**Day - 2**

1. **Operators**
2. **Bitwise Operators**
3. **Comparision Operators**
4. **Operator Overloading**
5. **Ternary Operator**
6. **Operator Precedence**
7. **Namespaces**
8. **Decision Making**
9. **Switch Case**
10. **Python Data Structure**

**Operators**

There are many operators in python

**Arithmetic Operators**

>>> a=5

>>> b=9

>>> a+b

14

>>> a-b

-4

>>> a\*b

45

>>> a/b

0.5555555555555556

>>> b%a

4

>>> b\*\*a

59049

>>> b//a

1

>>>

**Relational Operators**

>>> a=5

>>> b=9

>>> a>b

False

>>> a<b

True

>>> a>=b

False

>>> a<=b

True

>>> a==b

False

>>> a!=b

True

>>>

**Assigment Operators**

>>> a=5

>>> b=2

>>> a+=b

>>> a

7

>>> a-=b

>>> a

5

>>> a\*=b

>>> a

10

>>> a/=b

>>> a

5.0

>>> a%=b

>>> a

1.0

>>>a\*\*=b

>>> a

1.0

>>> a//=b

>>> a

0.0

>>>

**Logical Operators**

>>> a=5

>>> b=2

>>> a and b

2

>>> a or b

5

>>> True and False

False

>>> True or False

True

>>> not(5)

False

>>> not(False)

True

>>>

**Membership Operators**

>>> 'hello' in ['hi','hello','bye']

True

>>> 'welcome' in ['hi','hello','bye']

False

>>> 'welcome' not in ['hi','hello','bye']

True

>>>

**Identity Operators**

>>> 20 is 21

False

>>> 2 is 2

True

>>> 2 is '2'

False

>>> 2 is '2.0'

False

>>> 2 is 2.0

False

>>> 2 is not '2'

True

>>>

**Bitwise Operators**

**Bitwise AND (&)**

|  |  |
| --- | --- |
| **0 & 0** | **0** |
| **0 & 1** | **0** |
| **1 & 0** | **0** |
| **1 & 1** | **1** |

>>> True & False

False

>>> '$' & 'abc'

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: unsupported operand type(s) for &: 'str' and 'str'

>>>

**Bitwise OR (|)**

|  |  |
| --- | --- |
| **0 | 0** | **0** |
| **0 | 1** | **1** |
| **1 | 0** | **1** |
| **1 | 1** | **1** |

>>> True | False

True

>>> '$' | 'abc'

TypeError: unsupported operand type(s) for |: 'str' and 'str'

>>>

**Bitwise XOR (^)**

|  |  |
| --- | --- |
| **0 ^ 0** | **0** |
| **0 ^ 1** | **1** |
| **1 ^ 0** | **1** |
| **1 ^ 1** | **0** |

>>> True ^ False

True

>>> True ^ True

False

>>> '$' ^ 'abc'

TypeError: unsupported operand type(s) for ^: 'str' and 'str'

>>>

**Bitwise 1’s Complement**

>>> ~2

-3

>>> bin(2)

'0b10'

>>> bin(-3)

'-0b11'

>>>

**Bitwise Left Shift Operator (<<)**

Add 0s to the empty least significant places

>>> True<<2

4

>>> 2<<1

4

>>> 3<<2

12

>>>

**Bitwise Left Shift Operator (<<)**

Bits will be lost here where as bits will be added in shift left operator

>>> 3>>1

1

>>> 31>>3

3

>>>

**Comparison Operators**

>>> 3>3

False

>>> 3<3

False

>>> 3>=3

True

>>> 3<=3

True

>>> 'Abc'<'abc'

True # because of Ascii value A-65 a-97 so 65<97

>>> 0.999<1

True

>>> (1,2,3)<(1,2,3,4)

True

>>> (1,3,2)<(1,2,3)

False

>>> (1,2,3)<(1,3,2)

True

>>> () < (0,)

True

>>> (1,2) < ('one','two')

TypeError: '<' not supported between instances of 'int' and 'str'

>>> [0]<[False]

False

>>> (1,'one')<(2,'two')

True

>>> {1,2,3}<{1,3,2}

False # because set will rearrange its values as {1,2,3} so it returns False

>>> {1:'one',2:'two'}<{1:'three',2:'four'}

TypeError: '<' not supported between instances of 'dict' and 'dict'

>>> 0.5<False

False

>>> 3,4,5>3,4,5.0

(3, 4, True, 4, 5.0)

>>> 3,4,5 > 3,4,5.0

(3, 4, True, 4, 5.0)

>>> 3,4,5>(3,4,5.0)

TypeError: '>' not supported between instances of 'int' and 'tuple'

>>> (3,4,5)>(3,4,5.0)

False

>>> 3=='3'

False

>>> {1,2,3}=={1,3,2}

True

>>> 0==False

True

>>> 3!=3.0

False

>>> 3==3.0

True

>>>

**Operator Overloading**

Same operator for different purpose

>>> 42+1

43

>>> '42'+'1'

'421'

>>> 'hello'+' '+'world'

'hello world'

>>> [1,2,3]+[4,5,6]

[1, 2, 3, 4, 5, 6]

>>> (1,2,3)+(4,5,6)

(1, 2, 3, 4, 5, 6)

>>>

**Ternary Operator**

It is also known as conditional operator

**Syntax: [true] if [expression] else [false]**

>>> a=5

>>> b=9

>>> result = a if a>b else b

>>> result

9

>>>

There is an another way to achieve this

>>> a=5

>>> b=9

>>> res = (b,a)[a>b]

>>> res

9

>>>

In the above code

* (9,5)[5>9]
* (9,5)[0]
* Ans: 9

**Nested Ternary Operator**

from random import random

a=random()

print("less than 0" if a<0 else "between 0 and 1" if a>=0 and a<=1 else "greater than 1")

Output:

between 0 and 1

**Switch Case**

Python does not have in-built switch case option. But this can be achieved in other ways.

def week(i):

    switcher={

        0:'sunday',

        1:'monday',

        2:'tuesday',

        3:'wednesday',

        4:'thursday',

        5:'friday',

        6:'saturday'

    }

    return switcher.get(i,"Invalid day of week")

print(week(4))

def zero():

    return 'zero'

def one():

    return 'one'

def indirect(i):

    switcher={

        0:zero,

        1:one,

        2:lambda:'two'

    }

    func=switcher.get(i,lambda:'Invalid')

    return func()

print(indirect(2))

print(indirect(5))

**Python Data Structures**

List

Tuple

Set

Dictionary