



## lesson 36 Bitwise operator XOR

في هذا الدرس سنتعلم : Bitwise operator XOR

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

```
int z = x ^ y;
```

ما معنى  $z = x \wedge y$  ؟

كيفية عمل علامة  $\wedge$

**False** = إذا كانت ال bits متشابهان

**True** = إذا كانت ال bits مختلفان

بمعنى :

إذا كان هنا 1 و 0 هذا معناه **true** ، إذا يكون الناتج 1

إذا كان هنا 1 و 1 هذا معناه **false** ، إذا يكون الناتج 0

إذا كان هنا 0 و 0 هذا معناه **false** ، إذا يكون الناتج 0

وبالتالي  $z = x \wedge y$  تساوي 3

	<u>64</u>	<u>32</u>	<u>16</u>	<u>8</u>	<u>4</u>	<u>2</u>	<u>1</u>
<code>int x = 5</code>	0	0	0	0	1	0	1
<code>int y = 6</code>	0	0	0	0	1	1	0
<code>int z = x ^ y</code>	0	0	0	0	0	1	1



الصف الثاني هو قيمة  $x$  ب **Binary**  
 الصف الثالث هو قيمة  $y$  ب **Binary**  
 الصف الرابع هو ناتج قيمة  $x \wedge y$

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

```
int z = x ^ y;
```

	<u>64</u>	<u>32</u>	<u>16</u>	<u>8</u>	<u>4</u>	<u>2</u>	<u>1</u>
int x = 11	0	0	0	1	0	1	1
int y = 3	0	0	0	0	0	1	1
int z = x ^ y	0	0	0	1	0	0	0

هنا الناتج سيكون 8

**True ^ false = True**

**True ^ True = false**

**false ^ false = false**

يمكننا من خلال XOR تبديل القيم من  $x$  إلى متغير  $y$  ، المعتاد لعمل مثل هذا البرنامج كنا نقوم بتعريف متغير ثالث و يت

```
int x,y,t;
```

```
t = x;
```

```
x = y;
```

```
y = t;
```

لكن سنتعلم تبديل القيم ب XOR



```
int x = 5, y = 7;
x = x ^ y;
y = x ^ y;
x = x ^ y;
printf("x=%d ,y=%d", x, y);
output:
x=7 ,y=5
```

شرح الكود:

$x = 5, y = 7$

الخطوة الأولى

$x = x \wedge y;$

	<u>64</u>	<u>32</u>	<u>16</u>	<u>8</u>	<u>4</u>	<u>2</u>	<u>1</u>
int x = 5	0	0	0	0	1	0	1
int y = 7	0	0	0	0	1	1	1
int z = x ^ y	0	0	0	0	0	1	0

سيكون الناتج 2 ، إذاً أصبحت قيمة  $x = 2$

الخطوة الثانية :

$y = x \wedge y;$

	<u>64</u>	<u>32</u>	<u>16</u>	<u>8</u>	<u>4</u>	<u>2</u>	<u>1</u>
x = 2	0	0	0	0	0	1	0
y = 7	0	0	0	0	1	1	1



$z = x \wedge y$	0	0	0	0	1	0	1
------------------	---	---	---	---	---	---	---

سيكون الناتج 5 ، إذا أصبحت قيمة  $y = 5$

الخطوة الثالثة :  
أصبحت قيمة

$x = 2$  ,  $y = 5$

$x = x \wedge y;$

	<u>64</u>	<u>32</u>	<u>16</u>	<u>8</u>	<u>4</u>	<u>2</u>	<u>1</u>
$x = 2$	0	0	0	0	0	1	0
$y = 5$	0	0	0	0	1	0	1
$z = x \wedge y$	0	0	0	0	1	1	1

سيكون الناتج 7 ، إذا أصبحت قيمة  $x = 7$  وبالتالي تبدلت القيم.

يمكننا اختصار الكود كالتالي :

```
#include<stdio.h>
```

```
int main( ) {
```

```
    int x = 5, y = 7;
```

```
    x ^= y;
```

```
    y ^= x;
```



---

```
x ^= y;  
printf("x=%d ,y=%d", x, y);  
}
```

( قم بتجربة الكود بنفسك واضغط هنا )