iv) String Slicing. 17-08-22

Python Slicing means obtaining a sub-string from the given string.

Slicing done by- [:]

- For reverse slicing we give the negative numbers.

e.g. `string[0:5]`

v) capitalize()

It converts the first character to upper case and other in lower case.

e.g.
name.capitalize()~~name~~

vi) casefold()

It converts whole string in lower case.

e.g.

name..casefold().

vii) center()

It returns a centered string.

viii) count()

It returns a specified value occurs in a string.

e.g.

ix) encode()

Returns ~~to~~ an encoded version of string.

x> endswith()

Returns true if the string ends with the specific value.

e.g. name.endswith(".")

xi> expandtabs()

Sets the tab size of string.

e.g. txt.expandtabs(2)

xii> find()

Searches the string for a specified value and returns the position of where it was found.

e.g. name.find("Pratiksha")

xiii> format()

It is used to format the string.

e.g. name="My name is {}"
name.format("Pratiksha")

It gives output -
My name is Pratiksha.

Bool function

It is used to return or convert value to Boolean value i.e. True or False, using the truth testing procedure.

syntax -
```
bool([x])
```

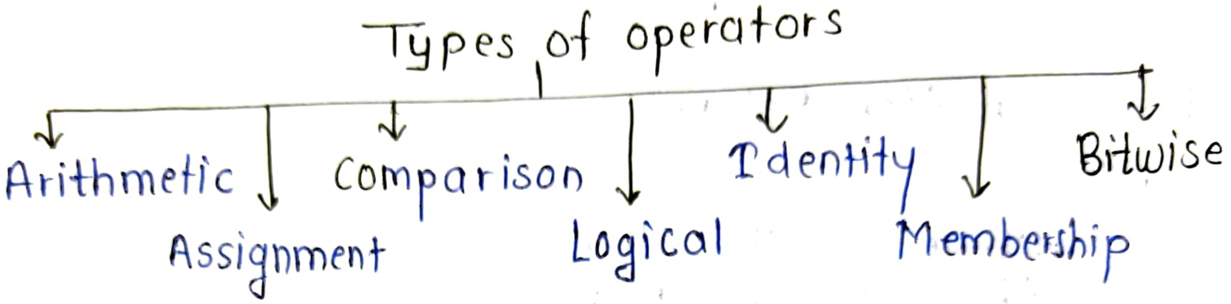
Bool function Returns

i) True
- there is any string
- there is any Number

ii) False
- empty string is given
- 0 in the parantesis
- ~~emp~~ blank is in the paranthesis

Operators in Python

- Operators are entity which is used to manipulate values and variable.
- It used to perform operations on variables and values.



Types of operators

Arithmetic

Assignment

Comparison

Logical

Identity

Membership

Bitwise

1) Arithmetic Operators.

Operator	Name	Example
+	Addition	x+y
-	Substraction	x-y
*	Multiplication	x * y
/	Division	x/y
%	Modulus	x%y
**	Exponentiation	x**y
//	Floor division	x//y

2) Assignment operator

=	used to assign value to variable	x=10
		x=y

* short-hand operators.

operator	Example	equivalant
+=	x+=3	x=x+3
-=	x-=3	x=x-3
*=	x*=3	x=x*3
/=	x/=~~4~~3	x=x/3
**=	x**=3	x=x**3
&=	x&=3	x=x&3
|=	x|=3	x=x|3
^=	x^=3	x=x^3
>>=	x~~<<~~>>=3	x=x>>3
<<=	x<<=3	x=x<<3

3> Comparison operators.

Operator	Name	Example
==	Equal	x==y
!=	Not equal	x!=y
>	Greater than	x>y
<	Less than	x<y
>=	Greater than equal to	x>=y
<=	Less than equal to	x<=y

4> Logical operators..

Operator	Description	Example
Logical AND	Returns TRUE if both statements are true	$x<5$ and $x<10$
Logical OR	Returns TRUE if one of the statement is true	$x<5$ or $x<4$
Logical NOT	Reverse the result, returns False if the result is true	not ($x<5$ and $x<10$)

5> Identity operators.

Operator	Description	Example
is	Returns true if both variable are same object	x is y
is not	Returns true is both variables are not same object	x is not y

6)Membership operator.

Operator	Description	Example
in	Returns True if a sequence with specified value is present in object	x in y
not in	Returns True if a sequence with specified value is not present in the object	x not in y

7) Bitwise Operator.

Operator	Description	Example
&	Bitwise AND	x&y
|	Bitwise OR	x|Y
~	Bitwise NOT	~x
^	Bitwise XOR	x^y
>>	Bitwise right shift	x>>
<<	Bitwise left shift	x<<

Python Lists

List are collection of different or same values in a single variable.

e.g.

① colours= ["Red", "Pink", "Black"] .
   print (colours)

② lists= [1, "Raju", "Jadhav". "Pune", 123]
   print (lists)

- List length.
  function- len()
  e.g. print (len (lists)

  output - 5

- List type
  function- type()
  e.g. type(lists)

  output - < class list>

* List() constructor.

  e.g.

  lists = list (("apple", "banana", "cherry"))
  # double round bracket
  print (lists).