

## 3\_operators

August 22, 2023

### Arithmetic

```
[1]: 1+2
```

```
[1]: 3
```

```
[2]: 3-2
```

```
[2]: 1
```

```
[4]: 3*2
```

```
[4]: 6
```

```
[5]: 3**2
```

```
[5]: 9
```

```
[6]: 4/2
```

```
[6]: 2.0
```

```
[7]: 4//2
```

```
[7]: 2
```

```
[8]: 4%3
```

```
[8]: 1
```

### Comparison operators

```
[9]: 1==2
```

```
[9]: False
```

```
[10]: 1!=2
```

```
[10]: True
```

```
[11]: 1<2
```

```
[11]: True
```

```
[12]: 2>3
```

```
[12]: False
```

```
[13]: 4<=2
```

```
[13]: False
```

```
[15]: 7>=6
```

```
[15]: True
```

Logical

```
[16]: True and True
```

```
[16]: True
```

```
[20]: 1 and 1
```

```
[20]: 1
```

```
[21]: 1 or 0
```

```
[21]: 1
```

```
[17]: True or False
```

```
[17]: True
```

```
[18]: not True
```

```
[18]: False
```

Bitwise operators

```
[22]: 10 & 10
```

```
[22]: 10
```

```
[25]: bin(10)
```

```
[25]: '0b1010'
```

1010 & 1010 = 101

[23]: 23 & 3

[23]: 3

[26]: bin(23)

[26]: '0b10111'

[27]: bin(3)

[27]: '0b11'

10111 00011

=00011

[24]: 24 & 5

[24]: 0

[2]: bin(24)

[2]: '0b11000'

[4]: bin(5)

[4]: '0b101'

11000 00101

00000

[5]: 30 | 20

[5]: 30

[7]: ~34

[7]: -35

[8]: bin(34)

[8]: '0b100010'

[10]: bin(-35)

[10]: '-0b100011'

#Shift operators

```
[11]: 30>>2 #Rights shift removing 2 digits in right
```

```
[11]: 7
```

```
[14]: bin(30)
```

```
[14]: '0b11110'
```

11110 -> 111=7

```
[21]: 35<<3 # left shift donot remove in left and add 3 zeros at right
```

```
[21]: 280
```

```
[16]: bin(35)
```

```
[16]: '0b100011'
```

100011 ->10011000

```
[19]: int('0b100011000',2)
```

```
[19]: 280
```

```
[4]: a=10
```

```
[5]: a+=2
```

```
[6]: a
```

```
[6]: 12
```

```
[7]: a-=4
```

```
[8]: a
```

```
[8]: 8
```

```
[ ]:
```