Introduction to Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics.

Firstly, let's begin with the print command!

Print Command

print()

> To display output or results, Python provides the print() function. This function can be used in different formats!

1. print("I'm learning Python")
2. print('I'm learning python")
3. print(""I'm learning Python"")
4. print(" I'm learning python and \n this is a new line")

```
print ("I am learning Python")
```

```
print("I'm learning Python")
```

```
print( ""I'm learning Python"")
```

```
print(" I'm learning python and \n this is a new line")
```

```
print("\"I\'m learning python\"")
```

```
print ('*'*15)
```

Variables

A variable allows you to store values by asigning it to a name which can be used to refer to the value later in the program.

```
#let
x = 15
y, z = 20, 34
python = "It is easy"
is_published = True #boolean
print(x,y,z)
print(python)
```

```
print (x,y,z, sep =',')
```

```
x = 26
print(x)
```

```
y = y+1
print(y)
```

Exercise 1

We check in a student named Vaibhav Trivedi. He is 21 years old and is enrolled for the workshop.

```
full_name =  'Vaibhav Trivedi'
age = 21
enrolled = True
```

Input Command

input()

To get input from the user, we use input function. This function takes a value from the user and returns it as a string.

```
name = input('Enter name ')
print('Hi ' + name + ' Welcome to the Python Workshop')
```

Exercise 2

Ask input of two questions: 1) Person's name (For example: Suresh)

2) Person's age (For example: 21)

And print: (Person's name) is (Person's age) years old.

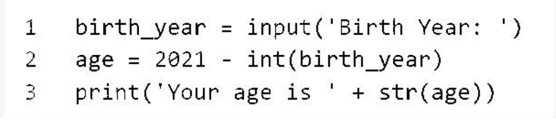
(For example: Suresh is 21 years old.)

```
person_name = input('Enter your name: ')
person_age = input('Enter your age: ')
print(person_name + ' is ' + person_age + ' years old')
```

Exercise 3

Write a code to calculate your age.

```
birth_year = input('Birth Year: ')
```



```python
birth_year = input('Birth Year: ')
age = 2021 - int(birth_year)
print('Your age is ' + str(age))
```

Exercise 4

Convert weight from pounds to kg and print the output.

```python
weight_lbs = input('Weight (lbs): ')
weight_kg = int(weight_lbs) * 0.45
print('Weight in kg is '+ str(weight_kg))
```

Extracting the String

```python
string = ''' Python



Workshop'''
print(string)

```

```python
whatever = '''We are learning Python'''
#             0123456789.....
print(whatever[4])
print(whatever[-4])
print(whatever[-2])
print(whatever[0:9])
print(whatever[0:])
print(whatever[1:])
print(whatever[:])
```

Exercise 5

Let

name = 'Jahaan'

print(name[1:-1])

Compute the output

```python
name = 'Jahaan'
print(name[1:-1])
```

Exercise 6

Take 2 inputs

1) First Name

2) Last Name

Print: First Name [Last Name] is a physics student

Example

First Name = Bhavya

Last Name = Thacker

Output: Bhavya [Thacker] is a physics student

```
first = input('First Name: ')
last = input('Last name: ')
output = first + ' [' + last + '] is a physics student'
output1 = f'{first} [{last}] is a physics student' #formatted string
print(output)
print(output1)
```

Double-click (or enter) to edit

String Methods

len() : calculates the number of characters in a string

```
count = 'I am learning Python'
print(len(count))
```

"." operator

```
print(count.upper())
print(count.lower())
print(count.find('P'))
print(count.replace('learning', 'Studying'))
print('Python' in count)
```

Arithmatic Operators

Addition (+)

Subtraction (-)

Division (/)

Multiplication (*)

To get integer in division (//)

To get remainder of division (%)

Exponent (**)

Augmented Assignment Operator (+-=)

```
a = 10 + 5
b = 16 - 6
c = 10 / 3
d = 2 * 2
e = 10 // 3
f = 10 % 3
g = 4**2
print(a, b, c, d, e, f, g, sep=', ')
h = 10 + 3 * 2 ** 2
print(h)
i = (10 + 3) * 2 ** 2
print (i)
```

Other useful functions

round() : rounds off the digits

abs() : It always returns a positive number. (Absolute)

Double-click (or enter) to edit

Operations

Python Operator	Description
+	addition
-	subtraction
*	multiplication
/	division
**	power

Addition:

As we have seen earlier, we can add two numbers simply by using '+' operator.

```
a = 10
b = 79
sum = a + b
print(sum) #=> 89
```

Examples

```
# Integers:
a = 12
b = 16
result_int = a + b

# Floating numbers:
e = 271e0
h = 6.63e0
result_float = e + h

# Imaginary numbers:
k = 10 + 12j
j = 15 + 34j
result_im = k + j

print(f"Integer sum: {result_int}\nFloat sum: {result_float}\nImaginary sum: {result_
```

Subtraction:

We use the '`-`' operator. Let *a* and *b* be any given number.

```
result = a - b
```

Examples

```
# Integers:
a = 12
b = 16
result_int = a - b

# Floating numbers:
e = 271e0
h = 6.63e0
result_float = e - h

# Imaginary numbers:
k = 10 + 12j
j = 15 + 34j
result_im = k - j

print(f"Integer result: {result_int}\nFloat result: {result_float}\nImaginary result:
```

Multiplication:

We use the '`*`' to multiply

[ ] ↳ *1 cell hidden*

Division:

There are two main types of division operators, "`/`" and "`//`".

- '`/`' is used for normal division.
- '`//`' is used for integer type division (rounded-down method).

Example:

```python
# Defining Variables:
a = 73
b = 4
# Normal division
print(f"Normal division: a/b = {a/b}")

# Integer division
print(f"Integer division: a//b = {a//b}")
```

```python
#Example 2:
a = -19
b = 3
# Normal division
print(f"Normal division: a/b = {a/b}")

# Integer division
print(f"Integer division: a//b = {a//b}")
```

Integer division gives us the closest lower integer from the actual division.

In case of complex number division:

```python
c = 2 + 3j
d = 4 +6j

d/c
```

Which is the answer to $(4 + 6i)/(2 + 3i)$ after *rationalizing* the denominator.

Note: You cannot take integer division for Imaginary numbers

- Error received is **TypeError: can't take floor of complex number.**

Useful link for Division in Python:

- Webiste: geeksforgeeks

Power:

We can use '`**`' operator to denote power (raised to-).

For example: If we want, $c = a^b$

```
# Defining variables:
a = 7
b = 3

# Using the Power operator
c = a**b

#Result:
c
```

In case of complex numbers:

We expect $(2 + 3i)^2 = -5 + 12i$

```
(2+3j)**2
```

Hence verified.

Let us take another example:

```
(1+1j)**(1-1j)
```

Math module

import math

It contains various mathematical functions like trigonomical unctgions, ceiling, floor, etc.

To know about all the functions, we can visit the website to get the list of all the available functions. [Click here]

```
import math
print(math,ceil(5.7))
print(math,floor(5.7))
```

Control FLow

- Control Flow statements like loops and conditionals have blocks indicated by indentations. Any number of whitespaces is syntactically correct as long as it is consistent within a block. Typically `tab` is used for indentation which is equivalent to 4 spaces.

- Looping statements are used to repeat tasks. To iterate over a sequence of numbers, the built-in function `range()` can be used. It generates arithmetic progression.
- `while` loop can execute a set of statements as long as a condition is `True`
- `for` loop is used for iterating over a sequence (either list, tuple, dictionary, set or string)
- `break` is used to stop the loop even if the while condition is `True`
- `continue` is used to stop the current iteration, and continue to the next.

If statement

We can use if statement to run a code if a certain condition holds. If a expression evaluates to True, some statements are caried out. Otherwise, they aren't carried out.

- An `if` statement checks for the given condition
- Notations for logical conditons:
    - `a==b` - Equals
    - `a != b` - Not equal
    - `a < b` - Less than
    - `a <= b` - Less than or equal to
    - `a > b` - Greater than
    - `a >= b` - Greater than or equal to
- `elif` means else if (if the previous conditions were not true, then try this condition)
- `else` is for any other condition which doesn't belong to previous conditions.

```
if 10 > 5 :
  print("10 greater than 5")
print("program ended")
```

Exercise 7

If it's a hot day,

print: It is a hot day

Enjoy your day.

```
is_hot = True
if is_hot:
  print("It is a hot day")
print("Enjoy your day")
```

Else Statement

An else statement follows an if statement, a nd comtains code that is called when the if statement evaluates to False. As with is statements, the code inside the block shpould be indended.

```
number = 5
if number == 4:
  print("Number Equals 4")
else:
  print("Number does not equal 4")
```

Elif Statement

The elif statement (short form of else if) is a shortcut to use when chaining if and else statement.

A series of elif statements can have a final else block, called if none of if or elif expression is true.

In the below example, you define two variables room and area. You then construct if-elif-else and if-else conditions each for room and area, respectively.

In the first condition, you check if you are looking in the kitchen, elif you are looking in the bedroom, else you are looking around elsewhere. Depending on the value of the room variable, the satisfied condition is executed.

```
room = "bed"
if room == "kit":
  print("Looking around in the kitchen.")
elif room == "bed":
  print("looking around in the bedroom.")
else :
  print("looking around elsewhere.")
```

```
a = 4
if a<5:
    print(a,"is less than 5.")
elif a==5:
    print(a,"is equal to 5.")
else:
    print(a,"is greater than 5.")
```

```
b = float(input("Enter the number: ")) # takes the number from user and defines b as
if b < 10:
    print(b,"is less than 10.")
elif b==10:
    print(b,"is equal to 10.")
else:
    print(b,"is greater than 10.")
```

Exercise 8

Simar to previous exercise, instad to one boolean, use two boleans (is_hot and is_cold) and create a elif structure which will give messages for each case.

if it is a hot day

Print: It's a hot day. Drink plenty of water

If it is a cold day

Print: It's a cold day. Wear warm clothes.

End with "Enjoy your Day"

```
is_hot = False
is_cold = True
if is_hot:
  print("It's a hot day. Drink plenty of water")
elif is_cold:
  print("It's a cold day. Wear warm clothes")
else:
  print("It's a lovely day")
print("Enjoy your day")

```

Exercise 9

Price of a house is 1 million

If buyer has good credit,

> they need to put down 10%

Otherwise

> they need to put down 20%

Print the down payment

```
price = 1000000
good_credit = True
if good_credit:
  down_payment = price * 0.1
else:
  down_payment = price * 0.2
print (f"Down Payment: ${down_payment}")
```

Logical Operators

and, or

Example 1: If an applicant has high income AND good credit

Eligible for loan

Example 2: If an applicant has high income or good credit

Eligible for loan

```python
#Example 1
high_income = False
has_good_credit = True

if high_income and has_good_credit:
  print("Eligible for loan")
else:
  print("Not eligible for loan")

```

```python
#Example 2
high_income = False
has_good_credit = True

if high_income or has_good_credit:
  print("Eligible for loan")
else:
  print("Not eligible for loan")
```

Example 3: If an applicant has good credit AND doesn't have a criminal record:

Eligible for loan

```python
#Example 3
has_good_credit = False
no_criminal_record = True

if has_good_credit and no_criminal_record:
  print("Eligible for loan")
else:
  print("Not eligible for loan")
```

Exercise 10

If name is less then 3 characeters long

> name must be at atleast 3 characters

otherwise if it's more than 50 characters long

> name must be a maximum of 50 characters

otherwise

> name looks good!

```
name = input("Enter your name: ")
if len(name) < 3:
  print("name must be at atleast 3 characters")
elif len(name) > 50:
  print("name must be a maximum of 50 characters")
else:
  print("Name looks good!")
```

Exercise 11

Create Weight calculator for converting (kg to lbs) or(lbs to kg).

(Lbs = Weight * 0.45)

```
weight = int(input('Weight: '))
unit = input('(L)bs or (K)g: ')
if unit.upper == 'L':
  converted = weight  * 0.45
  print(f"You are {converted} kilos")
else:
  converted = weight / 0.45
  print(f"You are {converted} pounds")
```

for/range

With `for` loop we can iterate over a sequence or execute a set of statements, once for each item in list, tuple, set, etc.

```
for x in "Physics":
  print(x)
```

```
for x in range(10):
  print(x)
```

```
# range(inclusive, exclusive, step size)
# range(0,10,2) will give 0,2,4,6,8
for i in range(0,10,2):
    print(2**i,end='  ')
#This will print powers of 2 upto 2^8
```

```
# Breaking the loop
topics = ["Gravitational Dynamics","Wave mechanics","Quantum Mechanics","Statistical
for x in topics:
  print(x)
  if x == "Quantum Mechanics":
    break #This will stop the loop when above condition is reached
```

Exercise 12

Create a code to calculate Factorial(!) of a number using for loop.

```
f = 1
n = int(input("Enter the number: "))
for i in range(1,n+1): # index of range is n + 1
  f = f*i
print(n,"! = ",f)
```

While / Continue Loop

- `while` loop executes a set of statements as long as a condition is true.
- `continue` stops the current iteration and continues with the next.

```
i = 0
while i < 10:
  i = i + 1
  print(i)
# Try changing the order of i = i + 1 and print command and notice the difference
```

```
i = 0
while i < 10:
  print(i)
  if i == 5:
    break # This will exit the loop when i is 5
  i = i + 1
```

```
i = 0
while i < 10:
  i = i + 1
  if i == 5:
    continue # When i is 5 it will skip that iteration and continue
  print(i)
```

Exercise 13

Repeat exercise 12 using while loop.

```
f = 1
n = int(input("Enter a number: "))
counter = 1
while (counter <= n):
  f = f*counter
  counter = counter + 1
print(n,"! = ",f)
```

Double-click (or enter) to edit

Lists

- Lists is an ordered collection of elements enclosed within `[ ]`
- Lists are similar to tuples, the difference is that lists are $\mathrm{mutable}$, i.e., we can change data or reassign items.
- `L = [1,2,'a','b',True,3.145,False]`
- Accessing elements is same as tuples.

  `L[0]`, `L[3]`, `L[2,6]`, etc
- To change the element in list enter the element number in square bracket and reassign

  `L[0] = 100`
- To add element in list we use 'append',

  `L.append( )`
- To add list inside list,

  `L.append([5,6,7,8])`
- To remove element from list,

  `L.pop( )` - pops out the last element in list
  `L.pop(index)` - pops out the element at the said index

```
# Define a list using [ ] square brackets
L = [1,2,'a','b',True,3.145,1+2j]
```

```
# Access elements using their index
L[4]
```

```
# Changing element in list
L[0] = 100
print(L)
```

```
# Adding element in a list
L.append(24)
print(L)
```

```
# Adding a list to a list
L.append([5,6,7,8])
print(L)
```

```
# Removing elements
# Removing elements
L.pop() # Will remove the last element
print(L)
```

```
L.pop(3) # Will remove element at 3rd index i.e 'b'
print(L)
```