Assignment 3 - Python

In [2]:
```
# Importing necessary fields
import sys
import keyword
import operator
import os
from datetime import datetime
```

Keywords

In [17]:
```
# Keywords are the Reserved words in Python and can't be used as an Identifier
```

In [5]:
```
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']
35
```

In [6]:
```
# Find Length of Keywords
print(len(keyword.kwlist))           # Total 35 Keywords
```

```
35
```

Identifiers

In [8]:
```
# An Identifier is a name given to entities like class, functions, variables etc.,
# Rules: can't start with a Digit, can't start with Special Numbers, can't start wi

1var = 10     # can't start with digits
print(1var)
```

```
  Cell In[8], line 4
    1var = 10     # can't start with digits
    ^
SyntaxError: invalid decimal literal
```

In [9]:
```
val2@ = 15   # can't start with special characters
print(var2@)
```

```
  Cell In[9], line 1
    val2@ = 15   # can't start with special characters
         ^
SyntaxError: invalid syntax
```

In [10]:
```
import = 25    # can't start with keywords
```

```
print(import)
```

```
  Cell In[10], line 1
    import = 25    # can't start with keywords
           ^
SyntaxError: invalid syntax
```

In [11]:
```
val2 = 10         # Correct Identifier
print(val2)
```

```
10
```

In [12]:
```
val_ = 30      # Correct Identifier
print(val2)
```

```
10
```

Comments in Python

In [13]:
```
# Comments can be used to explain the code for more readabilty
# Single Line Comment
val1 = 10
```

In [14]:
```
# Multiple
# Line
# Comment
val2 = 15
```

In [15]:
```
'''Multiple
Line
Comment
'''
val3 = 20
```

In [16]:
```
"""
Multiple
Line
Comment
"""
val4 = 25
```

Statements

In [21]:
```
# Instructions that a Python Interpreter can execute

p = 20 # p is the variable & storing int value is 20
q = 20.50 # q is the variable & storing float value is 20
r = q
print(p)
print(type(p))
print(id(p))
print("----------------")
```

```
print(q)
print(type(q))
print(id(q))
print("----------------")
print(r)
print(type(r))
print(int(r))
print(id(r))
```

```
20
<class 'int'>
140708779211800
----------------
20.5
<class 'float'>
2910431784976
----------------
20.5
<class 'float'>
20
2910431784976
```

In [22]:
```
p = 20
p = p + 10   # variable overwriting
print(p)
```

```
30
```

Variable Assignments

In [23]:
```
intvar = 10     # Integer value
floatvar = 2.57  # Float value
strvar = 'Vihari Nandan'  # String value
print(intvar)
print(floatvar)
print(strvar)
```

```
10
2.57
Vihari Nandan
```

Multiple Assignments

In [24]:
```
intvar, floatvar, strvar = 10,2.57, 'Vihari Nandan'
print(intvar)
print(floatvar)
print(strvar)
```

```
10
2.57
Vihari Nandan
```

In [25]:
```
p1 = p2 = p3 = p4 = 33    # Multiple variables are assigned same value
```

```
print(p1,p2,p3,p4)
```

```
33 33 33 33
```

Data Types

Numeric Data Type

In [26]:
```
val1 = 10   # Integer data type
print(val1)    # value of 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 [27]:
```
val2 = 92.78   # float data type
print(val2)     # value of 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 float
```

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

In [28]:
```
val3 = 25+10j # Complex data type
print(val3)   # complex data type
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 com
```

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

In [31]:
```
print(sys.getsizeof(int()))    # size of integer object in bytes
```

```
28
```

In [33]:
```
print(sys.getsizeof(float()))      # size of float object in bytes
```

```
24
```

In [34]:
```
print(sys.getsizeof(complex()))     # size of complex object in bytes
```

```
32
```

Boolean

In [35]:
```
# Boolean datatype can have only two possible values true or false

bool1 = True
```

In [36]:
```
bool2 = False
```

In [38]:
```
print(type(bool1))
print(type(bool2))
```

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

In [40]:
```
isinstance(bool1,bool)
```

Out[40]: True

In [42]:
```
print(bool(0))
print(bool(1))
print(bool(None))
print(bool(False))
```

```
False
True
False
False
```

Strings

In [44]:
```
# String Creation

str1 = 'VIHARI NANDAN'   # with Single quotes
print(str1)
```

```
VIHARI NANDAN
```

In [45]:
```
str2 = "Vihari Nandan"    # with Double quotes
print(str2)
```

```
Vihari Nandan
```

In [46]:
```
str3 = '''Vihari
           Nandan'''   # with Multiple lines triple quotes
print(str3)
```

```
Vihari
           Nandan
```

In [48]:
```
str4 = """Vihari
                Nandan"""           # with Multiple lines triple quotes
print(str4)
```

```
Vihari
                Nandan
```

In [49]:
```
str5 = ('Happy '
        'Monday '
        'Everyday ')
print(str5)
```

```
Happy Monday Everyday
```

In [55]:
```
str6 = 'Om '
str6 = str6*5
print(str6)
```

```
Om Om Om Om Om
```

In [52]:
```
str7 = "Om \n"
print(str7*5)
```

```
Om
Om
Om
Om
Om
```

In [56]:
```
print(len(str6))
```

```
15
```

String Indexing

In [58]:
```
str1 = 'VIHARI NANDAN'

print(str1[0])       # first character in string str1
```

```
V
```

In [60]:
```
print(str1[len(str1)-1])            # last character in string using len function
```

```
N
```

In [61]:
```
print(str1[-1])  # Last character in string
```

```
N
```

In [62]:
```
print(str1[7])     # fetch 7th element of the string
```

```
N
```

String Slicing

In [64]:
```
print(str1[0:5])       # String slicing - Fetch all characters from 0 to 5 index lo
```

```
VIHAR
```

In [65]:
```
print(str1[6:12])      # String slicing - Retreive all characters between 6 - 12 in
```

```
NANDA
```

In [66]:
```
print(str1[-4])       # Retreive last four characters of the string
```

```
N
```

In [67]:
```
print(str1[-6:])  # Retreive last six characters of the string
```

```
NANDAN
```

In [68]:
```
print(str1[:4])     # Retreive first four characters of the string n-1
```

```
VIHA
```

In [69]:
```
print(str1[:6])       # Retreive first six characters of the string n-1
```

```
VIHARI
```

Update & Delete String

In [73]:
```
# Strings are immutable which means elements of a string cannot be changed
str1 = 'VIHARI NANDAN'
str1[0:5] = 'chandan'
print(str1)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[73], line 3
      1 # Strings are immutable which means elements of a string cannot be changed
      2 str1 = 'VIHARI NANDAN'
----> 3 str1[0:5] = 'chandan'
      4 print(str1)

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

In [72]:
```
del str1    # Delete a string
print(str1)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[72], line 2
      1 del str1    # Delete a string
----> 2 print(str1)

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

String Concatenation

In [74]:
```
s1 = 'Vihari'
s2 = 'Nandan'
s3 = s1 + s2
print("Name of the Student is ", s3)
```

```
Name of the Student is  VihariNandan
```