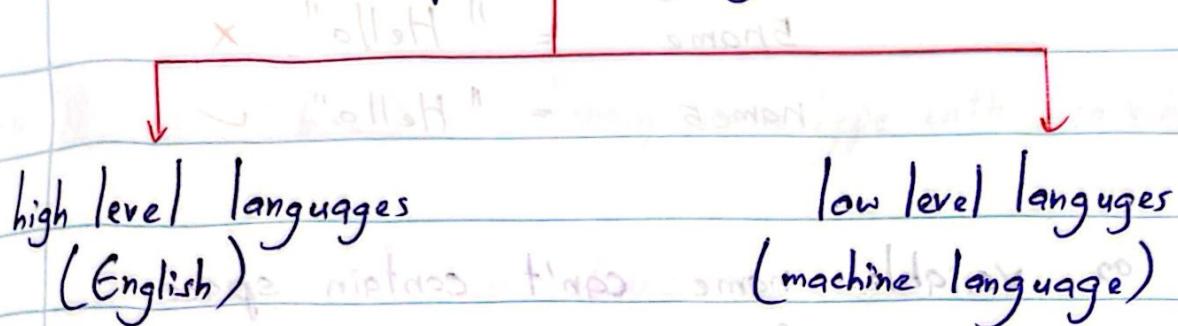


# Computer Languages



\* python      "Hello" = ~~DATA~~ text

\* java      "Hello" = ~~DATA~~ text

\* pascal

• Interpreter      language

• Compiler      translator

→

## Python Variables

x = 7i

x = 10 ← assigned value.

variable name      assigning operator

num1 = 100 → (it) ~~variable~~

correct = true

writing → (it) ~~variable~~

01. Variable name can't start with numbers

5name = "Hello" X

name5 = "Hello" ✓

02. Variable name can't contain space.

first name = "ABC" X

firstname = "ABC" ✓

03. variable name can't contain special characters.

(%, #, ? ...) but only under score allow (-)

name@ = "ABC" X

04. variable name can't use python key words.

(if, else, pass, while, continue, break)

if = 123 X

below described → if = X

## Data types in python.

01. strings (str) - sequence of character

Enclose with '' / " "

02. Integer (int) - whole number (positive / negative)  
without decimal number.

03. Float (float) - a number with decimal.

04. Boolean (bool) - binary data type with two value

## Python Function.

o) pre defined function. (x) taking

## 02. User defined function.

## 01. Pre defined function.

01. Simple letters. A ଅଙ୍ଗେ କାର୍ତ୍ତିକୀ ଶର୍ମିଷ୍ଠାନି

02. () → print()

type ()

len ()

input ()

01. Print

to display any kind of output.

- `print ("Hello")`
- $y = 10$   
`print (y * 2)`

- $x = "Hello"$

`print (x)` without  $y = "10"$

, without print  $y * 2 \rightarrow$  output  $1010$

02. type

to check any type of variable.

Ex:

$x = 10$

`type (x)`

*Output of this will be int, because display function print function return type of variable.*

`print (type (x))`

$x = 10$

$y = type (x)$

`print (y)`

03. len

to find the length of any variable.

 $x = "Hello"$ 
 $y = \text{len}(x)$ 
 $\text{print}(y) \checkmark$ 
 $x = 10$ 
 $y = \text{len}(x)$ 
 $\text{print}(y) \times$ 

*len නිසා නියම සහුව නො යොමු කළ ඇති*

අපේක්ෂණ ත්‍රීඩ් නිසා නො යොමු කළ ඇති. float,

integer හෝ.

( ) නිසා නො යොමු කළ ඇති

X නිසා නො යොමු කළ ඇති

+ නිසා නො යොමු කළ ඇති

04. input.

to get an input from the user.

 $x = \text{input}("Enter the number")$ 
 $\text{print}(x)$ 

- input function සහ user සඳහා පෙනෙන නියම මූල්‍ය නිසා නො යොමු කළ ඇති.

 $\text{print}(x * 2)$

# Python Operators.

## 01. Arithmetic operator

### 02. Comparison

### 03. logical

"Hello" = x

(x) not = p

x (p) thing

v (v) thing

## Arithmetic operator.

- brackets ()

- Division /

- addition +

- modulo %

- subtraction -

- exponentiation \*\*

- multiplication \* thing as top of

\*  $10 / 4 = 2.5$  and  $10 / 5 = 2$ . thing = x

division operator නො නැත්තුවන් තුළු මෙයින් නො නැත්තුවන් නො නැත්තුවන් නො නැත්තුවන් නො නැත්තුවන්

එහින් float සඳහා පිහිටුව.

ලේඛන කිරීමෙහි නො නැත්තුවන් නො නැත්තුවන්

\* modulo → උක්කය.

$10 \% 4 = 2 \rightarrow$  නො නැත්තුවන් නො නැත්තුවන් නො නැත්තුවන්

(%" ) නො නැත්තුවන් නො නැත්තුවන් නො නැත්තුවන් නො නැත්තුවන් නො නැත්තුවන්

(% x) = thing

Addition, subtraction, multiplication, division, less than or equal to, greater than or equal to, less than, greater than, equals, not equals.

## Operator precedence

01. ()

02. \*\*

03. \*, /, %, //

04. + -

$$\text{Ex: } 5 * 2 - 3 * * 2 = \underline{\underline{1}}$$

5 \* 2 % 2 + 3      Same precedence නියම සඳහා solve  
විද්‍යා මූල්‍ය පෙනීමෙන් අනුව  
left to right නියම යොමු කළයායි

$$= 10 \% 2 + 3$$

$$= 0 + 3 = \underline{\underline{3}}$$

( )

## Comparison operators

\* Give output as True or False.

• > greater than     $5 > 2$  - True.

• < less than.

•  $\geq$  greater than or equals

•  $\leq$  less than or equals

• == equals

• != not equal

number  
numbers  
දැන භාවිත කිරීම  
විද්‍යා මූල්‍ය පෙනීමෙන්

String වලට use කළයායි.  
Atlas

\*  $X = 3$  ~~is~~ ~~an~~ equal sign we ~~are~~ ~~using~~ variable  
the value ~~is~~ assign ~~to~~

\* "Hello" == "Hello" → True. ( ) 10

## Logical Operators

- and
  - or | = C A A
  - not → not | → 0

## Operator precedence in Python.

01. ()
  02. \*\*
  03. \*, /, %, , //
  04. +, - to shift to right side
  05. ==, !=, >, <, >=, <=
  06. not - shift to left side ( )
  07. and - shift to left side ( )
  08. or - always shift to left side ( )

## Python Index

`y = "Hello - world"`

- \* ගෙන සඳහා type සංඛ්‍යාව ප්‍රින්ට(y) සංඛ්‍යා.
- \* ගෙන string හිසේ letters සඳහා සංඛ්‍යා type සංඛ්‍යා නො නැත python index use සංඛ්‍යා වා.

01. Positive index → sentence වෙති තුළ ඉතුළු සංඛ්‍යා නො නැති.

`Hello - World`

positive index. ↓↓↓↓↓ ↓ ↓↓↓↓  
 start සියලු නො නැති. 0 1 2 3 4 5 6 7 8 9 10

02. Negative index

`Hello - World`

↓↓↓↓↓ ↓ ↓↓↓↓  
 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1

negative index start සියලු.

Ex:

`print(y[8])`

`print(y[-3])` → r

`print(y[4])`

`print(y[-7])` → o

\* Sentence නොවා සඳහා part වෙතින් print කළ ලද නොවා මෙය ඇත. Hello වෙත සියලු word වෙත වෙතින් print නොවා මෙය ඇත නේ format වෙත we සඳහා ඇත . range වෙතින් print සඳහා format වෙත.

`print(y [ starting : stop index + 1 ])`

Ex: `print(y [ 0 : 5 ])` → output, Hello

`print(y [ 1 : 5 ])` → ello

`print(y [ -10 : -6 ])` → ello

`print(y [ -11 : -6 ])` → Hello

\* letter වෙතින් තුළු හරිනුවෙන් ඇත්තේ තම print කළ ලද නොවා සිරසා මෙය සඳහා use සඳහා ප්‍රතික්‍රියා කිරීම් ප්‍රතික්‍රියා කිරීම්

`print(y [ starting : stop + 1 : Step ])`

letter එක නැත් දැක්වන නේ stop index 2 සහ 5  
ගැනීම් නැත් 3 , ගැනීම් නැත් 4

Step  
Index  
sign Date: + → ජා ඕස්  
+ → ගැ ඕස්

Ex: `print(y[0:11:2])` → Howard

මධ්‍ය වෙත type යෙනි තුළ ඇත්තේ type  
`print(y[4: : -1])` → alleH

python index යෙනි පිටපත නම් තුළදී left to right. (අඟඹවා  
letters type යෙනි තුළ එහි නම් step (index use නොවා) -1  
(minus use නොනො එහි)

`print(y[10: : -1])` → drowolleH  
(එම්බරස් sentence මූලය)  
from "drowolleH" = [0] to [9]

`print(y[10: : -2])` → drWolH  
(letters නොවා ඇත්තේ පෙන්වනු ලබයි)

## Python If condition. if...else

if (condition):

Intendence  
@define

if (True):  
 body

else ("ans"):

print("ans")

condition: True condition do true block.

False

condition if not body do

other block.

print("ans")

Ex:  $x = 2$

if  $x > 1$  : (( $x > 1$ ) p) true

print ( $x$ )

Hello ← (( $x > 1$ ) p) true

- Q1) Write a python code to print any name if it is starting with "a". prompt the user to insert the name.

Hello name = input ("Enter any name")

if name [0] == "a" :

print (name)

else :

print ("error")

- Q2) Write a python code to check any number is odd or even. prompt the user to input number.

$x = \text{int}(\text{input} ("Enter number"))$

$y = x \% 2$

if  $y == 0$  :

print ("even")

else :

print ("odd")

03) write python code length of any name which is input by the user and if it's length greater than output → yes else → no.

```
x = input ("Enter name")
```

```
y = len(x)
```

```
if y > 10:
```

```
    print ("yes")
```

```
else:
```

```
    print ("no")
```

$\hookrightarrow$  ( $[c]$ o) thing

04) write a python code to print the followings for any number which is input by user.

$> 10 \rightarrow$  yes

$= 10 \rightarrow$  equals

$< 10 \rightarrow$  no ( $[c]$ o) thing

$\hookrightarrow$  ( $[c][e]$ o) thing

```
x = int(input ("Enter number"))
```

```
if "x > 10": ( $[c]$ o) thing
```

```
    print ("Yes")
```

(elif)  $\rightarrow$  elseif  $x == 10$ : ( $[c]$ o) thing

python code  $\rightarrow$  print ("Equals")

else:

$\rightarrow$  print ("No") Atlas

## List (data type)

data set සාක්ෂි. list සාක්ෂි නොමැත්තු නොමැත්තු නොමැත්තු නොමැත්තු

සෑම සාක්ෂි මෙයින් ප්‍රතිඵලි කළ යුතු වේ. list සාක්ෂි සැවැස්සු සැවැස්සු square bracket ඇතුළු. [ ]

Ex:

$$y = [1]$$

$z = [] \rightarrow$  empty list

$$x = ["a", "b", "c", 1, 2, 3]$$

0 1 2 3 4 5

print ( $x[2]$ )  $\rightarrow c$

$$y = [1, 2, 3, "Hello", 5, 6, 7]$$

0 1 2 3 4 5 6

print ( $y[3]$ )  $\rightarrow$  Hello

print ( $y[3][1]$ )  $\rightarrow$  e

$$y = [1, 2, 3, ["a", "b", "c"], "Hello"]$$

print ( $y[3][1]$ )  $\rightarrow$  b

("a") string

list සාක්ෂි

ගැනීම

sub list

සැවැස්සු

("a") string

("a") string

No: \_\_\_\_\_

Date: \_\_\_\_\_

## (soft states) draft

and definitions of storage states for A

as well as some intermediate states to store

etc counts

$\{(\epsilon, 1), \epsilon, 2, 1, 2, 3, 1, 2, 3, 4, 1, 2, 3, 4, 1\} = R$

$\{1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1\}$

$\{1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1\}$

$\{1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1\}$

possible

$\{\text{"out": } \epsilon, \text{"in": } 1\} = R$

↑ ↑

subv. pos

very often

$\{\text{"out": } \epsilon, \text{"in": } 1\} = R$

② we put CFS in external SFS possible \*

before when we subv.

$\text{fut A} \rightarrow (\{\text{"open"}\}) \cup \{\text{fut}\}$

Atlas

## Tuple (data type)

A set of data separate by comma which is  
 data set එකක් සිංහල තේව. එකෙනු නො ඇත්තායි  
 කරගන්න තේ.

Ex:

$$x = ["a", "b", "c", 1, 2, 3, [1, 2, 3]]$$

↓      ↓      ↓      ↓      ↓      ↓      ↓  
 0      1      2      3      4      5      6

print (x [6] [2]) → 3

\* comma එකක් නො  
 separate  
 ගෙවනු ලබ.

## Dictionary

$$x = \{ 1 : "one", 2 : "two" \}$$

↑      ↑  
 key      value.

key value pair

$$y = \{ "mango" : "A fruit", "pumpkin" : "A veg." \}$$

\* dictionary නො index වේ. අමේ key නො ඇත්තා  
 value නො ගැනී යුතුයි.

print (y ("mango")) → A Fruit.

## Set

- \* curly bracket {} അളവു നിഃവിശ്വസ്ത ഡാറ്റാ സെറ്റ്.

$x = \{ 1, 2, 3, 4, "Hello" \}$

- \* set നിഃവിശ്വസ്ത ഡാറ്റാ ടൈപ്പ് ഇന്റീസ് എന്നും സെറ്റ്.

## Arrays

- \* ഡാറ്റാ ടൈപ്പ് ഓഫ് തിരുത്തു ഡാറ്റാ സെറ്റ് നിഃവിശ്വസ്ത്.

$x = \text{array} ("i", [2, 4, 6, 8])$

'integer' ഇന്റെ ഒരു അടിസ്ഥാനം.

$y = \text{array} ("f", [3.5, 2.1, 7.4])$  float type data.

- \* index നിയന്ത്രണ ഡാറ്റാ.

`print(x[0])`

$\Rightarrow "2"$  ) ഫോർമ്മാറ്റ്.

`print(y[1])`

$\Rightarrow 2.1 = x$

{ from array import array }

# Loops

while  
for

\* බොහෝතර තුළයේ මාධ්‍ය ආකෘති සංස්කීර්ණ නේ loops යුතු නො ප්‍රතිඵලිත යුතුයි.

if condition නොවූ ඇති  
ගුණීන් විභාග මෙහෙයුම්.  
නොවූ loop සංස්කීර්ණ නොවූ.

## while

while [ condition ] :

{ body . }

infinite loop :  $x = \text{stab}$  වැදගිත් - පෙන්වනු ලබයි \*

while  $x \leq 5$  :

Infinite loop

print ("Yes") ;

\* loop සංස්කීර්ණ නොවූ නොවූ condition නොවූ false

print ("Yes") ;

$x = 1$

1

Yes

while  $x \leq 5$  :

2

Yes

print ("Yes") ;

([0]) x 3 failing

Yes

$x = x + 1$

([1]) x 4 failing

Yes

5

Yes

([2]) x 6 failing

6

Yes

Q1) Write a python loop to print any name 10 times  
prompt the user to input name.

y = input ("Enter name")

x = 1

while x <= 10 :

print (y)

x = x + 1

Q2) Write a python code to print numbers 1 to 5.

x = 1

while x <= 5 :

print (x)

x = x + 1

Q3) Write a python code to print even numbers from

1 to 10

x = 2

while x <= 10 :

print (x)

if x % 2 == 0 :

x = x + 2

print (x)

x = x + 1

Atlas

# Mutable and Immutable Data

- \* Mutable data → data types that can be modified added or removed after creating

Ex: list, tuple, sets, dictionary, array

- \* Immutable data → after creating can't change.

Ex: int, float, tuple, str

•  $x = [1, 2, 3, 4, "Hello", (1.0, 2.0, 3.0)]$

(is) tuple

- Accessing elements in list.

print(x[4]) → Hello

- Modifying elements in list or not

$i = x[3] = 5 \leftarrow \text{change value } x$

print(x)

(x) tuple

: o1 => x

(x) tuple

print(len(x))

Atlas

### • Adding elements

```
x. append ("World")
print (x)
```

append command නො use

විටත් element නො add  
විටත් සාංස්කීර්ණයෙන් list  
නො ඇතුළත.

### • Removing elements

```
x. remove ("World")
print (x)
```

Pop නොදැක්වාගා  
නො list නොදැක්වා  
index number  
නො.

```
x. insert (1, "World")
```

insert නොනැවා කෙට  
Index නොනැවා value  
නොනැවා add විටත්කා  
වුතුවා.

```
print (x)
```

```
y = x. count ("Hello")
print (y)
```

list නොනැවා elements නො
 නො නොනැවා count නොනැවා
 නො නො command නො
 use නොනැවා.

### • Assending order.

```
x = [4, 5, 8, 70, 1, 5, 2]
```

```
x. sort ()
```

```
print (x)
```

sort නිවාන එකා
 use නොනැවා මුදලන්
 list නොනැවා තිබාව
 data type නොනැවා
 තිබාවන බැව.

### • x.reverse ()

```
print (x)
```

list නොනැවා element reverse

එක්වම type නොනැවා විභාග නැවා

ගෙවා ප්‍රේෂ්‍ය නොනැවා,

• `x.clear()`

list ~~বিবরণ করে~~ values

`print(x) → []`

remove ~~বিবরণ করে~~ values  
Empty list ~~বিবরণ করে~~

• `x = ("Hello", "Yellow", "Allow")`

list ~~বিবরণ করে~~ extend

`y = [1, 2, 3, 4]`

use ~~বিবরণ করে~~

`x.extend(y)`

"allow" ~~বিবরণ করে~~ + x

`print(x)`

(x) ~~বিবরণ করে~~

## ০২. Array

`from array import array`

`x = array("i", [1, 2, 3, 4, 5, 6, 7, 8, 9, 10])`

`x.append(4)`

`x.insert(1, 15)`

`print(x)`

`print(x)`

`x.remove(15)`

`(1) + x.pop(1)`

`print(x)`

`print(x)`

# slicing an array

array ~~বিবরণ করে~~ data এর

`y = x[0:3]`

type ~~বিবরণ করে~~ করা না

`print(y)`

Index use ~~বিবরণ করে~~ slicing  
command ~~বিবরণ করে~~

{  
Ten v count v sort v reverse v clear x  
extend v }

### 03. Tuple .

data set change command tuple

ଶ୍ରୀ କଣ୍ଠାନାଥ ମାତ୍ର ।

count  $\vee$  slicing  $\vdash \text{len}(\text{v}_2, \text{x}) = 3$

Birds enjoy the fresh air and water

$$x = 1, 2, 3, 4, 5$$

concatenation.

$$y = \underline{6, 7, 8, 9}$$

$$z = x + y \rightarrow \text{print}(z)$$

\* repetition → print( $x * 2$ ) → (1, 2, 3, 4, 5, 1, 2, 3, 4, 5)

(p) ~~no solution~~  $x = 8$

\* membership test → tuple သို့ မဟုတ် အကြောင်း အကြောင်း data value

list නිවෙස්  
ok

சிறை check முடியவில்லை.

$\text{Ex: } \text{print}(\text{3 in } x) \rightarrow \text{True}$

$\{e_i\}$   $\leftarrow$  ( $e$ ) fair

Nesting → tuple ဆုတေသန၏ အားလုံး ၁၃၁၂ မှာ tuple ဆုတေသန.

$$x = ((1, 2, 3), (4, 5, 6))$$

$\{e_1, e_2\} \leftarrow (\epsilon)$  thing

Note: \_\_\_\_\_

## 04 Set.

Some about operations we can apply on sets

$$x = \{1, 2, 3, 4, 5, 6\}$$

$$y = \{2, 5, 4, 8, 6\}$$

Append x insert x pop x len ✓

remove ✓

$$\{1, 2, 3, 4, 5, 6, 8\} = x$$

$$z = x \cup y$$

$$\text{print}(z) \rightarrow \{1, 2, 3, 4, 5, 6, 8\}$$

$$(1, 2, 3, 4, 5, 6, 8, 1) \leftarrow (c+x)$$

$$z = x \cap y$$

$$\text{print}(z) \rightarrow \{2, 4, 5, 6\}$$

do this for intersection

$$z = x - y$$

$$\text{print}(z) \rightarrow \{1, 3\}$$

Some about set operations like symmetric difference

$$z = x \Delta y$$

$$\text{print}(z) \rightarrow \{1, 3, 8\}$$

## 05. Dictionary

`x = { 1: "One", 2: "Two", 3: "Three" }`

(i) trying :  $x \rightarrow \{1: "One", 2: "Two", 3: "Three"\}$

Adding a new key pair

`x[4] = "Four"`  $\rightarrow x = \{1: "One", 2: "Two", 3: "Three", 4: "Four"\}$

`print(x)`

`y = x.get("Mango")`

`print(y)`  $\rightarrow$  A Fruit.

`x["Mango"] = "a fruit"`

`print(x)`

For loops.

`y = input("Enter any name")` } `y = input("Enter name")`

`X = 0`

`while X < 5 :`

`print(y)`

`X = X + 1`

`for i in range(0, 5):`  
 `print(y)`

i ആ ഫലങ്ങൾ തിരുത്താൻ കോഡ് ആണ്.  
run ചെയ്യാം.

Atlas അഭിരാജ്

Write a python code to print 1 to 5

```
x = 1
for i in range(1, 6):
    print(i)
    while x <= 5:
```

print(x)

x = x + 1

```
for i in range(15, 20):
    print(x)
```

(<sup>"spam"</sup>) x = x + 1

True → (<sup>v</sup>) False

Write to python code to print 1 to 100 even

"True" = [<sup>"spam"</sup>] x

x = 0

(for i in range,

while x <= 100 :

(0, 101, 2)

print(x+2)

print(i) <sup>step number</sup>

x = x + 2 /

"True" = <sup>v</sup> if (<sup>"and pass statement"</sup> if x % 2 == 0 :

x = 0

print(i)

: (x, 0) while x <= 100 :

: x > x - 1

(if x % 2 == 0

(<sup>v</sup>) True

as step 2nd : print(x)

i + x = x

x + 1

write a python code to get square numbers  
for i in range (1, 101):

$$y = i * i$$

if  $y < 100$ :

print(y)

Factorial value of 5  $\rightarrow 5 * 4 * 3 * 2 * 1$

x = input(int("Enter number"))

y = 1

while x > 0:

$$y = y * x$$

x = x - 1

print("Factorial value =", y)

x = [1, 2, "Hello"]

for i in x  $\rightarrow$  output x as list

print(x)  $\rightarrow$  output each element, print each.

: suit slides

("press to exit suit slide")

Atlas

## Infinite loops.

loop will run till condition is true.  
infinite loop use while True.

**while True :** (Condition will be True till user stops running the loop)

**Ex:** 01. Write a python code to print numbers given by the user if the user input -1 loop should stop.

**while True :**

x = int(input("Enter number"))

if x = -1 } user input -1 loop

(break) stop running the loop

print(x)

02. For egg seller he needs to find number of eggs sold within a day in dozens for a given period.

**while True :**

egg = int(input("Enter the no of eggs"))  
Atlas for the day

i if egg = -1  
break

date = input ("input the date")

z = egg // 12

y = egg % 12

print ("No of eggs sold in date", date)

print ("No of eggs in dozens", z)

print ("Remaining no of eggs", y)

## User defined Function.

def function name ( ) :

                  
} body .

- \* Function name ~~is~~ python keywords නොවා ඇත.  
name සඳහා නිව්වා යුතු කළ යුතු.

Ex:

```
x = int(input(" number 01"))
```

```
y = int(input(" number 02"))
```

```
def total (a,b):
```

```
    print (a+b)
```

```
total (x,y)
```

• x get value from a local argument and assign to it.

• y get value from b local argument and assign to it.

arguments      argument pass by value

user defined function also run like a normal function

Method Function also call method, function or function. But  
variables x & y variable @ values of function  
also assign.

```
x = int(input(" Enter number 01"))
```

```
y = int(input(" number 02"))
```

```
def total (a,b):
```

return a+b → return value from function

```
print (total (x,y))
```

return value from function  
"return" → return value from function.

\* return → return value from function

function return value from function

function return value from function

\* Function കുറഞ്ഞ കോണിൽ വരുത്തിയാൽ  
assign ചെയ്യുന്നത്.  
Ex:  $t = \text{total}(x, y)$   
 $\text{print}(t)$

### default parameter.

```
x = int(input("num o1"))
```

```
y = int(input("num o2"))
```

```
def total(a, b=10):
```

```
    return a+b
```

Function കുറഞ്ഞ കോണിൽ x വരുത്തിയാൽ

$t = \text{total}(x) \rightarrow$  ദിവസിലും ഒരു ഫലാവല്ലിക്ക് x എം വല്ല കുറഞ്ഞ.

```
print(t)
```

output = x value + 10 ← default value.

y വരുത്തിയാൽ കുറഞ്ഞ കോണിൽ y

value കുറഞ്ഞ കോണിൽ.

\* Function കുറഞ്ഞ കോണിൽ arguments കുറഞ്ഞത് അല്ലാൽ  
default value use ചെയ്യാം.

```
def total(a=10, b=20):
```

```
    return a+b
```

$t = \text{total}()$  → Function കുറഞ്ഞ കോണിൽ argument കുറഞ്ഞത്.

default value കുറഞ്ഞത് use ചെയ്യാം.

01) Write a function to calculate the area of rectangle. The function should take length and width as parameters and return the area.

```
l = int(input("Enter the length"))
```

```
w = int(input("Enter the width"))
```

```
def arearec(a, b):
```

```
    return a * b
```

```
print(arearec(l, w))
```

((Even)) True  
((Odd)) False

02) Create a function that checks if given number is even. The function should return True, if the number is even. ~ (e) False ~ f

```
x = int(input("Enter number"))
```

```
def even(a):
```

```
    if a % 2 == 0:
```

```
        return "True"
```

else: False

```
return "False"
```

```
print(even(x))
```

d + p marks

- If you type space then result is ( ) lstat = +

because sAtlas

# Modular Programming.

Modules	
Internal (built-in)	External
* math	(pandas) 20
* datetime	* numpy
* random	* request.
* os	

Math (For mathematical calculation)

```
import math.  
x = math.sqrt(16)  
print(x)
```

$x = \text{math. pow}(2, 3) \rightarrow 2^3$  power.  
print(x)  $\rightarrow 8.$

Random (To generate random numbers)

```
import random  
x = random.randint(1, 10)  
print(x)
```

Atlas

## Datetime (get date)

```
import datetime
```

```
today = datetime.date.today()
```

```
print(today)
```

## Os (File handling)

```
import os
```

```
os.mkdir("My new folder")
```

```
current_directory = os.getcwd()
```

```
print(current_directory)
```

(current directory position not ) item

item (program)

(y) type: file = x

(x) func

func = (x,y) func = x

x = (x) func

(random number generator of )