By Nabatanzi Gorret

Python Operators and Expressions

1. What is an Expression?

An **expression** in Python is any combination of **operands (values, variables)** and **operators (symbols or keywords)** that can be evaluated to produce a result.

Examples:

```
2 + 3          # evaluates to 5
x * y          # result depends on values of x and y
a + b / 2      # operator precedence applies
```

Each expression gives a **value** when Python evaluates it.

2. What is an Operator?

An **operator** is a special symbol that performs an operation on one or more operands.

Example:

```
5 + 3  # '+' is the operator, 5 and 3 are operands
```

By Nabatanzi Gorret

Types of Python Operators

1 Arithmetic Operators

Used to perform mathematical operations.

Operator	Description	Example	Output
+	Addition	5 + 3	8
-	Subtraction	10 - 6	4
*	Multiplication	4 * 2	8
/	Division (float)	8 / 3	2.6667
//	Floor Division	8 // 3	2
%	Modulus (Remainder)	10 % 3	1
**	Exponentiation	2 ** 3	8

Example:

```
a = 10
b = 3
print(a / b)   # 3.3333
print(a // b)  # 3
print(a % b)   # 1
print(a ** b)  # 1000
```

2 Comparison (Relational) Operators

Used to compare two values, they return **True** or **False**.

Operator	Description	Example	Output
==	Equal to	5 == 5	True
!=	Not equal to	5 != 3	True
>	Greater than	7 > 4	True
<	Less than	2 < 5	True
>=	Greater than or equal	6 >= 6	True

`<=`	Less than or equal	`3 <= 7`	`True`

Example:

```
x = 8
y = 10
print(x < y)   # True
print(x == y)  # False
```

3 Logical Operators

Used to combine multiple conditions.

Operator	Description	Example	Output
and	True if both conditions are true	`5 > 2 and 3 < 4`	`True`
or	True if at least one condition is true	`5 > 10 or 4 == 4`	`True`
not	Reverses the result	`not(5 > 3)`	`False`

Example:

```
a = 5
b = 10
print(a > 2 and b < 15)   # True
print(a < 2 or b == 10)   # True
print(not(a == 5))        # False
```

4 Assignment Operators

Used to assign values to variables.

Operator	Example	Equivalent To
`=`	`x = 5`	assigns 5 to x
`+=`	`x += 3`	`x = x + 3`
`-=`	`x -= 2`	`x = x - 2`
`*=`	`x *= 4`	`x = x * 4`

/=	`x /= 2`	`x = x / 2`
//=	`x //= 2`	`x = x // 2`
%=	`x %= 2`	`x = x % 2`
****=**	`x **= 2`	`x = x ** 2`

Example:

```
x = 10
x += 5
print(x)   # 15
```

5 Bitwise Operators

Used to perform operations on **binary numbers**.

Operator	Description	Example
&	AND	`5 & 3 → 1`
`	**`**	OR
^	XOR	`5 ^ 3 → 6`
~	NOT	`~5 → -6`
<<	Left Shift	`5 << 1 → 10`
>>	Right Shift	`5 >> 1 → 2`

6 Membership Operators

Used to check if a value exists in a sequence (list, string, tuple).

Operator	Description	Example	Output
in	True if value found	`'a' in 'apple'`	`True`
not in	True if value not found	`'z' not in 'apple'`	`True`

7 Identity Operators

Used to compare **memory locations** of objects.

Operator	Description	Example	Output

is	True if same object	`x is y`	`True / False`
is not	True if not same object	`x is not y`	`True / False`

Example:

```
a = [1, 2, 3]
b = a
c = [1, 2, 3]

print(a is b)  # True
print(a is c)  # False
```

🔢 3. Operator Precedence

Determines the **order** in which operations are executed.

Priority	Operator Type	Example
1	Parentheses `()`	`(2 + 3) * 4`
2	Exponentiation `**`	`2 ** 3`
3	Multiplication, Division, Floor, Modulus `* / // %`	`10 / 5 * 2`
4	Addition, Subtraction `+ -`	`5 + 3 - 2`
5	Comparison `> < == !=`	`a > b`
6	Logical `and`, `or`, `not`	`x > 3 and y < 5`

4. Expressions in Action

Let's combine operators into expressions.

```
x = 10
y = 5
z = 2

result = (x + y) * z / (x - y)
```

```
print(result)
```

Step-by-step evaluation:

1. `x + y = 15`
2. `x - y = 5`
3. `15 * z = 30`
4. `30 / 5 = 6.0`

Final Output: `6.0`

5. Practice Questions

A. Short Questions

1. What is the difference between = and ==?
2. What is the output of `10 % 3`?
3. Write an example using the `not` operator.
4. What will this print be?

```
a = 4
b = 2
print(a ** b // b)
```

B. Coding Practice

1. **Area and Perimeter of a Rectangle**

```
length = 8
width = 3
area = length * width
perimeter = 2 * (length + width)
print("Area:", area)
print("Perimeter:", perimeter)
```

2. **Check if a Number is Even**

```
num = int(input("Enter a number: "))
print(num % 2 == 0)
```

3. **Combine Logical Operators**

```
x = 7
y = 10
print(x > 5 and y < 15)
```

4. **Temperature Check**

```
temp = 30
print(temp > 25 and temp < 35)
```

Summary

- Operators perform actions; expressions combine them to produce values.
- Use parentheses ( ) to control order.
- Always test your code to understand how precedence affects results.