1. 字符串运算符
   1. 字符串运算符就是英文状态下的点(.),在PHP语言中就是一个点 只有链接运算。
2. 赋值运算符
   1. = 普通赋值运算符，为变量和常量赋值，将等号右边的值赋值给等号左边的变量或者常量。
   2. += 加法赋值运算符
   3. -= 减法赋值运算符
   4. \*= 乘法赋值运算符
   5. /= 除法赋值运算符
   6. %= 取模赋值运算符
   7. .= 连接赋值运算符
3. 比较运算符

> 大于

< 小于

>= 大于等于

<= 小于等于

== 等于 只判断值，不判断类型

!= 不等于

=== 全等于 判断值和类型都相等

!== 非全等于 除了全等于剩下都是非全等于

1. 逻辑运算符

逻辑与 && 或者 and

两个值都为真时，结果为真，其余情况都为假

例子：张波洗脸

1. 只洗了左脸(true),没洗右脸(false),结果：没洗完脸(flase)
2. 洗了右脸(true),没洗左脸(false),结果，没洗完脸(false)
3. 左脸没洗(fasle),右脸也没洗(false),结果；没有脸(false)
4. 左脸洗了(true) ，右脸洗了(true),结果洗完脸了(true)

逻辑或 || 或者 or

两个值都为假时结果为假，其余情况都为真

例子：杰哥媳妇丢了

1. 我找到了杰哥媳妇(true),杰哥没找到(false),结果找到杰哥的媳妇(true)
2. 我没找到杰哥媳妇(false),杰哥找到了媳妇(true),结果找到杰哥媳妇了(true)
3. 我(true)和杰哥(true)一起找到了他媳妇，结果：找到他媳妇了(true)
4. 我没找到他媳妇(false),杰哥也没找到他媳妇(false),结果：媳妇和老王比翼双飞(false)

逻辑非 ! 或者 not

真变假、假变真

例子：强哥纯爷们(true),去泰国被人砍了一刀就不那么纯了(false)

逻辑异或 xor

相同为假、不同为真。

例子：

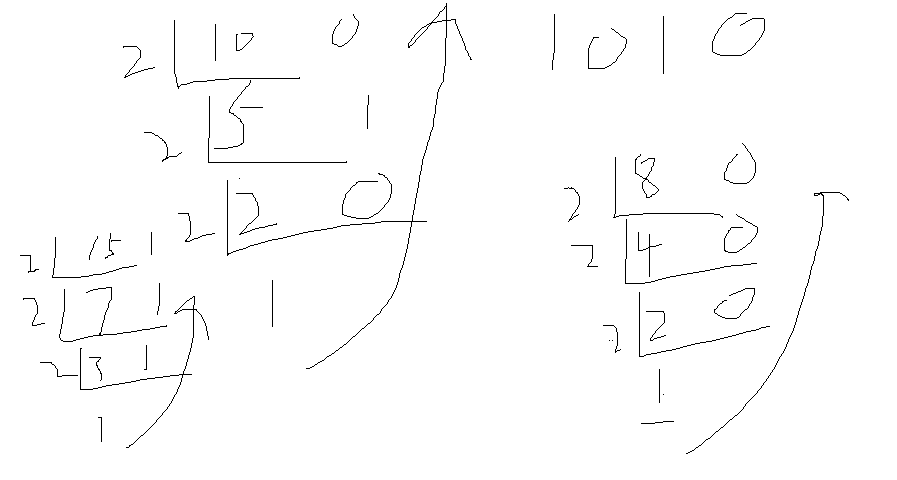
中国式婚姻

纯爷们(true) + 纯爷们(true)->不能结婚(false);

妹纸(false) + 妹纸(false) ->不能结婚(false);

纯爷们(true) + 妹纸(false)->可以结婚(true);

妹纸(false) + 纯爷们(true)->可以结婚(true);

1. 位运算符（了解）
   1. 

按位与 &

在二进制的基础上对每一位进行逻辑与运算

按位或 |

在二进制的基础上对每一个进行逻辑或运算

按位非 ~ 不讲 涉及反码补码

按位异或 ^（英文状态下键盘：shift+6）m

在二进制的基础上对每一个进行逻辑异或运算

左移<<

在二进制的基础上将所有二进制的数值向左移动，移动之后空白位置补0，效果类似与乘以2

右移>>

在二进制的基础上将所有二进制数向右移动，移动之后移出的位数会被删除，效果类似与除以2

1. 其他运算符
   1. @ 错误抑制符

作用：用于屏蔽单行错误，对于多行无效，尽量少用@符进行错误屏蔽。效率低。

1. **?: 三元运算符**

格式：条件表达式?真值:假值；

作用：根据条件的真假决定取真值还是假值

如果条件结果为true,那么将使用真值

如果条件结果为false,那么将使用假值

-> 对象成员访问符号

=> 数据元素访问符号

, (逗号) 声明多变量的分隔符(一般在循环时使用)

`` 反引号(键盘1前面的哪个键 英文状态下打) 指令执行符（执行系统命令的）

关于运算符的优先级

解决运算及最好的办法：推荐使用()



**流程控制**

1. 什么是流程控制
   1. 对程序代码执行流程的控制就是流程控制。
2. 什么是流程、
   1. 程序执行顺序就是流程。
3. 流程的分类
   1. 顺序结构(默认)
   2. 分支结构(选择结构)
   3. 循环结构
4. 顺序结构
   1. 程序自上而下的执行过程就是顺序结构，所有的程序默认都是顺序结构。
5. 分支结构
   1. 程序在执行过程中，往往要根据不同的运行结果做出不同的选择或者忽略部分代码的执行。
6. 分支结构的分类(4类)
   1. 单项分支结构
   2. 双项分支结构
   3. 多项分支结构
   4. 巢状分支结构
7. 单项分支结构(2种格式)
   1. 格式1：
      1. If（条件表达式）
         * 1. 一条PHP代码

如果条件表达式为真，将会执行if之后的第一条PHP代码

如果条件表达式为假，那么将不会执行if之后的第一条PHP代码

注意：该格式的分支结构只能控制if之后的第一条PHP语句

2.格式2

If(条件表达式){

一条PHP代码

二条PHP代码

……

}

该格式的分支结构，可以控制if之后的一个完整的代码块，{}区域的所有内容。

1. 双项分支结构
   1. 格式1：

If(条件表达式)

一条php语句

Else

一条pHP语句

双项分支是在单项分支的基础上增加了一个else分支

如果条件表达式的结果为真，将执行if后面的第一条PHP语句

如果条件表达式的结果为假，将执行else后面的第一条PHP语句

If后面的代码称之为if区间 或者 叫 真区间

Else后面的代码称之为esle区间 或者 假区间

2格式2

If(条件表达式){

一条PHP语句

二条PHP语句

……

}else{

一条PHP语句

二条PHP语句

……

}

推荐使用有{}格式版本的，带{}代码的可读性高，不容易混淆

格式2的功能已经包含了格式1的功能，{}内可以写一条PHP语句

1. 多项分支结构
   1. 格式1：
      1. If...else分支扩展

If(条件表达式){

PHP代码

}else if(条件表达式){

PHP代码

}elseif(条件表达式){

PHP代码

}else{

PHP代码

}

注意：elseif关键词可以写成else if也可以写成elseif，都市合格的语法，每一个if()内都有一个条件表达式。

格式：

Switch(变量){

Case 值:

PHP语句

[Break;]可以写可以不写

Case 值；

PHP语句

[break;]

Defaule:

PHP语句

}

注意：

1. switch...case的分支选择和if..elseif..else的分支选择并不相同。
2. Switch...case的分支结构可以同时执行多条分支(分支中没有break语句，那么将从复合当前case开始一直执行到最后)
3. 如果使用switch...case的实现多项分支的单项选择，可以为每一个分支添加break语句即可
4. Switch...case的取值从字符串和整型以及布尔值中选取，不推荐使用浮点型，不可以使用复合数据类型和特殊数据类型
5. Switch...case 只用于精确值判断，不用于范围判断。

巢状分支结构明天再讲

作业：

今日课堂代码3遍。

拿到班级同学学出来的效果代码手写三遍。

扩展作业：如果可以使用单项分支、多项分支的两种结构分别写出计算器案例和背景颜色变换案例。代码1遍。

计算器和背景颜色变换 可以不考虑使用 双项分支结构

扩展作业1:计算器



扩展作业2：背景颜色变换

