

#18 - Precedence

01/06/2025

- Operator precedence is a set of rules that dictates how Python determines which operands each operator is applicable to.
- Generally two operands for an operator
- There is a unary operator that takes one operand.

$2 + 5$ # Two operands (2 and 5)
not True # Unary: One operand (True)

- Precedence determines the meaning of an expression.
 - High precedence operators are prioritised over those with lower precedence.
 - controls the order of evaluation

$3 + 5 * 7$ # Result: 38

- precedence determines whether the $+$ operator uses 3 and 5 as operands or 3 and 35. Likewise, whether the $*$ operator uses 8 and 7 or 5 and 7.

- $*$, $/$ and $//$ have precedence over $+$ and $-$.
- To override precedence, you need to implement parentheses.

$(3 + 5) * 7$ # Result: 56

- An operator that has higher precedence is said to bind more tightly to its operands. Operands either side of $*$ bind tighter than operands either side of $+$.

01/06/2025

Evaluation Order

```
def value(n):  
    print(n)  
    return(n)
```

```
print(value(3) + value(5) * value(7))
```

```
# 3
```

```
# 5
```

```
# 7
```

```
# 38
```

- This demonstrates expression does not strictly evaluate left-to-right. i.e., $5 * 7$ evaluated before the $+$ operator is applied. So it is $3 + 35$ rather than $8 * 7$.
- In the instance of multiple assignments or multiple $**$ operators, right-to-left operation occurs.

```
a = b = c = 3
```

```
5 ** 3 ** 2 # 1953125 (same as 5 ** (3 ** 2))
```

- Short-circuit operators **and** and **or** can lead to unexpected behaviour

```
5 and 1/0
```

```
# ZeroDivisionError
```

```
None or 1/0
```

```
# ZeroDivisionError
```


01/06/2025

rearranging the above so $1/0$ is not needed

None and $1/0$	# None as None is falsy
5 or $1/0$	# 5 as 5 is truthy

short-circuiting

- $1/0$ never gets executed as in **and** operations, the operation ceases as soon as an operand is determined to be **falsy**.

In an **or** operation, the operation ceases as soon as an operand is determined to be **truthy**.

DO NOT RELY ON PRECEDENCE RULES WHEN MIXING OPERATORS. USE PARENTHESES