Variables

Variables are used to store data which can be used just by calling the variable name.

In [4]:
```
num  = 10892345345348563
```

In [5]:
```
print(num)
```

```
10892345345348563
```

In [8]:
```
_n1  = 10
1N = 20      # variable names cannot start with any special character and nu

print(n1)
```

```
  File "C:\Users\BHUPEN~1\AppData\Local\Temp/ipykernel_14420/3650129820.py", li
ne 2
    1N = 20
     ^
SyntaxError: invalid syntax
```

In [10]:
```
first_name = 'Shravan'     # you cannot give any space in variable names
```

In [11]:
```
first_name
```

Out[11]:
```
'Shravan'
```

In [ ]:
```
num = 'Shravan'
```

In [13]:
```
n = 10.67

type(n)
```

Out[13]:
```
float
```

Type Casting

- technique of converting one data type into another

In [15]:
```
a = 10
f = float(a)    # type casting
```

In [16]:

```
f = 34.567
int(f)
```

Out[16]:

```
34
```

In [20]:

```
b = False
type(b)
```

Out[20]:

```
bool
```

In [21]:

```
int(b)
```

Out[21]:

```
0
```

In [26]:

```
num =0
bool(num)
```

Out[26]:

```
False
```

In [35]:

```
num = int(input("Enter a number : "))    # type casting
print(num*2)
```

```
Enter a number : 464
928
```

In [31]:

```
print("Hello")
```

```
Hello
```

Sequence Type

Strings

In [37]:

```
'This is a string'
```

Out[37]:

```
'This is a string'
```

In [40]:

```
"This is a string"
```

Out[40]:

```
'This is a string'
```

In [41]:

```
"""This is also a string"""
```

Out[41]:

```
'This is also a string'
```

In [43]:

```
"This is Shyam's car"
```

Out[43]:

```
"This is Shyam's car"
```

In [46]:

```python
print("""This is line 1
This is line 2
THis is line 3""")
```

```
This is line 1
This is line 2
THis is line 3
```

In [49]:

```python
movie = "SHERSHAAH"

len(movie), type(movie)
```

Out[49]:

```
(9, str)
```

In [51]:

```python
movie = "SHERSHAAH"

print(len(movie))
print(type(movie))
```

```
9
<class 'str'>
```

Indexing

In [57]:

```python
movie = "SHERSHAAH"
movie[8], movie[-1]    # indexing
```

Out[57]:

```
('H', 'H')
```

Slicing [start:end:step]

In [59]:

```python
movie = "SHERSHAAH"

movie[0:4:1]
```

Out[59]:

```
'SHER'
```

In [60]:

```python
movie[4:9:1]
```

Out[60]:

```
'SHAAH'
```

In [62]:

```python
movie[4:]
```

Out[62]:

```
'SHAAH'
```

In [63]:

```python
movie[0::2]
```

Out[63]:

```
'SESAH'
```

In [65]:
```
movie = "SHERSHAAH"

movie[-1::-1]
```

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

In [69]:
```
movie[::-1],movie[::1]
```

Out[69]:
```
('HAAHSREHS', 'SHERSHAAH')
```

String Methods

In [71]:
```
mystr = "pyThon is AmazinGGG"

mystr.capitalize()
```

Out[71]:
```
'Python is amazinggg'
```

In [74]:
```
mystr.upper()
```

Out[74]:
```
'PYTHON IS AMAZINGGG'
```

In [75]:
```
mystr.lower()
```

Out[75]:
```
'python is amazinggg'
```

In [76]:
```
mystr.title()
```

Out[76]:
```
'Python Is Amazinggg'
```

In [77]:
```
mystr
```

Out[77]:
```
'pyThon is AmazinGGG'
```

In [78]:
```
mystr.index('A')
```

Out[78]:
```
10
```

In [79]:
```
mystr[10]
```

Out[79]:
```
'A'
```

In [81]:

```
mystr.index('X')
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
C:\Users\BHUPEN~1\AppData\Local\Temp/ipykernel_14420/4213330218.py in <module>
----> 1 mystr.index('X')

ValueError: substring not found
```

In [82]:

```
mystr
```

Out[82]:

```
'pyThon is AmazinGGG'
```

In [84]:

```
mystr.split(" ")
```

Out[84]:

```
['pyThon', 'is', 'AmazinGGG']
```

In [85]:

```
mystr.split("A")
```

Out[85]:

```
['pyThon is ', 'mazinGGG']
```

In [87]:

```
data = mystr.split(" ")
print(data)
print(len(data))
```

```
['pyThon', 'is', 'AmazinGGG']
3
```

In [93]:

```
data = ["Apple","is","red"]
```

In [94]:

```
"----".join(data)
```

Out[94]:

```
'Apple----is----red'
```

In [95]:

```
f = "apple"
"*".join(f)
```

Out[95]:

```
'a*p*p*l*e'
```

In [96]:

```
# replace

"apple".replace('p','P')
```

Out[96]:

```
'aPPle'
```

In [97]:

```
# isalpha, isnumeric, isalnum

'Apple'.isalpha()
```

Out[97]:

```
True
```

In [98]:
```
'Apple2'.isalpha()
```
Out[98]:
```
False
```

In [101]:
```
"34535".isnumeric()
```
Out[101]:
```
True
```

In [102]:
```
"apple".isalnum()
```
Out[102]:
```
True
```

In [103]:
```
"2345".isalnum()
```
Out[103]:
```
True
```

In [104]:
```
"2sdf345".isalnum()
```
Out[104]:
```
True
```

In [105]:
```
"2345sads$%#".isalnum()
```
Out[105]:
```
False
```

In [106]:
```
s = "2345sads$%#"
s.isalnum()
```
Out[106]:
```
False
```

In [107]:
```
# startswith, endswith

"Apple".startswith('A')
```
Out[107]:
```
True
```

In [108]:
```
"Apple".startswith('a')
```
Out[108]:
```
False
```

In [109]:
```
"Apple".endswith('a')
```
Out[109]:
```
False
```

In [111]:
```
"Apple".endswith('le')
```
Out[111]:
```
True
```

In [112]:
```python
text = "Apple is red"

text.replace('red','green')
```

Out[112]:
```
'Apple is green'
```

In [114]:
```python
# strip,lstrip, rstrip

'Apple       '.strip()
```

Out[114]:
```
'Apple'
```

In [115]:
```python
'  Apple       *'.strip()
```

Out[115]:
```
'Apple       *'
```

In [120]:
```python
'  Apple   #    *'.strip("*# ")
```

Out[120]:
```
'Apple'
```

In [121]:
```python
'  Apple   #    *'.lstrip("*# ")
```

Out[121]:
```
'Apple   #    *'
```

In [122]:
```python
'  Apple   #    *'.rstrip("*# ")
```

Out[122]:
```
'  Apple'
```

In [125]:
```python
"ApPle*%_".strip('Aap*%_')
```

Out[125]:
```
'Ple'
```

In [126]:
```python
"aPPLE"*3
```

Out[126]:
```
'aPPLEaPPLEaPPLE'
```

In [129]:
```python
'aPPLE'+'DFHDFS'
```

Out[129]:
```
'aPPLEDFHDFS'
```

LIST

- []
- is a sequence of elements
- it is mutable/changeable in nature
- ordered
- indexable

```
In [131]: a = [12,10,30,40,True,34.56]

          type(a)

Out[131]: list
```

```
In [132]: a

Out[132]: [12, 10, 30, 40, True, 34.56]
```

```
In [133]: a[4]

Out[133]: True
```

```
In [134]: a[4:]

Out[134]: [True, 34.56]
```

```
In [136]: a[4] = False

          print(a)

          [12, 10, 30, 40, False, 34.56]
```

tuple

```
In [137]: mtup = (24,45)
          type(mtup)

Out[137]: tuple
```

```
In [139]: mtup[1] = 50

---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
C:\Users\BHUPEN~1\AppData\Local\Temp/ipykernel_14420/794863364.py in <module>
----> 1 mtup[1] = 50

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

SETS: { }

In [142]:

```
mohit = {"lakshya","Arshit","Arshit",'Jatin',"Annu"}
ronit = {"Abhishek",'Arshit','Girish','Annu'}

mohit & ronit
```

Out[142]:

```
{'Annu', 'Arshit'}
```

In [143]:

```
mset = {2,10,11,1,'A','b','a'}
mset
```

Out[143]:

```
{1, 10, 11, 2, 'A', 'a', 'b'}
```

In [144]:

```
mset[0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
C:\Users\BHUPEN~1\AppData\Local\Temp/ipykernel_14420/3083543654.py in <module>
----> 1 mset[0]

TypeError: 'set' object is not subscriptable
```

range()

range(start,end,step)

In [150]:

```
list(range(1,11,1))
```

Out[150]:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

In [194]:

```
list(range(2,21,2))
```

Out[194]:

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

In [157]:

```
m = input("Enter a string : ")
m[len(m)//2]
```

```
Enter a string : sjhghfskfjbskjf
```

Out[157]:

```
'k'
```

In [158]:

```
m = input("Enter a string : ")
index = len(m)//2
m[index]
```

```
Enter a string : abcdEfghi
```

Out[158]:

```
'E'
```

In [163]:

```
m = "abcDefg"
mid_index = len(m)//2
mid = m[mid_index]
left = m[0:mid_index]
right = m[mid_index+1:]

print(right,mid,left, sep = "")
```

```
efgDabc
```

In [165]:

```
new_word = right+mid+left
print(new_word)
```

```
efgDabc
```

In [166]:

```
"This is my new string right mid left"
```

Out[166]:

```
'This is my new string right mid left'
```

In [167]:

```
"This is my new string {right} {mid} {left}"
```

Out[167]:

```
'This is my new string {right} {mid} {left}'
```

In [169]:

```
# f-string

f"This is my new string {right}{mid}{left}"
```

Out[169]:

```
'This is my new string efgDabc'
```

In [174]:

```
name = input('Aapka subh naam ? ')
print(f"Hello! {name} {2+3*4}")
```

```
Aapka subh naam ? Sam
Hello! Sam 14
```

In [175]:

```
name = input('Aapka subh naam ? ')
print("Hello!",name)
```

```
Aapka subh naam ? Sam
Hello! Sam
```

In [177]:

```
# take two integers and print the sum using f-string

a = int(input('Enter first num : '))
b = int(input('Enter the second num : '))

print(f"The sum of {a} and {b} is {a+b}")
```

```
Enter first num : 3
Enter the second num : 5
The sum of 3 and 5 is 8
```

In [179]:
```
list(range(20,1,-2))
```

Out[179]:
```
[20, 18, 16, 14, 12, 10, 8, 6, 4, 2]
```

Dictionary

- unordered collection of key:value pairs
- key:values
- unique keys
- {key:value}

In [190]:
```
car = {'model':'safari','brand':'Tata','year':1990, 'year':1991}

car
```

Out[190]:
```
{'model': 'safari', 'brand': 'Tata', 'year': 1991}
```

In [195]:
```
len(car)
```

Out[195]:
```
3
```

In [196]:
```
car['model']
```

Out[196]:
```
'safari'
```

In [192]:
```
car['year']
```

Out[192]:
```
1991
```

In [198]:
```
contacts = {'Rahul':[23647234,"rahul@gmail.com"],
           'Shravan':[234726748,'shravan@yahoo.com'],
           'Bali':[27934283]}
```

In [199]:
```
contacts['Rahul']
```

Out[199]:
```
[23647234, 'rahul@gmail.com']
```

In [206]:
```
contacts['Rahul'][1]
```

Out[206]:
```
'rahul@gmail.com'
```

In [203]:
```
[234,'A','B'][-1]
```

Out[203]:
```
'B'
```

In [205]:

```
[23647234, 'rahul@gmail.com'][0]
```

Out[205]:

```
23647234
```

In [209]:

```
contacts['Shravan'][1] = "shravan@gmail.com"
```

In [210]:

```
contacts
```

Out[210]:

```
{'Rahul': [23647234, 'rahul@gmail.com'],
 'Shravan': [234726748, 'shravan@gmail.com'],
 'Bali': [27934283]}
```

In [ ]:

```

```