



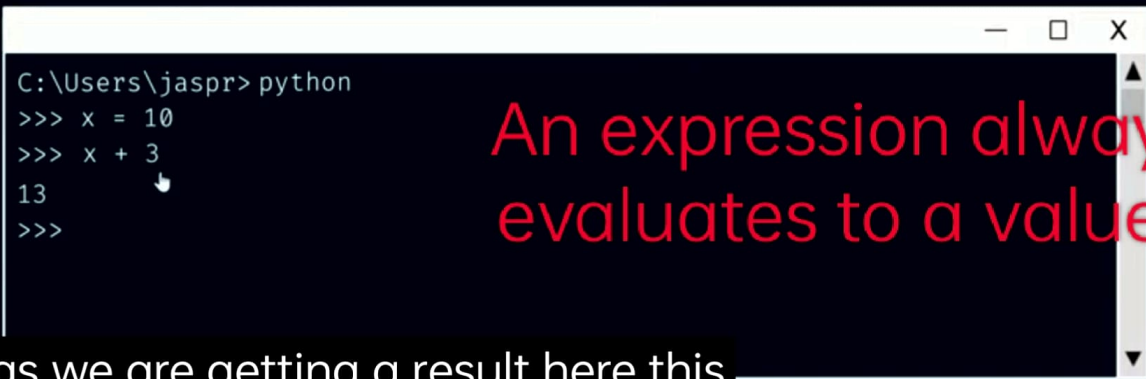
Topics

- ☐ Expressions in Python
- ☐ Constant Expressions
- ☐ Arithmetic Expressions
- ☐ Integral Expressions
- ☐ Floating-point Expressions



Expressions in Python

- Combination of **operators** and **operands**.



```
C:\Users\jaspr> python
>>> x = 10
>>> x + 3
13
>>>
```

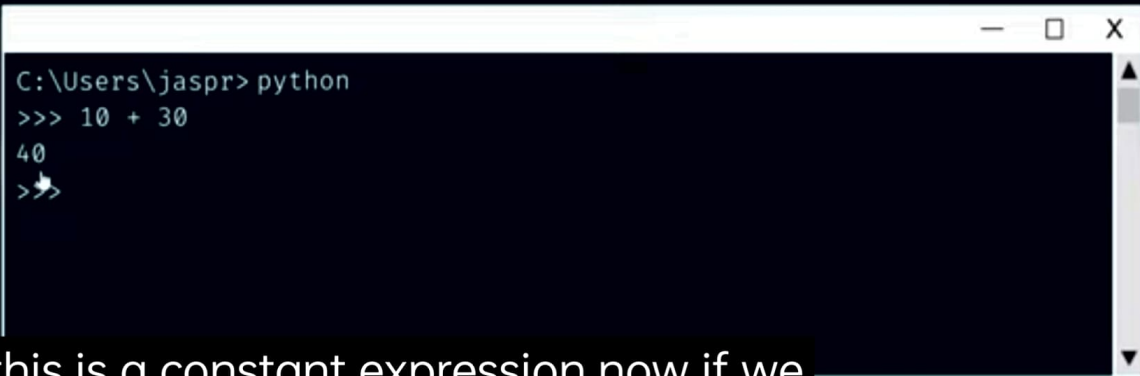
An expression always evaluates to a value

as we are getting a result here this means that $x + 3$ is an expression



Constant Expressions

- A constant expression has **only constants** as operands.

A screenshot of a Windows command prompt window. The title bar shows standard window controls (minimize, maximize, close). The command prompt shows the following text:

```
C:\Users\jaspr> python
>>> 10 + 30
40
>>>
```

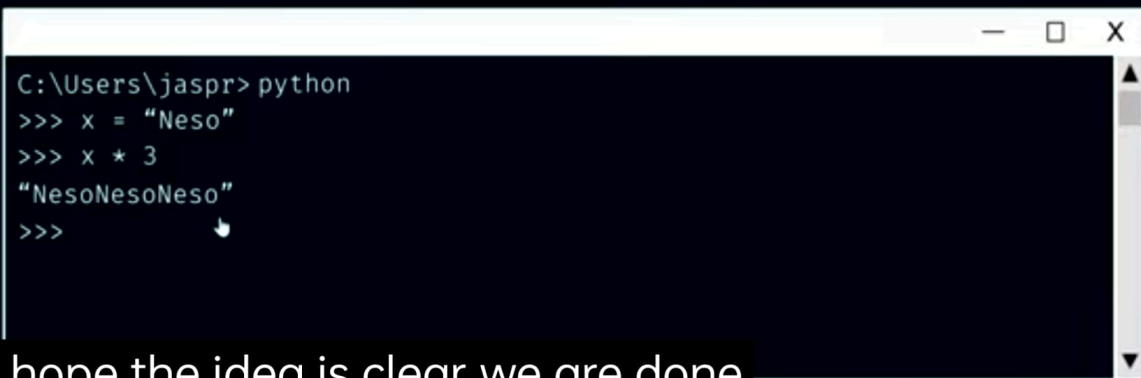
A mouse cursor is visible over the second prompt line. The window has a vertical scrollbar on the right side.

this is a constant expression now if we hit enter we will get 40 as a result



Arithmetic Expressions

- Contains **numeric values** or **strings** as operands, **arithmetic operators** and sometimes **parentheses**.

A screenshot of a Windows command prompt window with a white title bar and standard minimize, maximize, and close buttons. The prompt shows the user running 'python' in the directory 'C:\Users\jaspr>'. The Python interpreter then executes three lines: '>>> x = "Neso"', '>>> x * 3', and '>>>'. The output of the third line is '"NesoNesoNeso"', displayed on the line immediately following the prompt. A mouse cursor is visible on the line '>>>'.

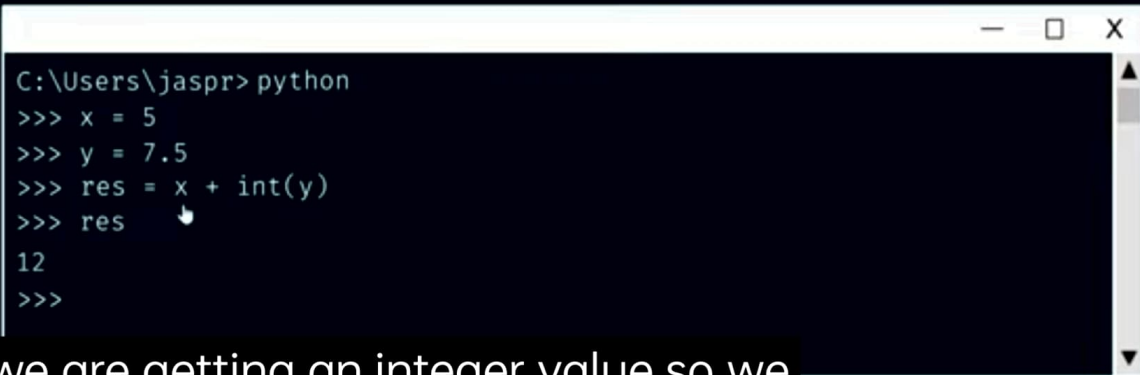
```
C:\Users\jaspr> python
>>> x = "Neso"
>>> x * 3
"NesoNesoNeso"
>>>
```

I hope the idea is clear we are done
with arithmetic Expressions as



Integral Expressions

- Results an **integer value** after performing the necessary type conversions.

A screenshot of a Windows command prompt window with a white title bar and standard minimize, maximize, and close buttons. The prompt shows a user running 'python' in the directory 'C:\Users\jaspr>'. The Python interpreter then executes several commands: 'x = 5', 'y = 7.5', and 'res = x + int(y)'. When the user enters '>>> res', the interpreter outputs '12' on the next line, demonstrating that the float value 7.5 is converted to an integer before being added to x.

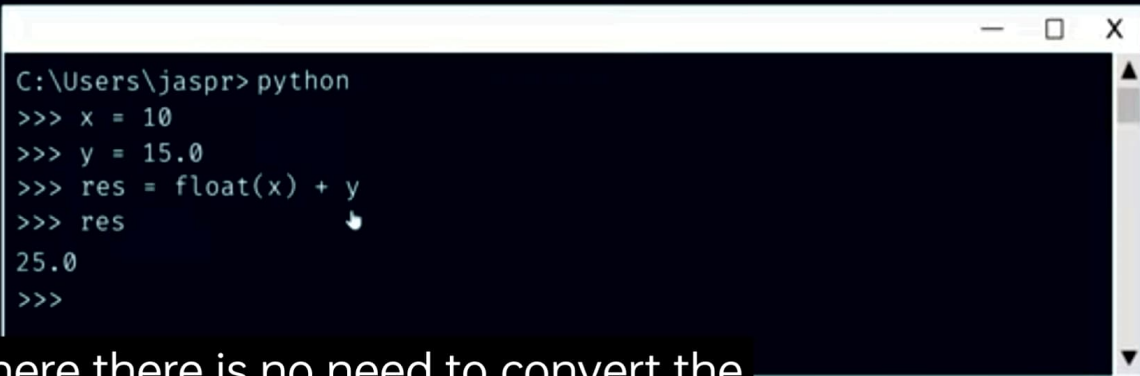
```
C:\Users\jaspr> python
>>> x = 5
>>> y = 7.5
>>> res = x + int(y)
>>> res
12
>>>
```

we are getting an integer value so we
can say



Floating-point Expressions

- Results a **floating-point value** after performing the necessary type conversions.

A screenshot of a Windows command prompt window with a white title bar and standard minimize, maximize, and close buttons. The prompt shows the execution of a Python script. The user enters 'python' at the C:\Users\jaspr> prompt. The Python interpreter then executes several lines: '>>> x = 10', '>>> y = 15.0', and '>>> res = float(x) + y'. The final line shows the result of the expression: '>>> res' followed by '25.0' on the next line, and another '>>>' prompt. A mouse cursor is visible over the space after the last '>>>' prompt.

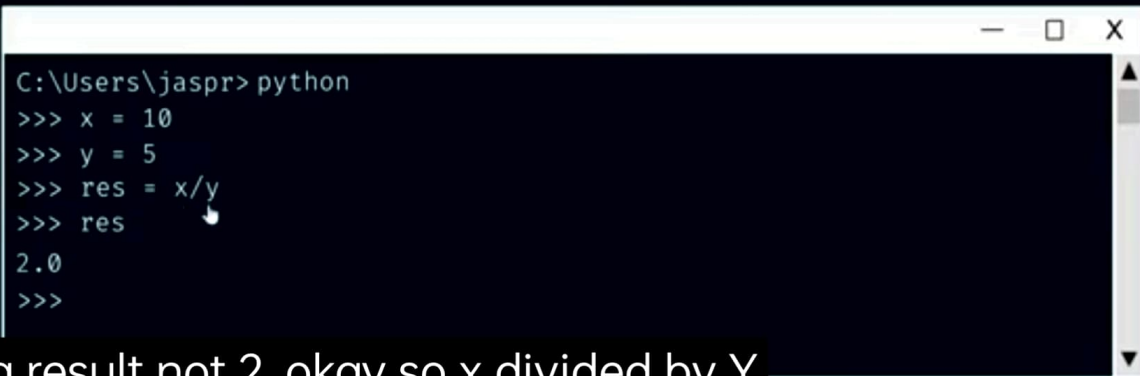
```
C:\Users\jaspr> python
>>> x = 10
>>> y = 15.0
>>> res = float(x) + y
>>> res
25.0
>>>
```

here there is no need to convert the type of X we can directly Type X Plus y **because python does implicit type conversion for x**



Floating-point Expressions

- Results a **floating-point value** after performing the necessary type conversions.



```
C:\Users\jaspr> python
>>> x = 10
>>> y = 5
>>> res = x/y
>>> res
2.0
>>>
```

a result not 2. okay so x divided by Y
is a floating Point expression as it is




Topics

- ☐ Relational Expressions
- ☐ Logical Expressions
- ☐ Bitwise Expressions
- ☐ Combinational Expressions
- ☐ Precedence of Operators



Relational Expressions

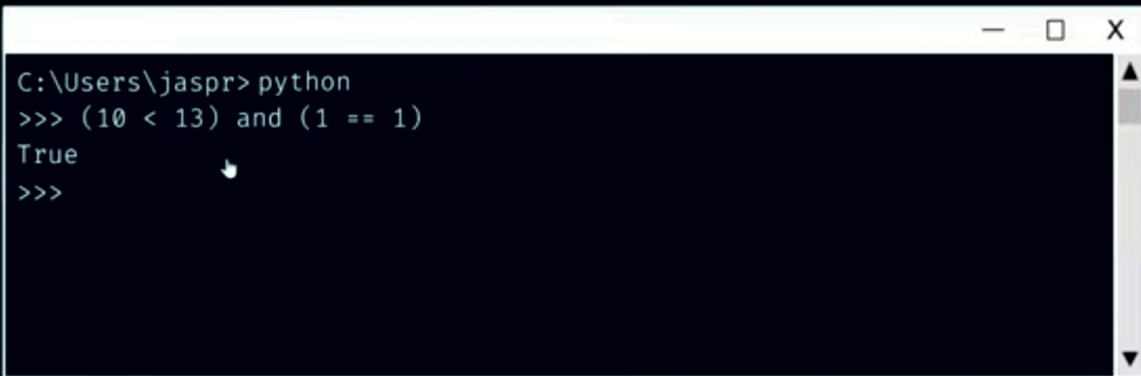
- Also called **Boolean expressions**.
- Returns a **Boolean value**.



```
C:\Users\jaspr> python
>>> (10 + 13) <= (2 + 3)
False
>>>
```

Logical Expressions

- Consists of **relational expressions** connected using **logical operators**.
- Returns a **Boolean value**.



```
C:\Users\jaspr> python
>>> (10 < 13) and (1 == 1)
True
>>>
```



Bitwise Expressions

- Contains Bitwise operators.
- Computations are performed at bit-level.



```
C:\Users\jaspr> python
>>> 10 << 2
40
>>>
```



Precedence of Operators

Highest

Operators
<code>()</code> , <code>[]</code> , <code>{}</code>
<code>**</code>
<code>+a</code> , <code>-a</code>
<code>*</code> , <code>/</code> , <code>//</code> , <code>%</code>
<code>+</code> , <code>-</code>
<code><<</code> <code>>></code>
<code>&</code>

·
·

<code>^</code>
<code> </code>
<code>>=</code> <code><=</code> <code>></code> <code><</code>
<code>!=</code> , <code>==</code>
<code>is</code> , <code>is not</code> , <code>in</code> , <code>not in</code>
<code>not</code>
<code>and</code>
<code>or</code>
<code>=</code> , <code>+=</code> , <code>-=</code> , <code>/=</code> , <code>*=</code>

Lowest



Combinational Expressions

- Combination of different expressions.

```
C:\Users\jaspr> python
>>> x = 10
>>> y = 20
>>> z = y + (x << 1) - x * 3
>>> z
10
>>>
```

