

php-1.17

//原码就是一个二进制数学，从数学观念上表达出的形式，其中规定
// 一个数字最左边以为是符号位，0表示整数，1表示负数：

反码：

正数的反码就是其本身

负数的反码就是，符号位不变，其他位取反

补码

正数的补码就是其本身

负数的补码就是符号位，其他位取反后+1--就是反码+1

位运算符的应用：管理一组事物的开关状态

很多数据都只有两种结果，对应的就是布尔类型的值

所谓管理一组事物的开关状态，应该理解为其实就是管理若干两个状态的数据符号

使用一个变量，就可以表达若干个数据的当前状态

- 1.通过变量，可以获知任何一个数据的当前状态
- 2.通过该变量，可以将一个个数据的状态的“关闭”
- 3.通过该变量，可以将一个个数据的状态的“开启”

1.单分支的基本语法结构

```
if(条件表达式){  
    语句;
```

```
//....
```

```
//....
```

```
}
```

这里的条件表达式不管多复杂，最总运算的结果是真

双分支

基本语法结构

1.

```
if（条件表达式）{  
    //语句
```

```
}else{
```

```
    //语句
```

```
}
```

2.代码事例

```
$age=15;  
if($age>12){  
    echo "111";  
}else{  
    echo "222";  
}  
echo "程序结束<br/>";
```

多重分支

基本语法结构

```
if(条件表达式){
```

```
    //语句
```

```
}else if(条件表达式){
```

```
    //语句
```

```
}//这里还有更多的else if
```

```
else{
```

)

对上面这个结构请注意：

- 1.else if 可以有一个，也可以有多个
- 2.else 可有可无，但也可以有

快速入门案例：

代码如下

```
$age=20;
if($age>18){
    echo "你的年龄大于18了，要对自己负责了";
}else if($age<=18 && $age>10){
    echo "你要进入少管所";
}else{
    echo "下次注意";
}
```

多分支-switch

基本语法结构

```
switch(表达式){
case 常量 1:
    // 多语句
break;
case 常量2:
    //多语句
break;
case 常量3:
    //多语句
break;
}
```

请注意：

- 1.case 语句有一到多个
 - 2.default 语句可以没有（要根据自己的代码的业务逻辑决定）
default 本身对顺序没有影响
 - 3.break 通常在case语句后，要带上break，表示要推出switch
 - 4.常量的类型（int，float，字符串，布尔型，null）
 - 5.小数也可以进行switch匹配
 - 6.可以使用布尔型，当用布尔类匹配时，会自动转换数据类型
 - 7.如果匹配到一个入口case，则会一直执行，直到遇到break
- 最终的结论是：首先按照case顺序进行配置，如果一个都没有匹配到则执行default语句的内容，直到遇到break，则退出switch