OPERATORS

COMPARISON OPERATORS

Operators	Description	Example
==	If the values of the two operands are equal, then the condition becomes true	(4 == 4) is true
!=	If values of the two operands are not equal, then the condition becomes true	(2!= 4) is true
>	If the value of the left operand is less than the value of the right operand, then the condition becomes false	(2 > 4) is false
<	If the value of the left operand is less than the value of the right operand, then the condition becomes true	(2 < 4) is true
>=	If the value of the left operand is less than or equal to the value of the right operand, then the condition becomes false	(2 >= 4) is false
<=	If the value of the left operand is less than or equal to the value of the right operand, then the condition becomes true	(2 <= 4) is true

LOGICAL OPERATORS

OPERATOR	DESCRIPTION	EXAMPLE
and (Logical AND)	If both the operands are true, then condition becomes true	(*True* and *False*) is False
or (Logical OR)	If any of the two operands are non-zero then condition becomes true	(*True* or *False*) is True
not (Logical NOT)	Used to reverse the logical state of its operand	not(*False*) is True not(true) is false

ASSIGNMENT OPERATORS

Operator	Description	Example
`=`	Assigns values from the right side operands to the left side operands	`x = a + b` assigns value of a + b into x
`+=` (Add AND)	Adds the right operand to the left operand and assigns the result to the left operand	`x += a` is equivalent to `x = x + a`
`-=` (Subtract AND)	Subtracts the right operand from the left operand and assigns the result to the left operand	`x -= a` is equivalent to `x = x - a`
`*=` (Multiply AND)	Multiplies the right operand with the left operand and assigns the result to the left operand	`x *= a` is equivalent to `x = x*a`
`/=` (Divide AND)	Divides the left operand with the right operand and assigns the result to the left operand	`x /= a` is equivalent to `x = x / a`
`%=` (Modulus AND)	Takes the modulus using two operands and assigns the result to left operand	`x %= a` is equivalent to `x = x % a`
`**=` (Exponent AND)	Performs exponential (power) calculation on operators and assigns value to the left operand	`x **= a` is equivalent to `x = x ** a`
`//=` (Floor Division)	Performs the floor division on operators and assigns value to the left operand	`x //= a` is equivalent to `x = x // a`

MEMBERSHIP OPERATORS

Operator	Description	Example
in	Evaluates to true if it finds a variable in the specified sequence and false otherwise	`x` in `y`, here in results in a 1 if `x` is a member of sequence `y`
not in	Evaluates to true if it does not finds a variable in the specified sequence and false otherwise	`x` not in `y`, here not in results in a 1 if `x` is not a member of sequence `y`

IDENTITY OPERATORS

Operator	Description	Example
is	Evaluates to true if the variables on either side of the operator point to the same object and false otherwise	`x` is `y`, here is results in `1` if `id(x)` equals `id(y)`
is not	Evaluates to false if the variables on either side of the operator point to the same object and true otherwise	`x` is not `y`, here is not results in `1` if `id(x)` is not equal to `id(y)`

input() function - to allow a user to take custom inputs
input([prompt]), prompt is the string we want to display into the screen and is optional
Regardless of the type of input we enter, the input is always taken in the form of a string.