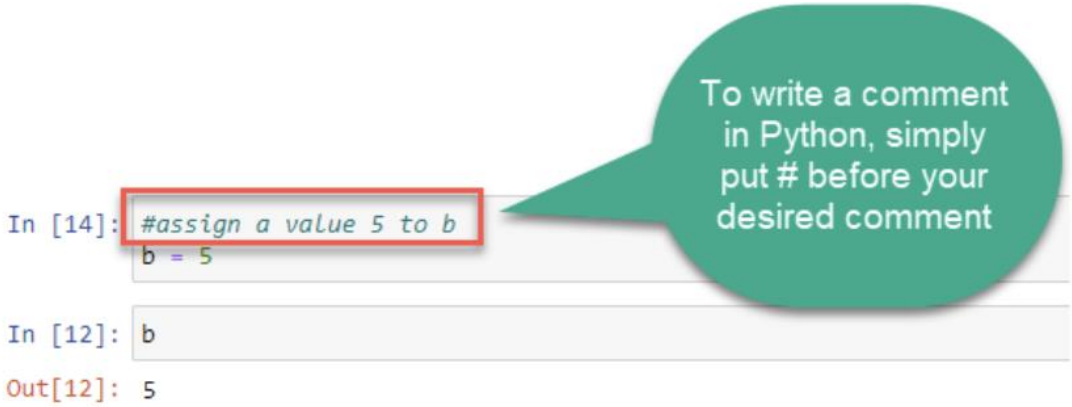
Comments

- When writing code in Python, it's important to make sure that your code can be easily understood by others, say by your friend who wants to see your code.
- Python ignores everything after the hash mark and up to the end of the line. You can insert them anywhere in your code!
- A shortcut for adding comments is by using CTRL + /



Operators

- Operators are used to perform simple operations such as additions, comparisons etc on variables and values.
- Python supports the following types of operators, we will be looking at some of the commonly used operators.

- Arithmetic Operators
- Comparison (Relational) Operators
- Assignment Operators
- Logical Operators
- Bitwise Operators
- Membership Operators
- Identity Operators

Arithmetic Operator

We can perform some basic arithmetic operations such as additions, subtraction, multiplication etc using python
Example: x = 2 y = 3 x + y # addition x * y # multiplication

```
In [2]: 4**2 #Exponentiation
Out[2]: 16

In [3]: 18%7 #Modulo
Out[3]: 4
```

Arithmetic Operators, Meanings & Examples

Operator	Meaning	Example
+	Add two operands or unary plus	x + y+ 2
-	Subtract right operand from the left or unary minus	x - y- 2
*	Multiply two operands	x * y
/	Divide left operand by the right one (always results into float)	x / y
%	Modulus - remainder of the division of left operand by the right	x % y (remainder of x/y)
//	Floor division - division that results into whole number adjusted to the left in the number line	x // y
**	Exponent - left operand raised to the power of right	x**y (x to the power y)

Comparison Operator

- These are used to compare two values
- Gives a boolean result (True/False)

Numeric Calculations:

```
In [1]: 2 < 3
Out[1]: True

In [2]: 2==3
Out[2]: False

In [3]: 2 <= 3
Out[3]: True
```

Other Comparisons:

```
In [4]: "rahul"< "rohan"
Out[4]: True
```

Comparison Operators, Meanings & Examples

Operator	Meaning	Example
>	Greater than - True if left operand is greater than the right	x > y
<	Less than - True if left operand is less than the right	x < y
==	Equal to - True if both operands are equal	x == y
!=	Not equal to - True if operands are not equal	x != y
>=	Greater than or equal to - True if left operand is greater than or equal to the right	x >= y
<=	Less than or equal to - True if left operand is less than or equal to the right	x <= y

Logical Operator

- Logical operators are used to combine conditional statements
- Gives a boolean result (True/False)

```
In [1]: x = 6
        x < 5 and  x < 10
Out[1]: False

In [3]: x < 10 or x < 4
Out[3]: True

In [4]: not(x < 5 and x < 10)
Out[4]: True
```

Operator	Description
and	Returns True if both statements are true
or	Returns True if one of the statements is true
not	Reverse the result, returns False if the result is true