20th June Task identifier, datatype

In [1]:

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

keyword

In [2]:

```
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 [3]:

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

Out[3]:

```
35
```

identifiers

In [4]:

```
1var = 10 #identifiers cannot start with a digit
```

```
  Cell In[4], line 1
    1var = 10
    ^
SyntaxError: invalid decimal literal
```

In [6]:

```
val2@=35 #identifier cant use special symbols
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[6], line 1
----> 1 val2@=35

NameError: name 'val2' is not defined
```

In [7]:

```
import = 125 #as import is a keyword it cant be used a varname
```

```
  Cell In[7], line 1
    import = 125 #as import is a keyword it cant be used a varname
           ^
SyntaxError: invalid syntax
```

In [8]:

```
""" Correct way of defining an indentifier
(Identifiers can be a comb of letters in lowercase (a to z) or uppercase
"""
val2 = 10
```

```
In [10]: val_ = 99
```

comments in python

```
In [11]: val1 = 10 #single line comment
```

```
In [12]: #multiple
         #line
         #comment
         val1= 10
```

```
In [13]: '''
         Multiple
         line
         comment
         '''
         val1 = 10
```

```
In [14]: """
         Multiple
         line
         comment
         """
         val1=10
```

Statements

```
In [15]: p = 20
         q = 20
         r = q
         p , type(p), hex(id(p))

Out[15]: (20, int, '0x7fff80732c18')
```

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

Out[16]: (20, int, '0x7fff80732c18')
```

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

Out[17]: (20, int, '0x7fff80732c18')
```

```
In [18]: p = 20
         p = p + 10 # Variable Overwriting
         p

Out[18]: 30
```

variable assignment

In [19]:

```
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
```

mul assignment

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]:

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

```
44 44 44 44
```

datatypes

Numeric

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))
```

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

In [30]:

```
val2 = 96.69 # complex data type
print(val2)
print(type(val2)) # type of object
print(sys.getsizeof(val2)) # size of complex object in bytes
print(val2, " is Float?", isinstance(val2, float))
```

```
96.69
<class 'float'>
24
96.69  is Float? True
```

In [31]:

```
val3 = 10+20j # 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))
```

```
(10+20j)
<class 'complex'>
32
(10+20j)  is Complex? True
```

In [32]:

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

Out[32]:

```
28
```

In [33]:

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

Out[33]:

```
24
```

In [34]:

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

Out[34]:

```
32
```

boolean

In [35]:

```
bool1 = True
```

In [36]:

```
bool2 = False
```

In [37]:

```
print(type(bool1))
```

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

In [38]:

```
isinstance(bool1,bool)
```

Out[38]:

```
True
```

In [40]:

```
bool(0)
```

Out[40]:

```
False
```

In [41]:

```
bool(1)
```

Out[41]:

```
True
```

In [42]:

```
bool(None)
```

Out[42]:

```
False
```

In [43]:

```
bool(False)
```

Out[43]:

```
False
```

String

String Creation

In [46]:

```
str1 = "HELLO PYTHON" #using double quotes
print(str1)
```

```
HELLO PYTHON
```

In [47]:

```
mystr = 'Hello World' #using single quotes
print(mystr)
```

```
Hello World
```

In [48]:

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

```
Hello
            World
```

In [49]:

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

```
Hello
            World
```

In [50]:

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

```
Happy Monday Everyone
```

In [52]:

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

Out[52]:

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

In [53]:

```
len(mystr2)
```

Out[53]:

```
35
```

string indexing

In [54]:

```
str1
```

Out[54]:

```
'HELLO PYTHON'
```

In [56]:

```
str1[0]
```

Out[56]:

```
'H'
```

In [61]:

```
str1[len(str1)-1] #last char in string using len func
```

Out[61]:

```
'N'
```

```
In [62]: str1[-1]
```

```
Out[62]: 'N'
```

```
In [64]: str1[6]
```

```
Out[64]: 'P'
```

```
In [65]: str1[5]
```

```
Out[65]: ' '
```

string slicing

```
In [67]: str1[0:5]
```

```
Out[67]: 'HELLO'
```

```
In [68]: str1[1:]
```

```
Out[68]: 'ELLO PYTHON'
```

```
In [69]: str1[6:12]
```

```
Out[69]: 'PYTHON'
```

```
In [71]: str1[-4:] #to retrieve last 4 letters of a string
```

```
Out[71]: 'THON'
```

```
In [72]: str1[-6:]
```

```
Out[72]: 'PYTHON'
```

```
In [73]: str1[:4]
```

```
Out[73]: 'HELL'
```

```
In [74]: str1[:6]
```

```
Out[74]: 'HELLO '
```

update & delete string

```
In [75]: str1
```

```
Out[75]: 'HELLO PYTHON'
```

```
In [76]: #Strings are immutable which means elements of a string cannot be changed once t
         str1[0:5] = 'HOLAA'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[76], line 2
      1 #Strings are immutable which means elements of a string cannot be changed 
once they are defined
----> 2 str1[0:5] = 'HOLAA'

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

In [78]:

```
del str1 #delete a string
print(str1)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[78], line 1
----> 1 del str1 #delete a string
      2 print(str1)

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

string concatenation

In [79]:

```
str1 = 'Hello'
str2 = 'World'
str3 = str1 + str2 #concat
print(str3)
```

```
HelloWorld
```

In [ ]: