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

In [3]:
```
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 [4]:
```
len(keyword.kwlist)
```

Out[4]:
```
35
```

Identifiers

In [7]:
```
1var = 10 # Identifier can't start with a digit
```

```
 Cell In[7], line 1
   1var = 10 # Identifier can't start with a digit
   ^
SyntaxError: invalid decimal literal
```

In [8]:
```
v@= 35 # Identifier can't use special symbols
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[8], line 1
----> 1 v@= 35

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

In [9]:
```
import = 125 # Keywords can't be used as identifiers
```

```
 Cell In[9], line 1
   import = 125 # Keywords can't be used as identifiers
          ^
SyntaxError: invalid syntax
```

Correct way to identify a identifiyer

In [12]:
```
val2 = 10
```

In [13]:
```
val_ = 99
```

Statements

In [15]:
```
a= 12
b=21
c=b
```

```
a,type(a),hex(id(a))
```

Out[15]:

```
(12, int, '0x7ffdaf102b18')
```

In [16]:

```
b,type(b),hex(id(b))
```

Out[16]:

```
(21, int, '0x7ffdaf102c38')
```

In [17]:

```
c,type(c),hex(id(c))
```

Out[17]:

```
(21, int, '0x7ffdaf102c38')
```

In [18]:

```
a=20
v=a+10
v
```

Out[18]:

```
30
```

Variable Assigment

In [19]:

```
intvar = 12 # Integer variable
floatvar = 5.57 # Float Variable
strvar = "Python Language" # String variable
print(intvar)
print(floatvar)
print(strvar)
```

```
12
5.57
Python Language
```

Multiple Assignments

In [20]:

```
intvar , floatvar , strvar = 10,2.57,"Python Language" # Using commas to separat
print(intvar)
print(floatvar)
print(strvar)
```

```
10
2.57
Python Language
```

In [21]:

```
a=s=d=f=g=33
print(a,s,d,f,g)
```

```
33 33 33 33 33
```

In [22]:

```
val1 = 10 # 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
```

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

In [23]:

```python
val2 = 92.78 # Float data type
print(val2)
print(type(val2)) # type of object
print(sys.getsizeof(val2)) # size of float object in bytes
print(val2, " is float?", isinstance(val2, float)) # Val2 is an instance of floa
```

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

In [24]:

```python
val3 = 25 + 10j # Complex data type
print(val3)
print(type(val3)) # type of object
print(sys.getsizeof(val3)) # size of float object in bytes
print(val3, " is complex?", isinstance(val3, complex)) # val3 is an instance of
```

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

In [25]:

```python
sys.getsizeof(int()) # size of integer object in bytes
```

Out[25]:

```
28
```

In [26]:

```python
sys.getsizeof(float()) # size of float object in bytes
```

Out[26]:

```
24
```

In [27]:

```python
sys.getsizeof(complex()) # size of complex object in bytes
```

Out[27]:

```
32
```

In [ ]: