

Bitwise Operator

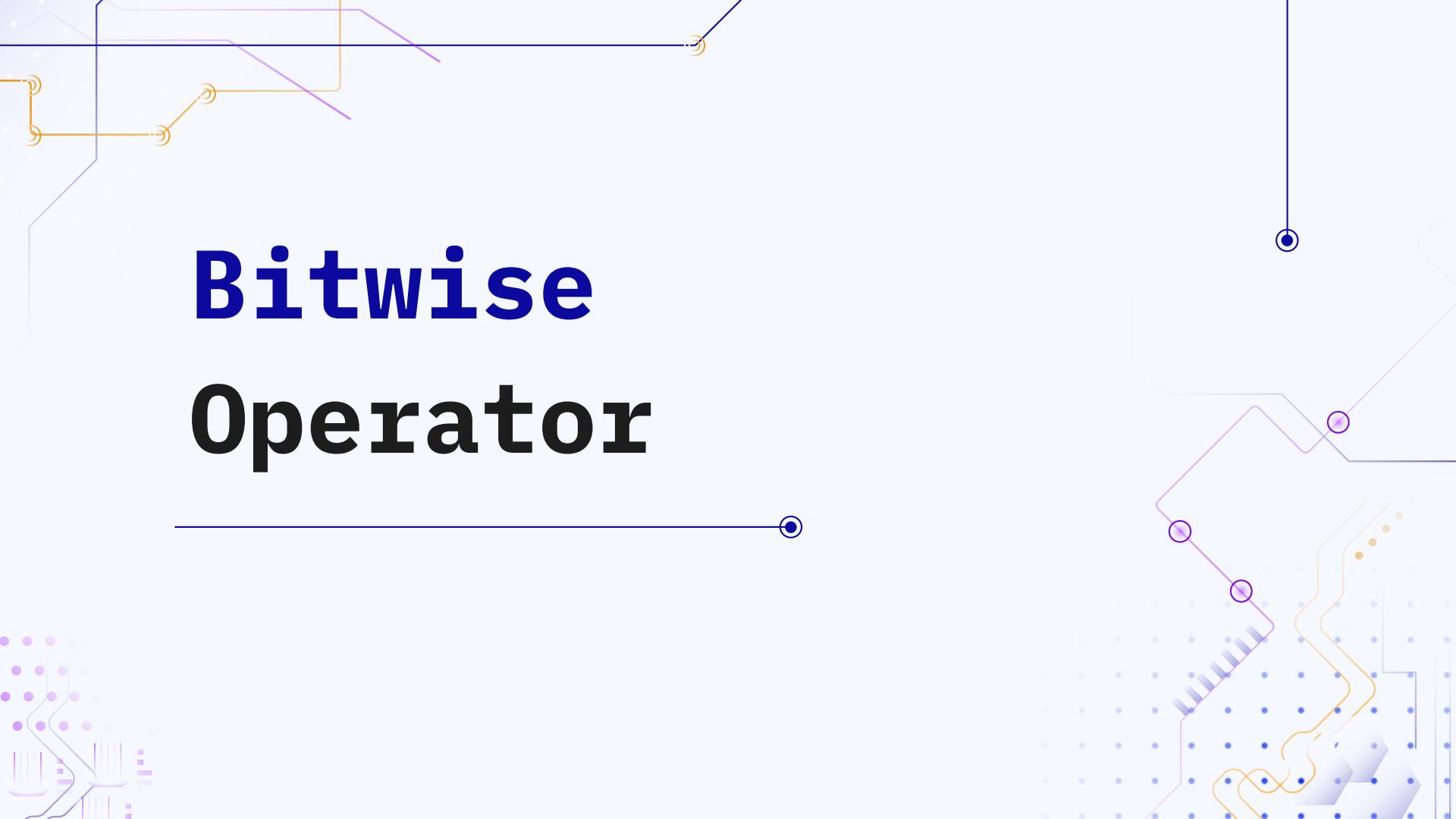




Table of contents

01

Basics of Bits

Basic understanding about bits

02

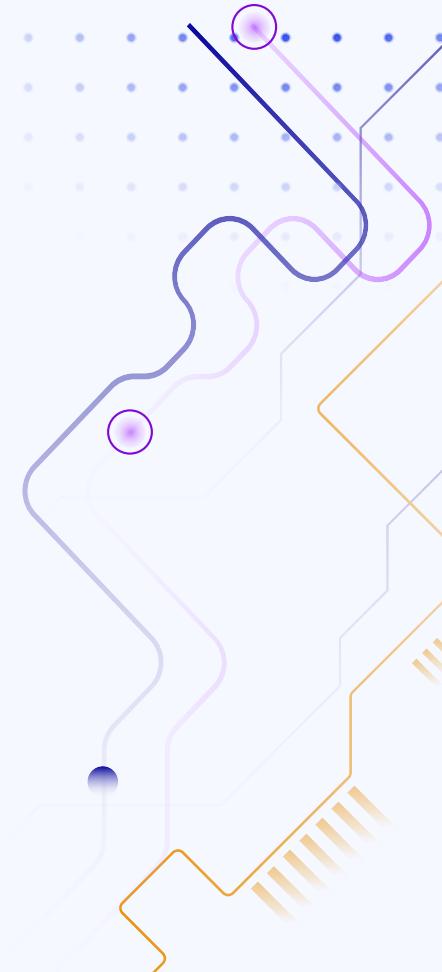
Bitwise Operators

Truth Table and bitwise operations



01

Basics of Bits



What are bits?



Basic Unit of Data

A bit is the most **basic unit** of data in computing.



0 or 1

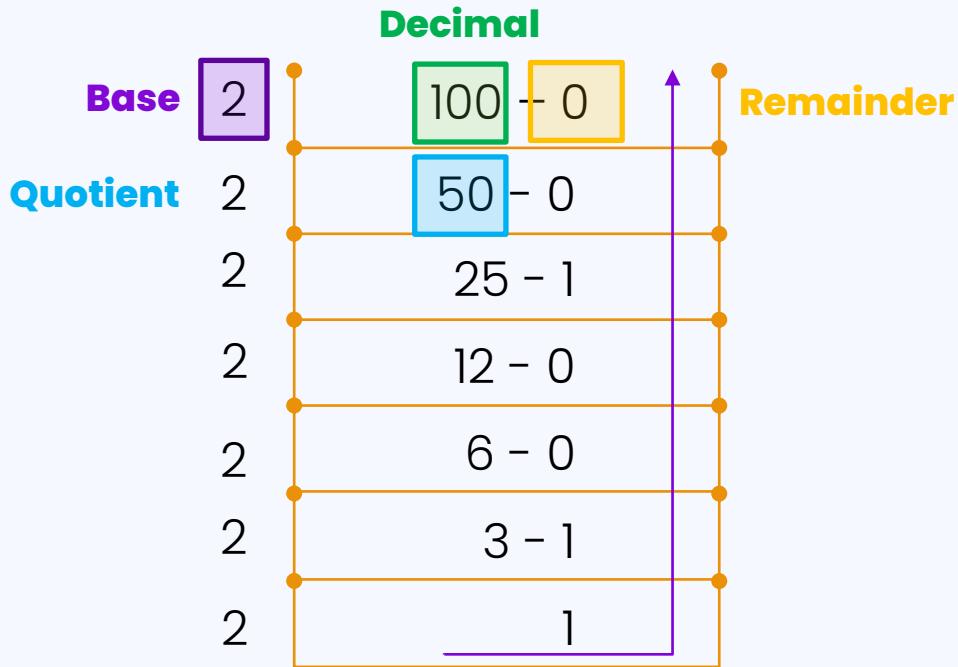
Bits can have a value of either **0 or 1**



It is everywhere

Computers use bits to represent **all types of data**, whether it's numbers, text, images, or any other form of information.

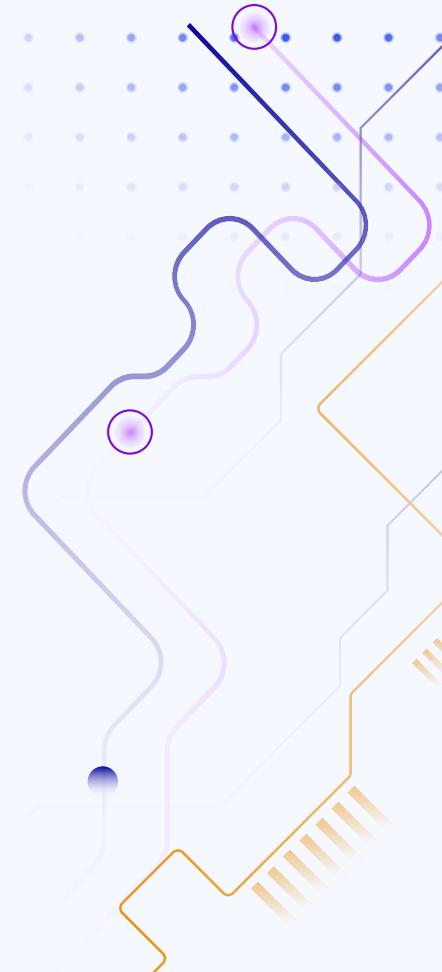
Decimal To Binary Conversion



$$(100)_{10} = (1100100)_2$$

02

Bitwise Operator



AND OPERATOR (&)

Truth Table and Code Implementation

AND Operator (&)

BIT A	BIT B	A & B
0	0	0
0	1	0
1	0	0
1	1	1

Interpretation: The **AND** operator returns **1 only if both bits are 1**. Otherwise, **it returns 0**.

AND Operator (&) – Python Code

Python

```
A = 5 # 0101  
B = 3 # 0011
```

```
Result = A & B
```

Output

```
1      # 0001
```

OR OPERATOR (I)

Truth Table and Code Implementation

OR Operator (|)

BIT A	BIT B	A B
0	0	0
0	1	1
1	0	1
1	1	1

Interpretation: The **OR** operator returns **1 if at least one of the bits is 1**. If **both** bits are **0**, it **returns 0**.

OR Operator (|) - Python Code

Python

```
A = 5 # 0101  
B = 3 # 0011
```

```
Result = A | B
```

Output

```
7      # 0111
```

XOR OPERATOR (\wedge)

Truth Table and Code Implementation

XOR Operator (^)

BIT A	BIT B	A ^ B
0	0	0
0	1	1
1	0	1
1	1	0

Interpretation: The **XOR** operator returns **1** if the **bits are different**, and **0** if they are the **same**.

XOR Operator (^) – Python Code

Python

```
A = 5 # 0101  
B = 3 # 0011
```

```
Result = A ^ B
```

Output

```
6      # 0110
```

NOT OPERATOR (\sim)

Truth Table and Code Implementation

NOT Operator (~)

BIT A	$\sim A$
0	1
1	0

Interpretation: The **NOT** operator **flips the bit**: **0** becomes **1** and **1** becomes **0**.

Note: Applying the NOT operator to an integer **not only flips the bits** but also **changes** the **sign** of the number.

NOT Operator (\sim) - Python Code

Python

```
A = 1 # 0001
```

```
Result = ~A
```

Output

```
-2      # -0b10
```

LEFT SHIFT(<<)



Left Shift Operator (`<<`)

The *Left Shift Operator* **shifts** the bits of a number to the **left** by a **specified number of positions**.

Each shift to the left **doubles** the number, effectively **multiplying the number by 2^n** .

Examples

	1st shift	2nd Shift	3rd Shift	4th Shift
Decimal Representati on	$5 \ll 1 \rightarrow 10$	$5 \ll 2 \rightarrow 20$	$5 \ll 3 \rightarrow 40$	$5 \ll 4 \rightarrow 80$
Binary Representa tion	$101 \ll 1 \rightarrow$ 1010	$101 \ll 2 \rightarrow$ 10100	$101 \ll 3 \rightarrow$ 101000	$101 \ll 4 \rightarrow$ 1010000



RIGHT SHIFT(>>)

Right Shift Operator (`>>`)

The *Right Shift Operator* **shifts** the bits of a number to the **right** by a **specified number of positions**.

Each shift to the left **halves** the number, effectively **dividing the number by 2^n** .

Examples

	1 st shift	2 nd Shift	3 rd Shift	4 th Shift
Decimal Representati on	$5 \gg 1 \rightarrow 2$	$5 \gg 2 \rightarrow 1$	$5 \gg 3 \rightarrow 0$	$5 \gg 4 \rightarrow 0$
Binary Representa tion	$101 \gg 1 \rightarrow 10$	$101 \gg 2 \rightarrow 1$	$101 \gg 3 \rightarrow 0$	$101 \gg 4 \rightarrow 0$

Knowledge Reinforcement

What is the result of the following bitwise operation in Python?

Python

```
a = 12 # 1100 in binary  
b = 5 # 0101 in binary  
result = a & b
```

Answer

- A) 4
- B) 5
- C) 8
- D) 0

What is the result of the following bitwise operation in Python?

Python

```
a = 12 # 1100 in binary  
b = 5 # 0101 in binary  
result = a & b
```

Answer

A) 4

B) 5

C) 8

D) 0

What is the result of the following bitwise operation in Python?

Python

```
a = 7 # 0111 in binary  
result = ~a  
print(result)
```

Answer

- A) -7
- B) -8
- C) 7
- D) 8

What is the result of the following bitwise operation in Python?

Python

```
a = 7 # 0111 in binary  
result = ~a  
print(result)
```

Answer

A) -7

B) -8

C) 7

D) 8

What is the result of the following bitwise operation in Python?

Python

```
a = 7 # 0111 in binary  
result = ~a  
print(result)
```

Answer

A) -7

B) -8

C) 7

D) 8

Which of the following expressions correctly performs a left shift operation on the number 3 to shift its bits by 2 positions to the left?

Answer

- A) $3 \gg 2$
- B) $3 \ll 2$
- C) $3 \& 2$
- D) $3 | 2$

Which of the following expressions correctly performs a left shift operation on the number 3 to shift its bits by 2 positions to the left?

Answer

- A) $3 \gg 2$
- B) $3 \ll 2$**
- C) $3 \& 2$
- D) $3 | 2$

What is the result of the following bitwise operation in Python?

Python

```
a = 10 # 1010 in binary  
b = 4 # 0100 in binary  
result = a | b  
print(result)
```

Answer

A) 2

B) 6

C) 14

D) 12

What is the result of the following bitwise operation in Python?

Python

```
a = 10 # 1010 in binary  
b = 4 # 0100 in binary  
result = a | b  
print(result)
```

Answer

A) 2

B) 6

c) 14

D) 12

What is the result of the following bitwise operation in Python?

Python

```
a = 6 # 0110 in binary  
b = 2 # 0010 in binary  
result = a ^ b  
print(result)
```

Answer

A) 4

B) 6

C) 8

D) 2

What is the result of the following bitwise operation in Python?

Python

```
a = 6 # 0110 in binary  
b = 2 # 0010 in binary  
result = a ^ b  
print(result)
```

Answer

A) 4

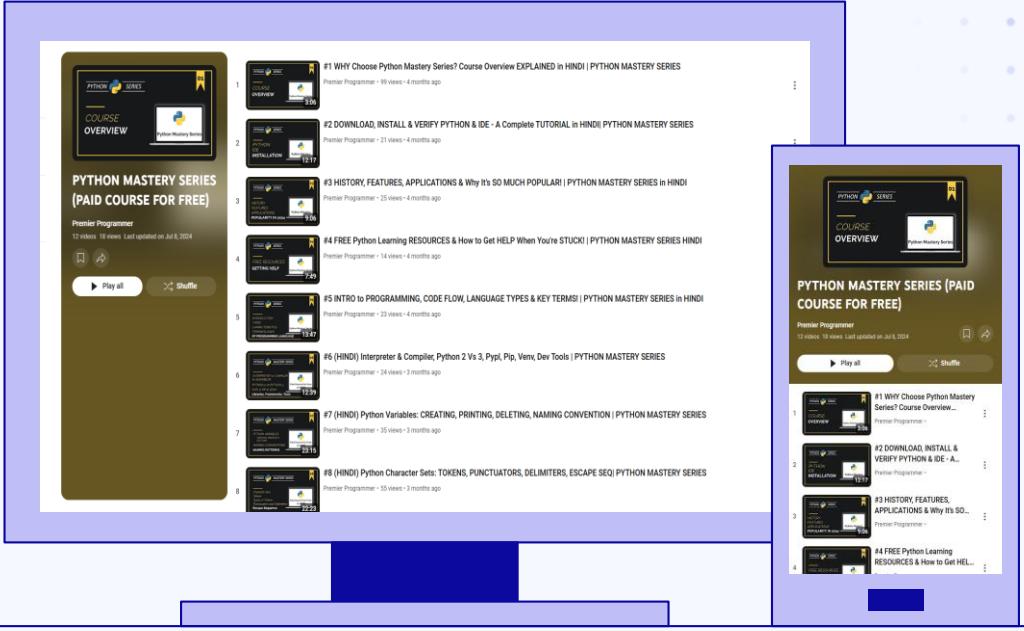
B) 6

C) 8

D) 2

WATCH

Level up your coding with each episode in this focused Python series.



Next Video!

**Assignment Operator -
In Depth**

