In [1]:
```
import sys
import keyword
import operator
from datetime import datetime
import os
```

In [4]:
```
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 [6]:
```
len(keyword.kwlist)
```

Out[6]:
```
35
```

Identifiers

In [9]:
```
1var = 10
```

```
  Cell In[9], line 1
    1var = 10
    ^
SyntaxError: invalid decimal literal
```

In [11]:
```
val2@ = 35
```

```
  Cell In[11], line 1
    val2@ = 35
          ^
SyntaxError: invalid syntax
```

In [13]:
```
import = 125
```

```
  Cell In[13], line 1
    import = 125
           ^
SyntaxError: invalid syntax
```

In [15]:
```
val2 = 10
```

In [17]:
```
val_ = 9
```

Comments in Phython

In [20]:
```
# Single line comment
val1 = 10
```

In [22]:
```
# Multiple
# line
# comment
val1 = 10
```

```
In [26]: '''
         Multiple
         line
         comment
         '''
         val1 = 10
```

```
In [28]: """
         Multiple
         line
         comment
         """
         val1 = 10
```

Statements

```
In [31]: p = 20 #Creates an integer object with value 20 and assigns the variable p to p
         q = 20 # Create new reference q which will point to value 20. p & q will be poi
         r = q # variable r will also point to the same location where p & q are pointin
         p , type(p), hex(id(p)) # Variable P is pointing to memory location '0x7fff6d71a

Out[31]: (20, int, '0x7ffb46852c18')
```

q , type(q), hex(id(q))

```
In [43]: r , type(r), hex(id(r))

Out[43]: (20, int, '0x7ffb46852c18')
```

```
In [37]: p = 20
         p = p + 10
         p

Out[37]: 30
```

Variable Assigment

```
In [40]: intvar = 10 # Integer variable
         floatvar = 2.57 # Float Variable
         strvar = "Python Language" # String variable
         print(intvar)
         print(floatvar)
         print(strvar)

        10
        2.57
        Python Language
```

Multiple Assignment

```
In [48]: intvar , floatvar , strvar = 10,2.57,"Python Language" # Using commas to separat
         print(intvar)
print(floatvar)
print(strvar)
```

```
10
2.57
Python Language
```

In [50]:

```
p1 = p2 = p3 = p4 = 44 # All variables pointing to same value
print(p1,p2,p3,p4)
```

```
44 44 44 44
```

Data Types

Numeric

In [54]:

```
val1 = 10
print(val1)
print(type(val1))
print(sys.getsizeof(val1))
print(val1, " is Integer?", isinstance(val1, int))
```

```
10
<class 'int'>
28
10  is Integer? True
```

In [56]:

```
val2 = 92.78
print(val2)
print(type(val2))
print(sys.getsizeof(val2))
print(val2, " is float?", isinstance(val2, float))
```

```
92.78
<class 'float'>
24
92.78  is float? True
```

In [58]:

```
val3 = 25 + 10j
print(val3)
print(type(val3))
print(sys.getsizeof(val3))
print(val3, " is complex?", isinstance(val3, complex))
```

```
(25+10j)
<class 'complex'>
32
(25+10j)  is complex? True
```

In [60]:

```
sys.getsizeof(int())
```

Out[60]:

```
28
```

In [62]:

```
sys.getsizeof(float())
```

Out[62]:

```
24
```

In [64]:
```
sys.getsizeof(complex())
```

Out[64]:
```
32
```

Booleam

In [67]:
```
bool1 = True
```

In [79]:
```
bool2 = False
```

In [83]:
```
print(type(bool1))
```

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

In [85]:
```
print(type(bool2))
```

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

In [87]:
```
isinstance(bool1, bool)
```

Out[87]:
```
True
```

In [89]:
```
bool(0)
```

Out[89]:
```
False
```

In [91]:
```
bool(1)
```

Out[91]:
```
True
```

In [93]:
```
bool(None)
```

Out[93]:
```
False
```

In [95]:
```
bool (False)
```

Out[95]:
```
False
```

String

String Creation

In [99]:
```
str1 = "HELLO PYTHON"
print(str1)
```

```
HELLO PYTHON
```

In [101…
```
mystr = 'Hello World'
print(mystr)
```

```
Hello World
```

In [103...

```
mystr = "Hello World"
print(mystr)
```

```
Hello World
```

In [105...

```
mystr = '''Hello
World '''
print(mystr)
```

```
Hello
World
```

In [107...

```
mystr = """Hello
World"""
print(mystr)
```

```
Hello
World
```

In [109...

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

```
Happy Monday Everyone
```

In [111...

```
mystr2 = 'Woohoo '
mystr2 = mystr2*5
mystr2
```

Out[111...

```
'Woohoo Woohoo Woohoo Woohoo Woohoo '
```

In [113...

```
len(mystr2)
```

Out[113...

```
35
```

String Indexing

In [116...

```
str1
```

Out[116...

```
'HELLO PYTHON'
```

In [118...

```
str1[0]
```

Out[118...

```
'H'
```

In [120...

```
str1[len(str1)-1]
```

Out[120...

```
'N'
```

In [122...

```
str1[-1]
```

Out[122...

```
'N'
```

In [124...

```
str1[6]
```

```
Out[124…  'P'
```

```
In [126…  str1[5]
```

```
Out[126…  ' '
```

String Slicing

```
In [129…  str1[0:5]
```

```
Out[129…  'HELLO'
```

```
In [131…  str1[6:12]
```

```
Out[131…  'PYTHON'
```

```
In [133…  str1[-4:]
```

```
Out[133…  'THON'
```

```
In [135…  str1[-6:]
```

```
Out[135…  'PYTHON'
```

```
In [137…  str1[:4]
```

```
Out[137…  'HELL'
```

```
In [139…  str1[:6]
```

```
Out[139…  'HELLO '
```

Update and Delete String

```
In [142…  str1
```

```
Out[142…  'HELLO PYTHON'
```

```
In [146…  str1[0:5] = 'HOLAA'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[146], line 1
----> 1 str1[0:5] = 'HOLAA'

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

```
In [148…  del str1 # Delete a string
          print(srt1)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[148], line 2
      1 del str1 # Delete a string
----> 2 print(srt1)

NameError: name 'srt1' is not defined
```

String concatenation

In [151…

```
s1 = "Hello"
s2 = "Asif"
s3 = s1 + s2
print(s3)
```

```
HelloAsif
```

In [ ]: