In [1]:

```
import sys
import keyword
import operator
from datetime import datetime
import os
```

In [ ]:

```
```

Keywords

In [2]:

```
print(keyword.kwlist)
```

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

In [3]:

```
len(keyword.kwlist)
```

Out[3]:

```
35
```

In [ ]:

```
```

Identifiers

1var = 10 # variables cant start with a digit

In [5]:

```
val2@ =35   # variables cant use special symbols except _
```

```
  Cell In[5], line 1
    val2@ =35
          ^
SyntaxError: invalid syntax
```

In [6]:

```
import = 125     # keyword cant be used as identifiers
```

```
  Cell In[6], line 1
    import = 125
           ^
SyntaxError: invalid syntax
```

In [7]:

```
val2=10
val2
```

Out[7]:

```
10
```

In [8]:

```
val_=99
val_
```

Out[8]:
```
99
```

In [ ]:
```

```

Comments in python

In [9]:
```
val1=10

# single line comment
```

In [10]:
```
# Multi
# line
# comment

val1 = 15
```

In [11]:
```
'''
multi
line
comment
'''

val2 =20
```

In [12]:
```
"""
Multi
line
comment
"""
val3=52
```

In [ ]:
```

```

Statements

In [15]:
```
p=20
q=20
r=q

p,type(p),id(p),hex(id(p))
```

Out[15]:
```
(20, int, 140703530495000, '0x7ff817f42c18')
```

In [16]:
```
q,type(q),id(q),hex(id(q))
```

Out[16]:
```
(20, int, 140703530495000, '0x7ff817f42c18')
```

In [17]:
```
r,type(r), id(r), hex(id(r))
```

Out[17]:
```
(20, int, 140703530495000, '0x7ff817f42c18')
```

In [ ]:

In [18]:
```
p=20
p=p+15      #variable overwriting
p
```

Out[18]:
```
35
```

In [ ]:

Variable assignment

In [19]:
```
intvar=10
floatvar=2.57
strvar="python language"

print(intvar)
print(floatvar)
print(strvar)
```

```
10
2.57
python language
```

In [ ]:

Multiple assignments

In [20]:
```
intvar,floatvar,strvar=15,5.89,'nit ameerpet'
print(intvar)
print(floatvar)
print(strvar)
```

```
15
5.89
nit ameerpet
```

In [ ]:

Datatypes

In [22]:
```
val1 = 10

print(val1)
print(type(val1))
print(sys.getsizeof(val1))    #size of integer object in bytes
print(val1,"is integer?", isinstance(val1,int))   #val1 is an instance of integer
```

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

In [25]:

```
val2 = 76.84

print(val2)
print(type(val2))
print(sys.getsizeof(val2))
print(val1,"is float?", isinstance(val2,float))
```

```
76.84
<class 'float'>
24
10 is float? True
```

In [26]:

```
val3 = 7+9j

print(val3)
print(type(val3))
print(sys.getsizeof(val3))
print(val1,"is complex?", isinstance(val3,complex))
```

```
(7+9j)
<class 'complex'>
32
10 is complex? True
```

In [ ]:

In [27]:

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

Out[27]:

```
28
```

In [28]:

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

Out[28]:

```
24
```

In [30]:

```
sys.getsizeof(complex())
```

Out[30]:

```
32
```

In [31]:

```
sys.getsizeof(bool())
```

Out[31]:

```
28
```

In [ ]:

Boolean

```
In [32]: bool1=True
         bool2=False
```

```
In [33]: print(type(bool1))
         print(type(bool2))
```

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

```
In [35]: isinstance(bool1,bool)
```

```
Out[35]: True
```

```
In [36]: isinstance(bool2,bool)
```

```
Out[36]: True
```

```
In [37]: bool(0)
```

```
Out[37]: False
```

```
In [38]: bool(1)
```

```
Out[38]: True
```

```
In [39]: bool(None)
```

```
Out[39]: False
```

```
In [40]: bool(False)
```

```
Out[40]: False
```

```
In [41]: bool(True)
```

```
Out[41]: True
```

```
In [ ]:
```

Strings

```
In [42]: str1="Hello Python"
         print(str1)
```

```
Hello Python
```

```
In [43]: mystr ='Hello World'    # string in single quotes
         print(mystr)
```

```
Hello World
```

```
In [45]: mystr="Hello World"    # string in double quotes
         print(mystr)

         Hello World
```

```
In [47]: mystr = """Hello
         world"""                # string in triple quotes
         print(mystr)

         Hello
         world
```

```
In [53]: mystr=('Happy '
                'monday '
                'everyone ')
         print(mystr)

         Happy monday everyone
```

```
In [57]: mystr2='Woohoo '
         mystr2= mystr2*5
         print(mystr2)

         Woohoo Woohoo Woohoo Woohoo Woohoo
```

```
In [58]: len(mystr2)

Out[58]: 35
```

```
In [ ]:
```

String indexing

```
In [59]: str1

Out[59]: 'Hello Python'
```

```
In [60]: str1[0]

Out[60]: 'H'
```

```
In [65]: str1[len(str1)-1]

Out[65]: 'n'
```

```
In [66]: str1[-1]

Out[66]: 'n'
```

```
In [67]: str1[6]

Out[67]: 'P'
```

```
In [68]: str1[10]

Out[68]: 'o'

In [69]: str1[5]

Out[69]: ' '

In [ ]:
```

String Slicing

```
In [70]: str1[0:5]

Out[70]: 'Hello'

In [71]: str1[6:12]

Out[71]: 'Python'

In [73]: str1[-4:]

Out[73]: 'thon'

In [75]: str1[:6]

Out[75]: 'Hello '

In [ ]:
```

Update & Delete string

```
In [78]: str1

Out[78]: 'Hello Python'

In [80]: str1[0:5] = 'Holaa'    #strings are immutable
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[80], line 1
----> 1 str1[0:5] = 'Holaa'

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

```
In [81]: del str1
         print(str1)    #Delete a string
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[81], line 2
      1 del str1
----> 2 print(str1)

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

In [ ]:

String concantation

In [82]:

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

```
HelloAsif
```

In [ ]:

In [ ]:

In [ ]: