PHP小白极速入门 —— 运算符

主讲人与课程设计：耕耕

PHP小白极速入门

运算符(算术，赋值，字符串)

运算符

1. 赋值运算符
2. 算术运算符
3. 字符串运算符
4. 逻辑运算符
5. 位运算符
6. 比较运算符
7. 错误控制运算符
8. 合并运算符
9. 组合比较符
10. 数组运算符
11. 运算符优先级

赋值运算符

基本的赋值运算符是“=”。一开始可能会以为它是“等于”，其实不是的。
它实际上意味着把**右边表达式**的值赋给**左边的运算数**

```
$test = 5;
$a = ($b = 4) + 5; // $a 现在成了 9，而 $b 成了 4。


$c = 3;
$d = &$c; // $d 是 $c 的引用


$e = 100;
$e = $e + 10; // $e +=10;
var_dump($e);
```

算术运算符

还记得学校里学到的基本数学知识吗？就和它们一样

运算符	名称
+	加
-	减
*	乘
/	除
%	取余数
++	累加
--	累减
**	指数
intdiv()	整除

```php
$a = 5;
$b = 2;

var_dump($a + $b);          int(7)
var_dump($a - $b);          int(3)
var_dump($a * $b);          int(10)
var_dump($a / $b);          float(2.5)
var_dump($a % $b);          int(1)
var_dump($a++);             int(5)
var_dump($a--);             int(6)
var_dump($a ** $b);         int(25)
var_dump(intdiv($a, $b));   int(2)
```

算术运算符

我们来看++和--，他们有执行先后的顺序关系

```
$c = $a++;
var_dump($c);    int(5)
var_dump($a);    int(6)
```

++在变量后面，先赋值，再自增

```
$c = ++$a;
var_dump($c);    int(6)
var_dump($a);    int(6)
```

++在变量前面，先自增，再赋值

字符串运算符

有两个字符串（string）运算符。第一个是连接运算符（“.”），它返回其左右参数连接后的字符串

```php
$a = "Hello ";
$b = $a . "World!"; // now $b contains "Hello World!"

$c = "Hello ";
$c .= "World!";

echo $c;
```

PHP小白极速入门

运算符(逻辑，比较)

逻辑运算符

与，或，非，异或，4种形式&& (and)、|| (or)、!、xor
四个符号的**优先级**从高到低是：&&、||、AND、OR

&& (and)：两个都是true返回true；and一样

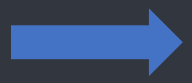
|| (or)：两个都是false返回false；or一样

xor：两个值不一样的时候，返回true

!: 取反，true返回false，false返回true

```php
// foo()短路了
$a = (false && foo());
$b = (true  || foo());
$c = (false and foo());
$d = (true  or  foo());
```

```
(true || false) and (false && true)
```



```
true || false and false && true
```

比较运算符

例子	名称	结果
$a == $b	等于	TRUE，如果类型转换后 $a 等于 $b。
$a === $b	全等	TRUE，如果 $a 等于 $b，并且它们的**类型**也相同。
$a != $b	不等	TRUE，如果类型转换后 $a 不等于 $b。
$a <> $b	不等	TRUE，如果类型转换后 $a 不等于 $b。
$a !== $b	不全等	TRUE，如果 $a 不等于 $b，或者它们的**类型**不同。
$a < $b	小与	TRUE，如果 $a 严格小于 $b。
$a > $b	大于	TRUE，如果 $a 严格大于 $b。
$a <= $b	小于等于	TRUE，如果 $a 小于或者等于 $b。
$a >= $b	大于等于	TRUE，如果 $a 大于或者等于 $b。
$a <=> $b	太空船运算符（组合比较符）	当$a小于、等于、大于$b时 分别返回一个小于、等于、大于0的integer 值。 PHP7开始提供.
$a ?? $b ?? $c	NULL 合并操作符	从左往右第一个存在且不为 NULL 的操作数。如果都没有定义且不为 NULL，则返回 NULL。PHP7开始提供。

比较运算符

```php
$b1 = "123";
$b2 = 123;
var_dump($b1 == $b2);
var_dump($b1 === $b2);
```

```php
$b1 = 10;
$b2 = 9;
var_dump($b1 <=> $b2);
```

```php
$b1;
$b2 = 9;
var_dump($b1??$b2);
```

操作符优先级表

结合方向	运算符	附加信息
无	clone new	clone 和 new
左	[	array()
右	**	算术运算符
右	++ -- ~ (int) (float) (string) (array) (object) (bool) @	类型和递增 / 递减
无	instanceof	类型
右	!	逻辑运算符
左	* / %	算术运算符
左	+ - .	算术运算符和字符串运算符
左	<< >>	位运算符
无	< <= > >=	比较运算符
无	== != === !== <> <=>	比较运算符
左	&	位运算符和引用
左	^	位运算符
左	|	位运算符
左	&&	逻辑运算符
左	||	逻辑运算符
左	??	比较运算符
左	? :	ternary
right	= += - = *= **= /= .= %= &= |= ^= <<= >>=	赋值运算符
左	and	逻辑运算符
左	xor	逻辑运算符
左	or	逻辑运算符

PHP小白极速入门

运算符(二进制)

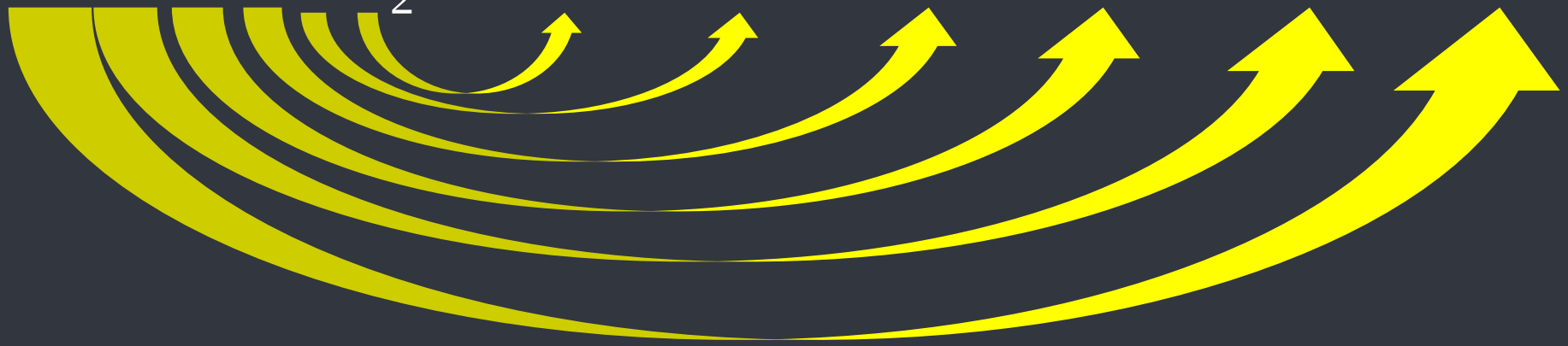
二进制

计算机底层能后识别的是二进制0和1组成的世界，数字电子电路**高频信号**和**低频信号**两种，为了方便就用0和1来表示

二进制转换十进制

转换公式：十进制数 = $d_0\times2^0 + d_1\times2^1 + d_2\times2^2 + \ldots$

$1\ \ 1\ \ 1\ 0\ 0\ 1_2 = 1\times\mathbf{2}^0+0\times\mathbf{2}^1+0\times\mathbf{2}^2+1\times\mathbf{2}^3+1\times\mathbf{2}^4+1\times\mathbf{2}^5 = 57_{10}$



十进制转换二进制

PHP小白极速入门

运算符(位运算)

位运算符

位运算符允许对**整型数**中指定的位**进行求值**和操作

例子	名称	结果
$a & $b	（按位与）	将把 $a 和 $b 中都为 1 的位设为 1。
$a | $b	（按位或）	将把 $a 和 $b 中任何一个为 1 的位设为 1。
$a ^ $b	（按位异或）	将把 $a 和 $b 中一个为 1 另一个为 0 的位设为 1。
~ $a	（按位取反）	将 $a 中为 0 的位设为 1，反之亦然。
$a << $b	（左移）	将 $a 中的位向左移动 $b 次（每一次移动都表示“乘以 2”）。
$a >> $b	（右移）	将 $a 中的位向右移动 $b 次（每一次移动都表示“除以 2”）。

Php中二进制正负数表示

在计算机中，负数以其**正值的补码**形式表达（下面是32位系统下）

00000000000000000000000000000101 转化为十进制为5（**原码**）

11111111111111111111111111111010 1变0，0变1（**反码**）

11111111111111111111111111111011 转化为十进制为-5（**补码**），反码**加1**称为补码

位操作符

```php
$r1 = 3 & 1; // & 上下比较（两个1得1，否则是0）
$r1 = 3 | 1; // | 上下比较（两个0得0，否则是1）
$r1 = 3 ^ 1; // ^ 上下比较（两个不一样得1，否则是0）
$r1 = ~3; // ~ 上下比较（取反）
$r1 = 3<<1; // 左移动1位
$r1 = 3>>1; // 右移动1位
var_dump($r1);
```

000011 (3)

000001 (1)

PHP小白极速入门

运算符(错误与数组)

错误控制运算符

PHP 支持一个错误控制运算符：@。当将其放置在一个 PHP 表达式之前，该表达式可能产生的任何**错误信息都被忽略掉**。

```
$php_errormsg ="错误信息";
$my_file = @file('non_existent_file') or
    die("Failed opening file: error was '$php_errormsg'");
```

数组运算符

```php
$a = array("a" => "apple", "b" => "banana");
$b = array("a" => "pear", "b" => "strawberry", "c" => "cherry");

$c = $a + $b; // Union of $a and $b, 顺序不一样结果不一样
```

```php
$a = array("apple", "banana");
$b = array(1 => "banana", "0" => "apple");

var_dump($a == $b); // bool(true),如果 $a 和 $b 具有相同的键/值对则为 TRUE
var_dump($a === $b); // bool(false), 同时会比较他们的顺序
```

执行运算符(作为了解)

PHP 支持一个执行运算符：反引号（``）。注意这不是单引号!
PHP 将尝试将反引号中的内容作为 shell 命令来执行，并将其输出信息返回

```
$output = `dir`;
echo "<pre>$output</pre>";
```