# WELCOME

Python Series

# AGENDA

Operators
Arithmetic Operators
Assignment Operators
Comparison Operators
Logical Operators
Bitwise Operators
Identity Operators
Membership Operators
Walrus Operator

# OPERATOR

- Python operators are special symbols that carry out arithmetic or logical computation. The value that the operator operates on is called the operand.

- **5 + 3**: `5` and `3` are the `**operands**` and `+` is the `**operator**`.

- Types of Operators:
  - Arithmetic Operators
  - Assignment Operators
  - Comparison (Relational) Operators
  - Logical Operators
  - Bitwise Operators
  - Identity Operators
  - Membership Operators
  - Walrus Operator

- We will have introduction to operators in this lecture. Each operators will be explained in more depth in its dedicated video lecture.

Python Series

# ARITHMETIC OPERATOR (a=10, b=20)

| Operator | Name | Example |
|----------|------|---------|
| **+** | Addition | a + b = 30 |
| **-** | Subtraction | a – b = -10 |
| ***** | Multiplication | a * b = 200 |
| **/** | Division | b / a = 2 |
| **%** | Modulus | b % a = 0 |
| ****** | Exponent | a**b =10**20 |
| **//** | Floor Division | 9//2 = 4 |

Python Series

# ASSIGNMENT OPERATOR

| Operator | Example | Same As |
|:---:|:---:|:---:|
| = | a = 10 | a = 10 |
| += | a += 20 | a = a + 20 |
| -= | a -= 20 | a = a − 20 |
| *= | a *= 10 | a = a * 10 |
| /= | a /= 50 | a = a / 50 |
| %= | a %= 50 | a = a % 50 |
| **= | a **= 43 | a = a ** 43 |
| //= | a //= 55 | a = a // 55 |
| &= | a &= 55 | a = a & 55 |
| \|= | a \|= 56 | a = a \| 56 |
| ^= | a ^= 56 | a = a ^ 56 |
| >>= | a >>= 56 | a = a >> 56 |
| <<= | a <<= 56 | a = a << 56 |

# COMPARISON OPERATOR (a=10, b=20)

| Operator | Name | Example |
|:---:|:---:|:---:|
| == | Equal | (a == b) is not true. |
| != | Not equal | (a != b) is true. |
| > | Greater than | (a > b) is not true. |
| < | Less than | (a < b) is true. |
| >= | Greater than or equal to | (a >= b) is not true. |
| <= | Less than or equal to | (a <= b) is true. |

Python Series

# LOGICAL OPERATOR

| Operator | Name | Example |
|----------|------|---------|
| **and** | AND | a and b |
| **or** | OR | a or b |
| **not** | NOT | not(a) |

Python Series

# BITWISE OPERATOR

| Operator | Name | Example |
|:---:|:---:|:---:|
| & | AND | a & b |
| \| | OR | a \| b |
| ^ | XOR | a ^ b |
| ~ | NOT | ~a |
| << | Left Shift | a << 3 |
| >> | Right Shift | a >> 3 |

Python Series

# IDENTITY OPERATOR

| Operator | Description | Example |
|----------|-------------|---------|
| **is** | Returns True if both variables are the same object and false otherwise. | a is b |
| **is not** | Returns True if both variables are not the same object and false otherwise. | a is not b |

Python Series

# MEMBERSHIP OPERATOR

| Operator | Description | Example |
| --- | --- | --- |
| **in** | Returns True if a sequence with the specified value is present in the object | 'a' in 'apple' |
| **not in** | Returns True if a sequence with the specified value is not present in the object | 'b' not in 'apple' |

Python Series

# WALRUS OPERATOR

- Also known as the **assignment expression operator.**
- It allows you to assign a value to a variable as part of an expression.
- This operator was introduced in Python 3.8.
- It is knows as "Walrus Operator" due to its resemblance to **the eyes and tusks of a walrus.**



| Operator | Description | Example |
|----------|-------------|---------|
| := | Assigns values within an expression (Python 3.8+). | if (n := len(a_list)) > 5:<br>print(n) |

Python Series

NEXT...

# Arithmetic Operators

Python Series