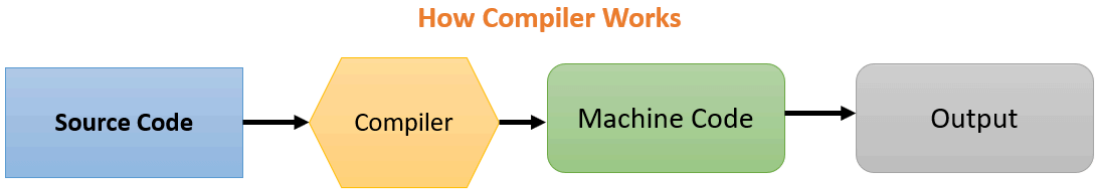
# python day -1

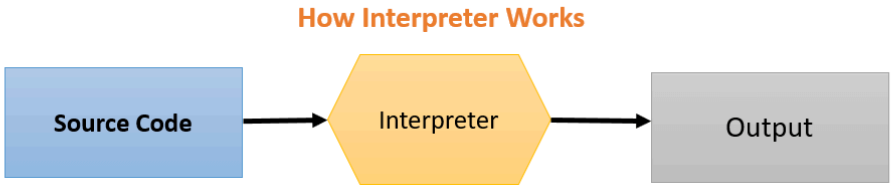
## python

### python errors:

```
1. unexacpted indent error                    - missing space like print or
for
2. sysntax error                              -
3. value error convert string into number     -
4. name error                                 - variable not defined
```

### what is python?




```
source code is executed directly by the interpreter, and the interpreter is
slow.
compilers are generally faster than interpreters because they translate the
entire source into machine code once,
where as interpreters translate the code line by line

- speed (compiler)
- debugging (interpreter)
- display all error once (compiler)
- less memory (interpreter)
- java convert bytecode (its support all operating system)
  JVM = development tool (java,javac,jar)
  JRE = runtime libraries, byte code
  JDK = java interpreter,JIT component,garbage collector

Python is a interpreter language. It was created by Guido van Rossum.
released in 1991.
```

```
- exit() - exit the python command line interface
- ctrl + z - stop python terminal

mutable   = something can be changed        => list, array
immutable = something cannot be changeable  => integer, tuple , str
```

**0. comments**

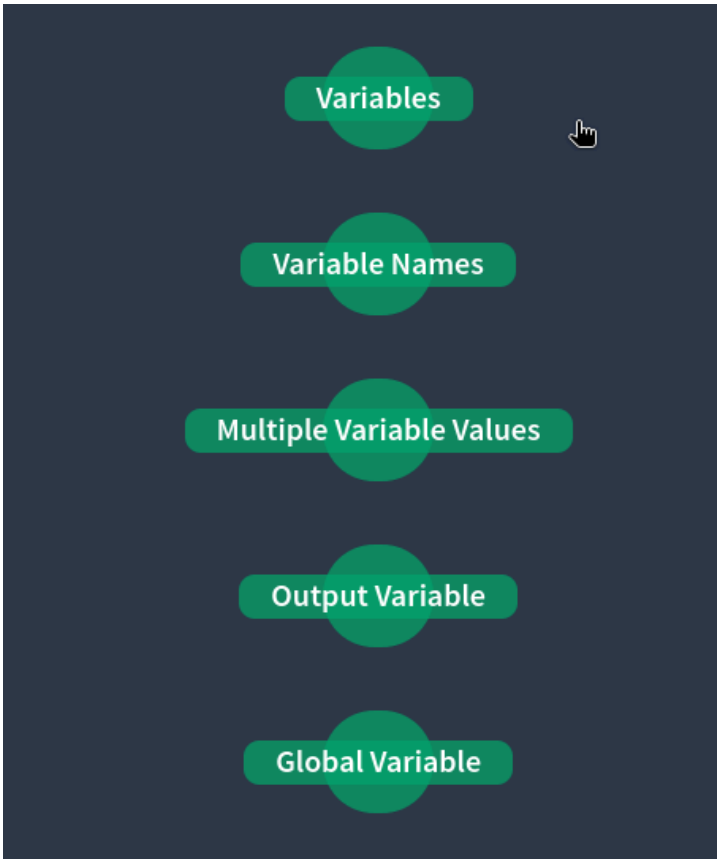
```
# This is a comment
'''
This is a multiline comment
'''
                                            """ + ''' = same meaning
"""
This is a multiline comment
"""

primitive operators:
____________________

+  - addition
-  - subtraction
*  - mulitiplication
/  - division float
// - division int
%  - reminder
** - exponend or power

comparision operator:
_____________________

== - equality
!= - inequality
>  - greater than
<  - less than
>= - greater than or equal to
<= - less than or equal to

boolean:
________
True  - 1
False - 0
```

```
logical operator:
_________________
And - two thinks correct
Or  - anyone is correct
Not - opposite think

Order of Execution:
____________________

B - bracket || function
O - order (power || root)
D - division
M - mulitplication
A - additon
S - subtraction
```

**1. find version**

```
import sys
print(sys.version)
```

**2. variables**




```
# variable cannot start with number
# variable cannot used python keywords
# variable is case-sensitive
```

```
# variable name must start with letter or underscore character

"""
camel case = myVariableName
pascal case= MyVariableName
snake case = my_vaibale_name
"""
x= 1
y= 2
print(x+y)
```

---

```
x = 5
y = "hello world"
print(a,b) # not work (x+y)
```

---

```
w, x, y, z = "king", "gifta", "jebaselvi", "jeyaraj"
print(x,y,z,w)
```

---

```
a = 'hello'
b = 'world'
print(a+b) # space not occur
print(a,b) # space occur
```

---

```
#inside & outside variable
x = 'kings'
def myfunc():
 x = 'kingslin'
 print('hello mr.' + x)  # its work inside function

myfunc()

print('hello mr.' + x) # its work outside function
```

---

```
#golbal keyword
x = 'kings'
def myfunc():
 global x
 x = 'kingslin'
```

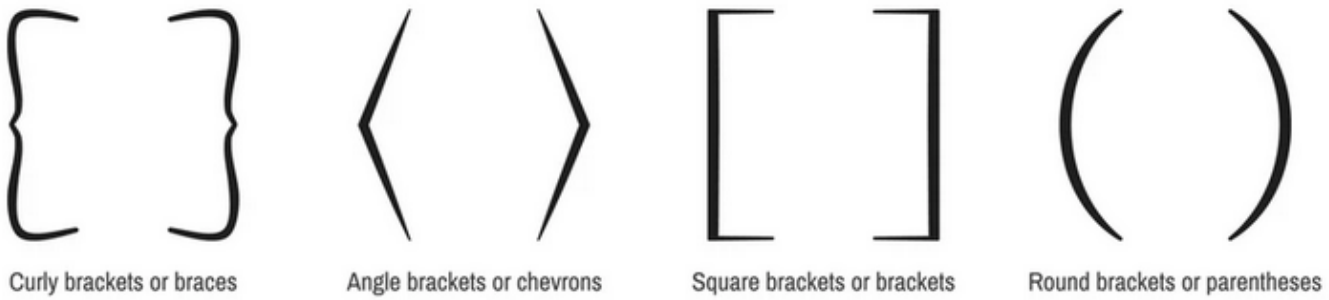
```
myfunc()

print("hello mr." + x)
```

---

**3. if condition**

```
if 20 > 10:
 print(" twenty is big number") # sometimes missing space, the syntax error occur
```

**4. datatype**



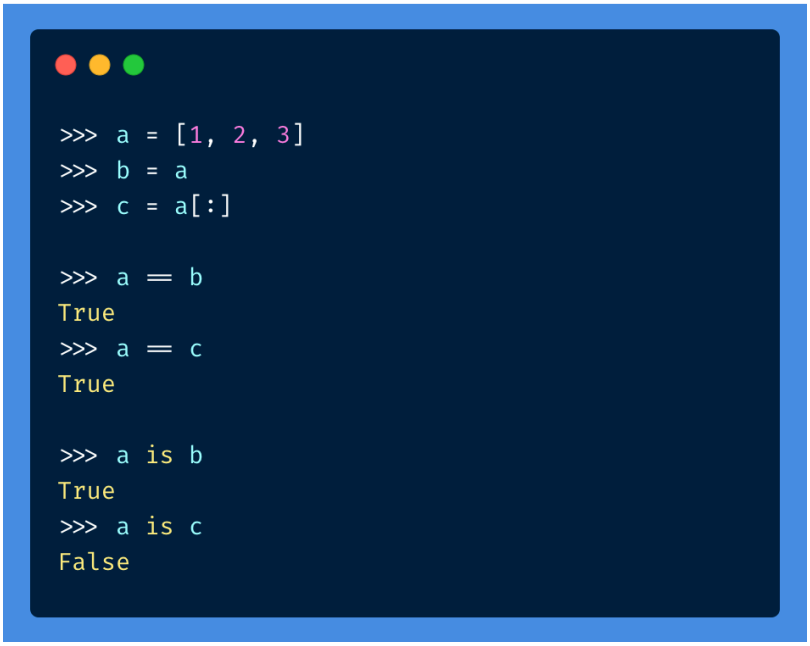

```
text type: str
nume type: int, float, complex
sequ type: list, tuple, range
mapp type: dict
set  type: set, frozenset
bool type: bool
bina type: bytes, bytearray, memoryview
none type: NoneType

x = str(4)      # '3'
y = int(4)      # 4
z = float(4)    # 4.0
print (type(x))
print (type(y))
print (type(z))

"hello world"   - str
44              - int
44.0            - float (Float can also be scientific numbers with an "e" to indicate the power of 10.)
1j              - complex
['app', 'soft'] - list
```

```
('app', 'soft') - tupe
range(6)        - range
{'app' : 'sof'} - dict
frozenset({'a', 'b'}) - frozenset
True & False    - bool
b'hello'        - bytes
bytearray(5)    - bytearray
memoryview(bytes(5))  - memoryview
None            - nonetype

#type conversion

x = 1
y = 'hello'
a = str(x)
b = int(y) # its not work because (string not convert numeric)

#random number
import random
print(random.randrange(1, 10))
```

**5. python casting**

```
# In python casting (or type conversion) is the process of converting one
data type into another.

#integer casting:
int("123") #123
int(3.14)  #3
int('hel') #value error

#float casting:
flaot(3) #3.0
float('e') or float(e)  #value error
x = 1e3 # 1000.0
```

**6. string manipulation:**

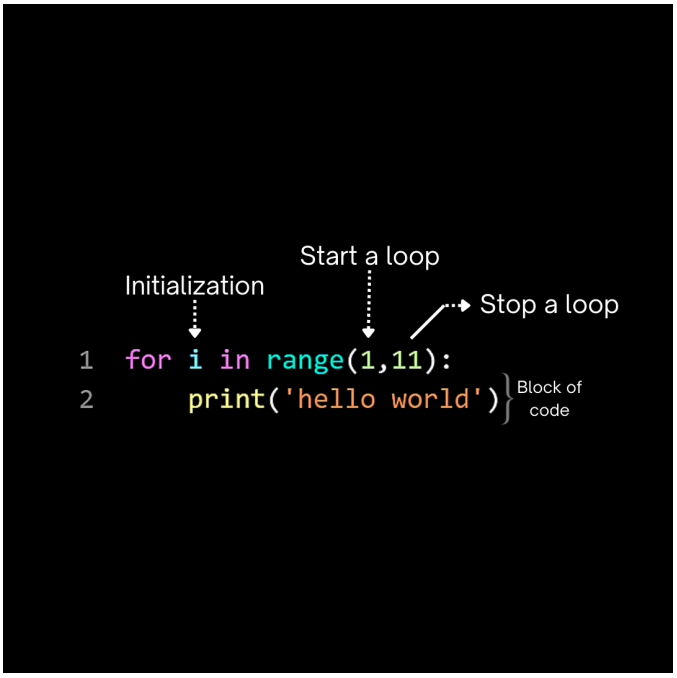


```
>>> a = [1, 2, 3]
>>> b = a
>>> c = a[:]

>>> a == b
True
>>> a == c
True

>>> a is b
True
>>> a is c
False
```

```
"Hello" == 'Hello'
a = "Hello"
print(a)

#python strings are Arrays of characters
#multiline string (its work on single & double quates)
a ='''hello everyone nice
      to meet you'''

#looping string using array format
for k in "banana":
 print(x)

#String length
a= "kingslin"
print(len(a)) # the output is 8

#checking String
txt = "kingslin jhon rajarikam"
 print("jhon" in txt)
if "king" not in txt:
 print("Not present")

#Slicing
a= "kingslin"
print(a[0:7])  print(a[:7])  print(a[0:])  print(a[-1])

#modify string(upper case)
a = "kingslin"
```

```
print(a.upper())    # OUTPUT IS KINGSLIN
print(a.lower())    # output is kingslin
a = a.replace("k", "g")
print(a.strip('n')) # its not work but it work remove white_space

#string concatenation (merging the two content)
a ="hello"
b ="world"
c = a + " " + b
print(c)

#string format
age = 20
details = f"my name is kingslin, Iam {age}"  # its support {10*100}
print(details)
```

**7. for loop:**




```
for k in range(1,100):  # for k in range(10*10):
 print("hello kingslin")
```

**8.while loop:**

```
i = 1
while i< 6:
 print(i)
 i+=2      #output is 1 3 5 incremented && condition executed by less than 6
```

**9.nested loop:**

```
why = ["small","big","tasty"]
fruits = ["apple","banana","cherry"]
 for x in why:
  for y in fruits:
   print(x+y)
```

**10. address or value:**

```
a = 10
b = 10
id(a)
id(b)     # address is same so followed 256
a = 1000
b = 1000
id(a)
id(b)     # address is not same so not followed 256
```

**11. function:**

```
# lambda function is anonymous function
x = lambda a : a+10
print(x(5))         # output is 15

x = lambda a,b : a+b
print(x(10,10))     # output is 20

def myfunc():
 print("hello everyone")

myfunc()
```

**12. IF condition:**

```
a = 10
b = 44
if b> a:
 print("b is big")
else
 print("a is big")
```

**13. file handling:**

```
f = open("filename.txt")
print(f.read())
f.close()
```

```
f=open("filename.txt", "w")
f.write("now kinglslin add something")
f.close()

import os
if os.path.exists("gui.py")
 os.remove("gui.py")              #rmdir used to remove folder
else
 print("the file does not exist")
```