Lesson 35 Bitwise Operator OR

In this lesson, we'll discuss the **bitwise operator OR**.

```
int x = 5, y = 6;
int z = x | y;
```

- what does z = x | y mean?

64	32	16	8	4	2	1	
0	0	0	0	1	0	1	→ x in binary
0	0	0	0	1	1	0	→ y in binary
0	0	0	0	1	1	1	→ x | y

The | operator means:

- If **both** bits are **1**, then the result of OR is 1 (true).
- If **any** of the two bits is **1**, then the result of OR is 1 (true).
- If **none** of the two bits is **1**, then the result of OR is 0 (false).

As a result, **z = 7.**

if(true | false)

Sometimes we use the OR operator (| |) in if statements.

If either (or both) of the two conditions are true then it returns true.

In the Bitwise operator, however, we use the (|) operator that works on the binary equivalent of decimal numbers bit by bit instead of the number as a whole.

```
int x = 11, y = 3;
int z = x | y;
```

64	32	16	8	4	2	1
0	0	0	1	1	1	1
0	0	0	0	1	1	1
0	0	0	1	1	1	1

Here, z = 11.

- - > True - True = True

- - > True - false = True