In [2]:
```
import sys
import keyword
import operator
from datetime import datetime
import os
```

Keywords

In [3]:
```
print(keyword.kwlist)
```

```
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'clas
s', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from',
'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass',
'raise', 'return', 'try', 'while', 'with', 'yield']
```

In [4]:
```
len(keyword.kwlist)
```

Out[4]: 35

Identifiers

In [12]:
```
1var=10 # Identifier can't start with a digit
```

```
  Cell In[12], line 1
    1var=10 # Identifier can't start with a digit
    ^
SyntaxError: invalid decimal literal
```

In [11]:
```
val2@=35 # Identifier can't use special symbols
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[11], line 1
----> 1 val2@=35

TypeError: unsupported operand type(s) for @=: 'int' and 'int'
```

In [10]:
```
import=125 # Keywords can't be used as identifiers
```

```
  Cell In[10], line 1
    import=125 # Keywords can't be used as identifiers
           ^
SyntaxError: invalid syntax
```

In [8]:
```
val2=10
```

In [9]:
```
val_=99
```

Comments in Python

In [14]:
```
# Single line comment
```

```
val1 = 10
```

In [15]:
```
# Multiple # line
# comment
val1 = 10
```

In [16]:
```
'''
Multiple line
comment '''
val1 = 10
```

In [17]:
```
"""
Multiple line
comment """
val1 = 10
```

Statements

In [20]:
```
p=20
q=20
r=q
p, type(p), hex(id(p))
```

Out[20]: (20, int, '0x7ffc38002c18')

In [21]:
```
q , type(q), hex(id(q))
```

Out[21]: (20, int, '0x7ffc38002c18')

In [22]:
```
r , type(r), hex(id(r))
```

Out[22]: (20, int, '0x7ffc38002c18')

In [24]:
```
p = 20
p = p + 10 # Variable Overwriting
p
```

Out[24]: 30

In [25]:
```
p, type(p), hex(id(p))
```

Out[25]: (30, int, '0x7ffc38002d58')

In [28]:
```
q, hex(id(q))
```

Out[28]: (20, '0x7ffc38002c18')

In [29]:
```
r, hex(id(r))
```

Out[29]: (20, '0x7ffc38002c18')

Variable Assigment

In [40]:
```
intvar = 10 # Integer variable
floatvar = 2.578960767067056 # Float Variable
strvar = "Python Language" # String variable
print(intvar)
print(floatvar)
print(strvar)
```

```
10
2.578960767067056
Python Language
```

In [33]:
```
p1 = p2 = p3 = p4 = 44 # All variables pointing to same value
print(p1,p2,p3,p4)
```

```
44 44 44 44
```

In [34]:
```
hex(id(p1)), hex(id(p2)), hex(id(p3)), hex(id(p4))
```

Out[34]:
```
('0x7ffc38002f18', '0x7ffc38002f18', '0x7ffc38002f18', '0x7ffc38002f18')
```

Data Types

Numeric

In [46]:
```
val1 = 1000000000000000000000 # Integer data type
print(val1)
print(type(val1)) # type of object
print(sys.getsizeof(val1)) # size of integer object in bytes
print(val1, " is Integer?", isinstance(val1, int)) # val1 is an instance of int
```

```
1000000000000000000000
<class 'int'>
36
1000000000000000000000  is Integer? True
```

In [43]:
```
print(sys.getsizeof(strvar))
```

```
56
```

In [44]:
```
print(sys.getsizeof(floatvar))
```

```
24
```

In [51]:
```
val2=9278999979760740940965409650964
print(val2)
print(type(val2))
print(sys.getsizeof(val2))
print(val2,"is float?",isinstance(val2,float))
```

```
9.278999979760741e+26
<class 'float'>
24
9.278999979760741e+26 is float? True
```

In [54]:
```
val3 = 2500000000000000000.6767 + 10.22342425j # Complex data type
print(val3)
print(type(val3)) # type of object
```

```
print(sys.getsizeof(val3)) # size of complex object in bytes
print(val3, " is complex?", isinstance(val3, complex)) # val3 is an instance of
```

```
(2.5e+18+10.22342425j)
<class 'complex'>
32
(2.5e+18+10.22342425j)  is complex? True
```

In [55]:
```
sys.getsizeof(int()) # size of integer object in bytes
```

Out[55]: 28

In [56]:
```
sys.getsizeof(float()) # size of float object in bytes
```

Out[56]: 24

In [57]:
```
sys.getsizeof(complex()) # size of complex object in bytes
```

Out[57]: 32

Boolean

In [58]:
```
bool1 = True
```

In [59]:
```
bool2 = False
```

In [60]:
```
print(type(bool1))
```

```
<class 'bool'>
```

In [61]:
```
print(type(bool2))
```

```
<class 'bool'>
```

In [62]:
```
isinstance(bool1, bool)
```

Out[62]: True

In [63]:
```
bool(0)
```

Out[63]: False

In [64]:
```
bool(1)
```

Out[64]: True

In [66]:
```
bool(201)
```

Out[66]: True

In [67]:
```
bool(None)
```

Out[67]: False

In [68]:
```
bool()
```

Out[68]: False

In [69]:
```
bool (False)
```

Out[69]: False

Strings

String Creation

In [70]:
```
str1 = "HELLO PYTHON"
print(str1)
```

```
HELLO PYTHON
```

In [71]:
```
mystr = 'Hello World' # Define string using single quotes
print(mystr)
```

```
Hello World
```

In [72]:
```
mystr = "Hello World" # Define string using double quotes
print(mystr)
```

```
Hello World
```

In [76]:
```
mystr = '''Hello

World ''' # Define string using triple quotes
print(mystr)
```

```
Hello

World
```

In [77]:
```
mystr = ('Happy '

'Monday '
         'Everyone')
print(mystr)
```

```
Happy Monday Everyone
```

In [90]:
```
mystr2 = 'Woohoo '
mystr2 = mystr2*5
mystr2
```

Out[90]: 'Woohoo Woohoo Woohoo Woohoo Woohoo '

In [91]:
```
len(mystr2)
```

Out[91]: 35

String Indexing

In [92]:
```
str1
```

Out[92]: 'HELLO PYTHON'

In [94]:
```
str1[0] # First character in string "str1"
```

Out[94]: 'H'

In [96]:
```
str1[len(str1)-1] # Last character in string using len function
```

Out[96]: 'N'

In [97]:
```
str1[-1] # Last character in string
```

Out[97]: 'N'

In [99]:
```
str1[6] #Fetch 7th element of the string
```

Out[99]: 'P'

In [100…
```
str1[5]
```

Out[100… ' '

String Slicing

In [103…
```
str1[0:5] # String slicing - Fetch all characters from 0 to 5 index
```

Out[103… 'HELLO'

In [108…
```
str1[6:12] # String slicing - Retreive all characters between 6 - 12 index loc
```

Out[108… 'PYTHON'

In [112…
```
str1[-4:] # Retreive last four characters of the string
```

Out[112… 'THON'

In [113…
```
str1[-6:] # Retreive last six characters of the string
```

Out[113… 'PYTHON'

In [121…
```
str1[:4] # Retreive first four characters of the string
```

Out[121… 'HELL'

In [120…
```
str1[:6] # Retreive first six characters of the string
```

Out[120… 'HELLO '

Update & Delete String

In [122…
```
str1
```

Out[122… 'HELLO PYTHON'

In [126…
```
str1[0:5] = 'HOLAA' # Strings are immutable
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[126], line 1
----> 1 str1[0:5] = 'HOLAA'

TypeError: 'str' object does not support item assignment
```

In [127…
```
del str1 # Delete a string
print(str1)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[127], line 2
      1 del str1 # Delete a string
----> 2 print(str1)

NameError: name 'str1' is not defined
```

String concatenation

In [129…
```
# String concatenation
s1 = "Hello"
s2 = "Asif"
s3 = s1 + s2
print(s3)
```

```
HelloAsif
```