



lesson 35 Bitwise operator OR

في الدرس السابق تعلمنا أن لإجراء عملية ال & بين رقمين، أولاً نقوم بتحويل الرقمين إلى النظام ال binary ثم إجراء عملية ال & بينهم .

و طريقة عمل علامة & هي كالآتي :

إذا كان هناك 1 و 1 متقابلان هذا معناه **true** إذا يكون الناتج (1)
إذا كان هناك 1 و 0 متقابلان هذا معناه **false** إذا يكون الناتج (0)
إذا كان هناك 0 و 0 متقابلان هذا معناه **false** إذا يكون الناتج (0)

في هذا الدرس سنتعلم **Bitwise operator OR** :

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

ما معنى $z = x | y$ ؟

طريقة عمل علامة | هي كالآتي :

إذا كان هناك 1 و 1 متقابلان هذا معناه **true** إذا يكون الناتج (1)
إذا كان هناك 1 و 0 متقابلان هذا معناه **true** إذا يكون الناتج (1)
إذا كان هناك 0 و 0 متقابلان هذا معناه **false** إذا يكون الناتج (0)



	<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 = 6	0	0	0	0	1	1	0
int z = x y	0	0	0	0	1	1	1

الصف الثاني هو قيمة x ب Binary

الصف الثالث هو قيمة y ب Binary

الصف الرابع هو ناتج قيمة x | y

وبالتالي z = x | y تساوى 7

if(true || false)

كنا فى امر if نستخدم احياناً || وهو معناه أنه إذا كان أحد الشرطين أو كلاهما

True يتم تنفيذ ما بداخل ال if

أما إذا كان الشرطين false يكون الناتج false ولا يتم تنفيذ ما بداخل ال if.

كذلك الأمر بالنسبة | فى bitwise لكن نستخدم OR | واحدة فقط وهنا

تتعامل مع bit واحد وليس قيمة x كلها

مثال آخر :

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>
<code>int x = 11</code>	0	0	0	1	0	1	1
<code>int y = 3</code>	0	0	0	0	0	1	1
<code>int z = x y</code>	0	0	0	1	0	1	1

هنا الناتج سيكون 11 ، كما ذكرنا | تتعامل مع كل bit

True | True = True

True | false = True

false | false = false