

## Day7-9按位或运算，按位左移运算

### 整数的按位或运算：(|)

形式：

`$n1 | $n2; // n1,n2是2个任意整数`

含义：

将该2个整数的2进制数字形式（注意，都是32位）的每一个对应位上的数字进行基本按位与运算之后的结果！

注意：他们运算的结果，仍然是一个普通数字（10进制）

图示：（只用8个位来演示）：

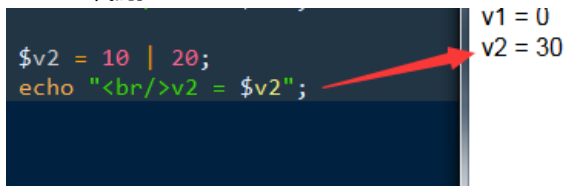
`$r1 = 10 | 20;`

10的2进制	0	0	0	0	1	0	1	0
20的2进制	0	0	0	1	0	1	0	0
运算结果	0	0	0	1	1	1	1	0

则结果该数据值大小为： $1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 = 16 + 8 + 4 + 2 = 30$

代码验证：

```
$v2 = 10 | 20;  
echo "<br/>v2 = $v2";
```



### 整数的按位左移运算

形式：

`$n1 << $m; // n1,n2是2个任意整数`

含义：

将10进制数字n1的2进制数字形式（也是32位）的每一个位上的数字都一次性往左边移动m位，并将右边空出来的位置补0，左边冒出去的不管，这样操作之后得到的结果。

注意：他们运算的结果，仍然是一个普通数字（10进制）

图示：（只用8个位来演示）：

`$r1 = 10 << 2;`

10的2进制	0	0	0	0	1	0	1	0
左移2位后	0	0	1	0	1	0	0	0
运算结果			$2^5$	0	$2^3$			

可见，结果为： $2^5 + 2^3 = 32 + 8 = 40$

代码示例：

```
echo "<br/>";  
$v3 = 10 << 2;  
echo "<br/>v3 = $v3";
```

