

## 2\_operators

October 17, 2023

### 1 Operators and comparisons

Most operators and comparisons in Python work as one would expect:

- Arithmetic operators `+`, `-`, `*`, `/`, `//` (integer division), `**` power

```
[1]: 1 + 2, 1 - 2, 1 * 2, 5**2
```

```
[1]: (3, -1, 2, 25)
```

```
[2]: 1.0 + 2.0, 1.0 - 2.0, 1.0 * 2.0, 1.0 / 2.0
```

```
[2]: (3.0, -1.0, 2.0, 0.5)
```

```
[7]: # Integer division of float numbers  
5.0 // 2.0
```

```
[7]: 2.0
```

```
[4]: # Note! The power operators in python isn't ^, but **  
2 ** 3
```

```
[4]: 8
```

**Note:** Division

The `/` operator always performs a floating point division in Python 3.x. This is not true in Python 2.x, where the result of `/` is always an integer if the operands are integers. To be more specific, `1/2 = 0.5 (float)` in Python 3.x, and `1/2 = 0 (int)` in Python 2.x (but `1.0/2 = 0.5` in Python 2.x).

The boolean operators are spelled out as the words **and**, **not**, **or**.

```
[8]: not False
```

```
[8]: True
```

```
[9]: not False
```

```
[9]: True
```

```
[12]: True and False
```

```
[12]: False
```

Comparison operators >, <, >= (greater or equal), <= (less or equal), == equality, is identical.

```
[15]: 2 > 1, 2 < 1
```

```
[15]: (True, False)
```

```
[11]: 2 > 2, 2 < 2
```

```
[11]: (False, False)
```

```
[13]: 2 >= 2, 2 <= 2
```

```
[13]: (True, True)
```

```
[12]: # equality  
[1,2] == [1,2]
```

```
[12]: False
```

```
[16]: # objects identical?  
11 = 12 = [1,2]  
  
11 is 12
```

```
[16]: True
```