Day8_Operators and Control Flow Statements

Special operators or Miscellaneous Operators:

- **Membership Operators**
- **Identity Operators**

Membership Operators : These operators deal with identification of members in a sequence. There are two operators

1. **in : Returns True if the specific element is a part of the collection otherwise False**
2. **not in: Returns True if the specific element is not a part of the collection otherwise False**

Examples

```
var = 10
lst = [8,10,12,14,16]
print(var in  lst) #is var a part of lst
#True

print(var not in  lst)#is var not a part of lst
#False

vr1 = 61
print(vr1 in  lst) #is vr1 a part of lst
#False

print(vr1 not in lst)#is vr1 not a part of lst
#True
```

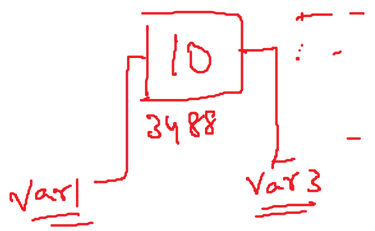
Identity Operators: These operators are used for comparing the id's of the variables.

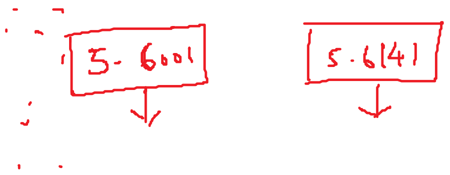
```
var1 = 10
var2 = 61
var3 = 10

id(var1)
#140704964443488

id(var2)
#140704964445120

id(var3)
#140704964443488
```





There are two identity operators

1. **is : return True if the id's of the two elements are same otherwise False**
2. **is not :return True if the id's of the two elements are not same otherwise False**

```
var1 = 10
var2 = 61
var3 = 10

id(var1)
#140704964443488

id(var2)
#140704964445120

id(var3)
#140704964443488

print(var1 is var2)
#False

print(var1 is not var2)
#True

print(var1 is var3)
#True
```

```
print(var1 is not var3)
#False

var4 = 5.6
var5 = 5.6

id(var4)
#2087120021008
id(var5)
#2087120020208
print(var4 is var5)
#False

print(var4 is not var5)
#True


s1 = "Hello"
s2 = "Hello"
id(s1)
#2087120227120
id(s2)
# 2087120227120
print(s1 is s2)
#True

print(s1 is not s2)
#False
```

Control Flow Statements in Python

-The control flow in python will take up a sequential order by default
-The sequential flow is enough for the basic operations but if we want to process the validations, iterations the sequential flow will not cater our needs
-As per the requirement we need to move to other types of Control Flow Statements, they are

1. **Conditional Control Statements**
2. **Loop Control Statements**
3. **Unconditional Control statements**

1. **Conditional Control Statements :**
    1. **These statements of control contain a condition or conditions that helps in validations.**
    2. **The conditions always return either True or False and depending on the condition result the respective statements get executed**
    3. **The Conditional Control Statements in python are**

        1. **if statement**
        2. **if..else statement**
        3. **if..elif statement**
        4. **nested if statements**

The Conditional Control Statements may be effectively used with the dynamic inputs i.e., making the user provide the value from the runtime through console
We have an input() function that will manage to take input values from the console
Syntax:

```
input("User Prompt")
```

Ex:

```
input("Enter any number")

input("Enter any number")
```

output:

```
Enter any number477
'477'
```

By default the input() will get us a string value

```
print("Enter a number")
```

```
input()

Enter a number
977
'977'
```

Conversions of strings to other datatype

```
int(input("Enter any number"))
#744

float(input("Enter any number"))
#4.25

v3,v4 = int(input("Enter a number1:")),int(input("Enter a number2:"))
print(v3)
print(v4)

Enter a number1:10
Enter a number2:56
10
56
```