Our Frist Python Program

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

```
Hello World
```

```
print('Hello World')
```

```
Hello World
```

```
print('Inception BD')
```

```
Inception BD
```

```
print(Inception BD)
```

```
  File "/tmp/ipython-input-2043638736.py", line 1
    print(Inception BD)
          ^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
```

```
print(9)
```

```
9
```

```
print("9")
```

```
9
```

```
print(9.9)
```

```
9.9
```

```
print(True)
```

```
True
```

```
print(False)
```

```
False
```

```
print("Hello",1,2,3,4,5,6.6,True)
```

```
Hello 1 2 3 4 5 6.6 True
```

```
print("Hello",1,2,3,4,5,6.6,True, sep="-")
```

```
Hello-1-2-3-4-5-6.6-True
```

```
print("Hello",1,2,3,4,5,6.6,True, sep=",")
```

```
Hello,1,2,3,4,5,6.6,True
```

```python
print("Bappy", end="-")
print("Inception")
```

```
Bappy-Inception
```

```python
print("Bappy", end=" ")
print("Inception")
```

```
Bappy Inception
```

Data Type in Python & Comments

1. Integer
2. Float (decimal)
3. Boolean
4. String (text)
5. Complex
6. List
7. Tuple
8. Sets
9. Dictionary

```python
# This is a single line comment, this print function prints a sentence
print("My name is Bappy")
```

```
My name is Bappy
```

```python
"""
This is Multiline Comment
this print function prints a sentence
fhsdjhfksdhfkldsf
sdkhfkdjshflkdsf
sdlhfjdlkshfkdjs
"""

print("My name is Bappy")
```

```
My name is Bappy
```

```python
# Integer
```

```python
print(2)
```

```
2
```

```python
print(200000000000)
```

```
200000000000

type(200000000000)

int

# Float

print(3.3)

3.3

type(3.3)

float

# Bolean
# Python is case sensative language

print(False)

False

type(False)

bool

# String

print("Bappy Inception")

Bappy Inception

type("Bappy Inception")

str

# Complex

print(5 + 7j)

(5+7j)

type(5 + 7j)

complex

# List

print([1,2,3])

[1, 2, 3]

type([1,2,3])
```

```
list
```

```python
# Tuple
```

```python
print((1,2,3))
```

```
(1, 2, 3)
```

```python
type((1,2,3))
```

```
tuple
```

```python
# Sets
```

```python
print({1,2,3})
```

```
{1, 2, 3}
```

```python
type({1,2,3})
```

```
set
```

```python
# Dictionary
```

```python
print({"Name":"Bappy", "Age":25})
```

```
{'Name': 'Bappy', 'Age': 25}
```

```python
type({"Name":"Bappy", "Age":25})
```

```
dict
```

Variables, Keywords & Identifiers in Python

Variables

```python
a = 2  # int
print(a)
```

```
2
```

```python
b = 3.3 # Float
print(b)
```

```
3.3
```

```python
c = True
d = 'Bappy'
```

```
print(c)
print(d)
```

```
True Bappy
Bappy
```

```
type(c)
```

```
bool
```

```
name = "Alex"
print("Welcome", name)
```

```
Welcome Alex
```

```
a = 2
b = 5
```

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

```
7
```

```
# Dynamic Typing
a = 7
```

```
# Static Typing
int a = 7;
```

```
type(a)
```

```
int
```

```
a = 7
a = "bappy"
print(a)
```

```
bappy
```

```
# Dynamic Binding
a = 5
print(a)
a = "bappy"
print(a)

# Static Binding

int a = 7;

str a = "bappy";
```

```
5
bappy
```

```
a = 1
b = 2
c = 3

print(a)
print(b)
print(c)
```

```
1
2
3
```

```
a = 1
b = 2
c = 3

print(a, b, c)
```

```
1 2 3
```

```
a,b,c = 1,2,3
print(a, b, c)
```

```
1 2 3
```

```
a = 5
b = 5
c = 5

print(a, b, c)
```

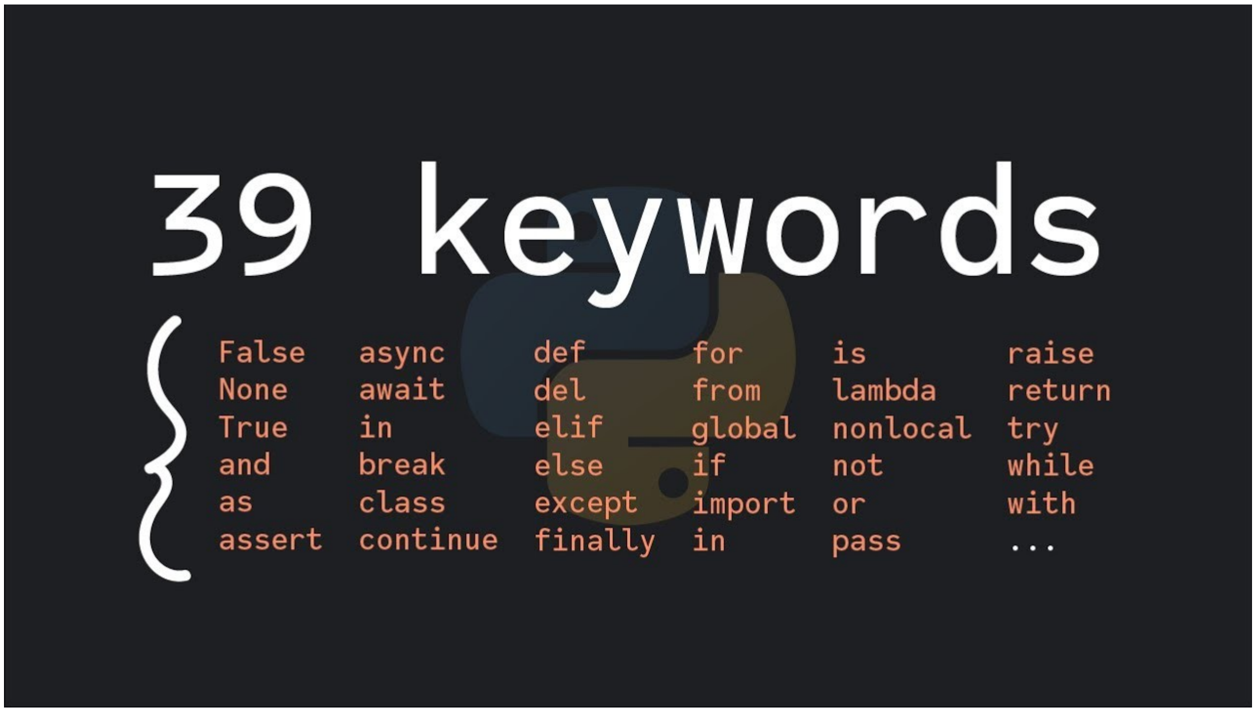
```
5 5 5
```

```
a = b = c = 5

print(a, b, c)
```

```
5 5 5
```

Keywords



39 keywords

False	async	def	for	is	raise
None	await	del	from	lambda	return
True	in	elif	global	nonlocal	try
and	break	else	if	not	while
as	class	except	import	or	with
assert	continue	finally	in	pass	...

Identifiers

```
name = "Bappy"
print(name)

Bappy

# You can't start with any digit

1name = 'bappy'

  File "/tmp/ipython-input-1109743195.py", line 3
    1name = 'bappy'
    ^
SyntaxError: invalid decimal literal


name1 = 'bappy'

# You can't use any special chars except _

*name = "Bappy"

  File "/tmp/ipython-input-1089038123.py", line 3
    *name = "Bappy"
    ^
```

```
SyntaxError: starred assignment target must be in a list or tuple
```

```
name$ = "Bappy"
```

```
  File "/tmp/ipython-input-3492505726.py", line 1
    name$ = "Bappy"
        ^
SyntaxError: invalid syntax
```

```
_name = "Bappy"
```

```
name_ = "bappy"
```

```
_ = "bappy"
```

Python Input

Static - Calender, Clock

Dynamic - Youtube, Facebook

```
input()
```

```
7
```

```
{"type":"string"}
```

```
var = input()
```

```
5.6
```

```
print(var)
```

```
5.6
```

```
type(var)
```

```
str
```

```
var = float(var)
type(var)
```

```
float
```

```
var = input("Enter Your Name: ")
print(var)
```

```
Enter Your Name: bappy
bappy
```

Simple Calculator using input

```
a = int(input("Enter the first number: "))
b = int(input("Enter the second number: "))


result = a - b
print(result)
```

```
Enter the first number: 5
Enter the second number: 4
1
```

Type Conversion in Python:

1. Implicit - Internally by Python
2. Explicit - By the Programmer

```
# Implicit

a = 5 + 5.5

print(a)
print(type(a))
```

```
10.5
<class 'float'>
```

```
# Explicit

b = 4 + int("4")
print(b)
print(type(b))
```

```
8
<class 'int'>
```

```
num = 34

print(float(num))
```

```
34.0
```

```
num = "34"

print(float(num))
```

```
34.0
```

```
num = 3.4

print(int(num))

3

num = 4 + 6j

type(num)

float(num)

----------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
/tmp/ipython-input-2729759340.py in <cell line: 0>()
      3 type(num)
      4
----> 5 float(num)

TypeError: float() argument must be a string or a real number, not
'complex'
```

Literals in Python

```
a = 3
print(type(a))

<class 'int'>

a = 0b1010 #Binary Literals
b = 100 #Decimal Literal
c = 0o310 #Octal Literal
d = 0x12c #Hexadecimal Literal

print(a)

10

print(b)

100

print(c)

200

print(d)

300
```

```
# Float Literals

float_1 = 10.5
float_2 = 1.5e2 # 1.5 * 10^2
float_3 = 1.5e-3 # 1.5 * 10^-3

print(float_1)
print(float_2)
print(float_3)
```

```
10.5
150.0
0.0015
```

```
#Complex Literal
x = 3.14j
```

```
info = """
My name is bappy.
I am teaching Python
I am a Data Scientist
shdflsdhlfsd
sdlkhflksdhnf
hnjklsdjfl
"""
```

```
info = "
My name is bappy.
I am teaching Python
I am a Data Scientist
"
```

```
  Cell In[15], line 1
    info = "
           ^
SyntaxError: unterminated string literal (detected at line 1)
```

```
string = 'This is Python'
strings = "This is Python"
char = "C"
multiline_str = """This is a multiline string with more than one line
code."""
unicode = u"\U0001f600\U0001F606\U0001F923"
raw_str = r"raw \n string"

print(string)
print(strings)
print(char)
print(multiline_str)
print(unicode)
print(raw_str)
```

```
This is Python
This is Python
C
This is a multiline string with more than one line code.
😀😆
raw \n string
```

```
# True = 1
# False = 0

a = True + 4
b = False + 10

print("a:", a)
print("b:", b)
```

```
a: 5
b: 10
```

```
# None

a = None

x = None
y = 2
z = 4

print(x,y,z)
```

```
None 2 4
```

Operators in Python

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators
- Membership Operators

Arithmetic Operators

```
print(4+3)
```

```
7
print(4-3)
1
print(4*3)
12
print(4/2)
2.0
print(4//2)
2
print(4%2)
0
print(5**2)
25
```

Relational Operators

```
print(4>5)
False
print(4<5)
True
print(4>=4)
True
print(4<=4)
True
print(4==4)
True
print(4!=4)
False
```

Logical Operators

```
print(0 and 1)
```

```
0
```

```
print(0 or 1)
```

```
1
```

```
print(not 1)
```

```
False
```

```
print(not 0)
```

```
True
```

Bitwise Operators

```
# bitwise and
print(2 & 3)

# bitwise or
print(2 | 3)

# bitwise xor
print(2 ^ 3)

#bitwise not
print(~3)

#right shift
print(4 >> 2)

#left shift
print(5 << 2)
```

```
2
3
1
-4
1
20
```

Assignment Operators

```
a = 2
```

```
a = 2

a += 1

print(a)

3
```

Membership Operators

```
# in/not in

print('B' in "Bangladesh")

True
```