8/8/2022 Week-3

Monday

PAGE NO.
DATE:

Operators -

used to perform operation on values and variables.

Types -

1) Arithmetic operators
2) Assignment operators
3) Comparision operators
4) Logical operators
5) Identify operators
6) Membership operators

1) Arithmetic operators.

used with numeric values to perform common mathematical operation.

+ → Addition	- → Subtraction
Ex-	Ex
x = 2	x = 7
y = 5	y = 5
print (x+y)	print (x-y)
output - 7	output = 2

PAGE NO.
DATE:

* → multiplication	/ → Division
EX: x = 10 y = 7 print (x * y) output : 70	EX: x = 20 y = 4 print (x / y) output : 5

% → Modules (It produces the remainder when x is divided by y).

EX - x = 5
y = 2
print (x % y)

output = 1

(2) Assignment operator.

= used to perform arithmetic operation while assigning value to a variable

1) = → y = a + b
2) += → m+ = 10 / m = m+10
3) -= → m- = 10 / m = m-10
4) *= → m* = 10 / m = m*10
5) /= → m/ = 10 / m = m/10
6) %= → m% = 10 /m = m%10

ex -

```
m = 15
m += 10
print (m)
```

output = 25

★ Comparision operator / Relational

are used to compare the value of operands (expressions) to produce a logical value. logical value is True OR False

operators -

1) < → less than → 5 < 2 → False
2) > → greater than → 5 > 2 → True
3) <= → less than or equal to → 5 <= 2 → False
4) >= → greater than or equal to → 5 >= 2 → True
5) == → Equal to → 5 == 2 → False
6) != → Not equal to → 5 != 2 → True.

ex -

```
a = 5
b = 2
value = a > b
print (value)
```

output = True

PAGE NO.
DATE:

logical operators -

are used to connect more relational operations to form a complex expression called logical expression.

* A value obtained by evaluting a logical expression is always logical that is True or false.

operators -

and - returns True if both condition are True → x<5 and x<10

or - Return True if one of statement is True → x<5 or x<4

not - Returns the result, returns false if the resulte is True → not(x<5 and x<10)

ex—

```
x = 10
y = 20
print (x == 10 and x < y)
```

output

False.

PAGE NO.
DATE:

9/8/2022
Tuesday

✱ Identify operators -

used to compare the objects.
Compare the memory location of two objects.

is → Returns True if both variables are the same object → x is y

is not → Returns True if both variables are not the same object → x is not y

ex-

```
x = ["grapes", "banana"]
y = ["grapes", "banana"]

new = x

print (x is new)
print (x is not new)
```

Output

True
False.

PAGE NO.
DATE:

membership operator.

used for test the presence of substring in a string, also tuples and so dictionaries

in
- used to find an element in the specified sequence.
- It Returns True if element is found in the specified sequence else it Return False.

ex -

```
str1 = "welcom to Home"
print("to" in str1)
```

output True.

(2) Not in
- work in reverse manner
- returns True if element is not found in the specified sequence and if element is found then it returns false.

Ex -

```
str = "welcome to Home"
print("I" not in str1)
```

output -
True

10/8/2022
Wednesday

⁂ String modify

1) upper case

The upper() method return the string in upper case

ex - x = "Hello, Pranali"
print(x.upper())

(2) lower case

The lower() method returns the string in lower case

ex - x = "HELLO"
print(x.lower())

output - hello

(3)

Remove whitespaces

whitespaces is the space before actual text or after the actual text, and you want to remove this space using

strip()

ex-

```
x = " Hello world "
print(x)
print(x.strip())
```

output - Hello world
Hello world.

* Replace string

replace() method replaces a string with another string.

ex- x = " Hello, pranali "
print(x.replace("H", "J"))

output- Jello, pranali.

* Split string

it divides the string into group / list of substring. with using split()

ex- x = "Hello, world"
y = x.split(",")
print(b)

output - 'Hello', 'world'

ex2)

```
x = " Hello World Pranali "
y = x.split(" ", 1)
print(y)
```

output- 'Hello', 'World Pranali'

* String concatenation

- To concatenate, or combine two string using + operator.

ex.

```
a = "Hello"
b = "World"
c = a + b
print(c)
```

* python format strings

We can combine strings and numbers by using the format() method.

format() passed arguments, formats them and place them into the strings where the placeholders {} are.

PAGE NO.
DATE:

ex- age = 18
x = "my name is Pranali, and I am {}"

print(x.format(age))

output - my name is Pranali, and I am 18

* python Booleans

Booleans represent one of two values
- True or False

Here most value are True
* Any string is True, except empty string
* any number is True, except 0
* any Tuples, dictionary, list, set are True except empty ones.

ex - bool
print(bool("abc"))
print(bool(123))
print(bool(["apple", "cherry"]))

output - True
True
True

Some values are false

- empty values - such as (), {}, [], '', the number 0 and value None.
- The value False evaluates to False

ex-
```
print(bool(False))
print(bool(None))
print(bool(0))
print(bool(""))
print(bool([]))
print(bool({}))
print(bool(()))
```

output-
```
False
False
False
False
False
False
False.
```

12/8/22
Friday

Write a program on membership operator

→

```
x = 24
y = 20
list = [10, 20, 30, 40, 50]

if (x not in list):
print ("x is NOT present in list")

else:
    print ("x is present in list")

if (y in list):
  print ("y is present in list")
else:
  print ("y is NOT present in list")
```

output-

x is NOT present in list
y is Present in list

DATE:

* Write a program on arithmetic operator

```
x = 10
y = 20
print (x+y)

print (x-y)

print (x*y)

print (x/y)
```

output:

```
30
-10
200
1.5
```

* write a program on assignment operator.

```
x = 25
y = 5
print (x - y)
```

```
a = 10
b = a    # assign value
print (b)  # assign value.

b += a   # Add and assign value.
print (b)

b -= a   # sub and assign value
print (b)

b *= a   # mul and assign
print (b)
```

output -

```
10
20
10
100
```

Write a program python string Formatting

~~List~~=

x = "my school is {}, I love my {}. my school is {}"

print(x.format("best", "school", "Near"))

output:

my school is best. I love my school. my school is near.