复习：

自学一门新的编程语言：

(1)了解背景

(2)搭建环境

(3)变量和常量

$age = 20; echo $age;

const PI = 3.14; echo PI;

$uname = 'tom';

echo "用户名：$uname";

(4)数据类型

原生值类型： int、float、string、boolean

复合类型：array、object

其它类型：null、resource

(5)运算符

算术：+ - \* / % ++ --

比较：> >= < <= == != === !==

逻辑：&& || !

(6)逻辑结构

(7)通用小程序

(8)函数和对象

(9)第三方库、框架

(10)实用项目

|  |
| --- |
| 有基础学员的自学内容：  PHP官方手册——“语言参考”相关章节  PHP扩展内容：面向对象&第三方库  https://pan.baidu.com/share/link?shareid=3623474815&uk=171029850 |

今日目标：

(1)补充：运算符

(2)逻辑结构：选择结构 —— 重点

(3)逻辑结构：循环结构 —— 重点

1.PHP中的运算符

(1)算术运算符

(2)比较运算符

(3)逻辑运算符

(4)位运算符

<<(按位左移) >>(按位右移动)

(5)赋值运算符

= += -= \*= /= %=

(6)拼接运算符

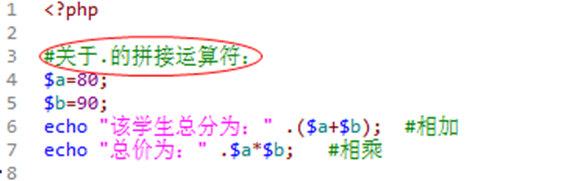
. .=

$str= 'hello';

$str = $str . 'world';

$str .= 'world';

注意：PHP中+只用于算术运算，不能用于拼接！



(7)三目运算符

? :

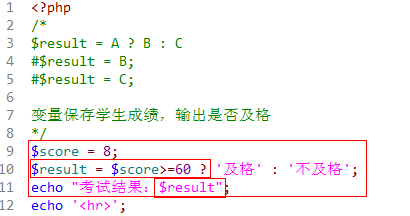
语法：式1 ? 式2 : 式3

含义：如果式1为true，则返回式2结果，否则返回式3的结果（含义等同于if...else...）

|  |
| --- |
| 面试题：下面的四种方法，哪一种可以最快速的实现一个数字\*8的效果  a) $i = $i\*8 #可以实现，速度不是最快  b) $i \* 8; #不能实现  c) $i = $i<<8; #\*2^8即\*256  d) $i = $i<<3; #\*2^3即\*8  $i = 5; #int，占4个字节，计算机存储为：  00000000 00000000 00000000 00000101  把此数据按位左移1位，变为： 10 = 5\*2^1  0000000 00000000 00000000 000001010  把此数据按位左移2位，变为： 20 = 5\*2^2  000000 00000000 00000000 0000010100  把此数据按位左移3位，变为： 40 = 5\*2^3  00000 00000000 00000000 00000101000  把此数据按位左移4位，变为： 80 = 5\*2^4  0000 00000000 00000000 000001010000  ...  把此数据按位左移4位，变为： 5\*2^N |
| $age = 20 ;  $age = $age + 5 ;  $age += 5 ; #作用与上一句相同，速度更快 |
| 单目运算符： $a++  双目运算符： $i + $j  三目运算符： $a ? $b : $c |

练习：创建变量保存学生的语文和数学成绩，用一个echo输出该学生的总分，形如：

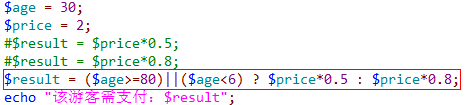
该学生总分为： 170



练习：创建变量保存商品单价和购买数量，用一个echo输出商品金额小计，形如：

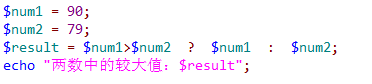
商品金额小计：4500

练习：变量保存学生成绩，输出是否及格

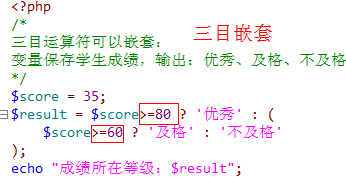
练习：创建一个变量表示游客的年龄，公园门票价格2元，如果是老人和儿童票价打五折，否则打八折，输出该游客最终需要付的门票金额。

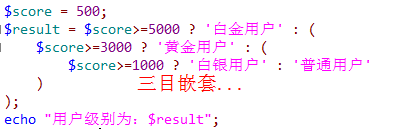
练习：创建一个变量表示论坛中用户的个人签名，若某用户签名为""，则自动赋值为“该用户很懒,什么都没留下”，输出该用户的签名。



练习：创建两个变量保存两个数字，使用三目运算符返回这两个数字中较大的那个值。

练习：变量保存学生成绩，输出：优秀、及格、不及格



练习：论坛中用户有积分，5000分评定为白金用户，5000~3000分评定为黄金用户，3000~1000评定为白银用户，否则评定为普通用户。创建变量表示用的积分，输出该用户的级别。

2.程序的逻辑结构

程序 = 数据 + 操作/算法

程序操作/算法由三种基本结构组成：

(1)顺序执行

(2)选择执行

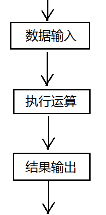
(3)循环执行

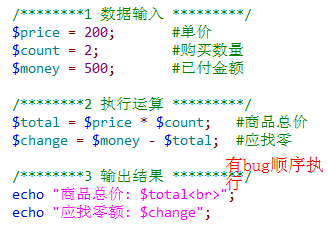
3.程序逻辑结构——顺序执行

顺序执行：1=>2=>3

示例：编写一个收银柜台收款程序，根据商品单价、购买数量以及收款金额计算并输出应收金额和找零。

程序执行步骤：顺序执行



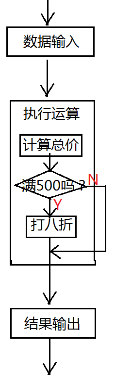


4.程序逻辑结构——选择执行（if...else...）

选择执行：有些代码会执行，有些则不会执行

示例：收银台程序V2，实现“满500打八折”功能

“如果.....就......”

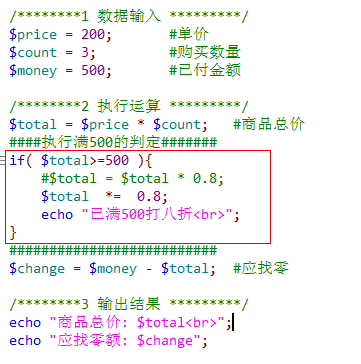


PHP中的选择执行语法：

**if( 条件判定 ){**

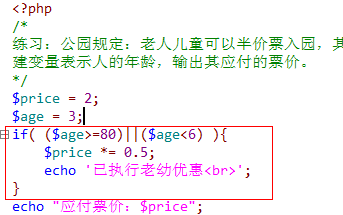
**#条件判定为true时执行的代码**

**}**

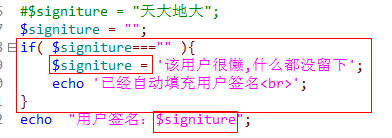


|  |
| --- |
| 语句1;  if( $total >= 500 ){  语句2;  }  语句3;  如果条件为true： 语句1=>语句2=>语句3  如果条件为false： 语句1=>语句3 |

练习：公园规定：老人儿童可以半价票入园，其他人全票入园，票价原价为2元。创建变量表示人的年龄，输出其应付的票价。



练习：创建一个变量表示论坛中用户的个人签名，若某用户签名为""，则自动赋值为“该用户很懒,什么都没留下”，输出该用户的签名。



示例：收银台程序V3，实现“已付金额足够则输出找零；否则输出错误提示”功能

“如果......就.....； 否则就......”

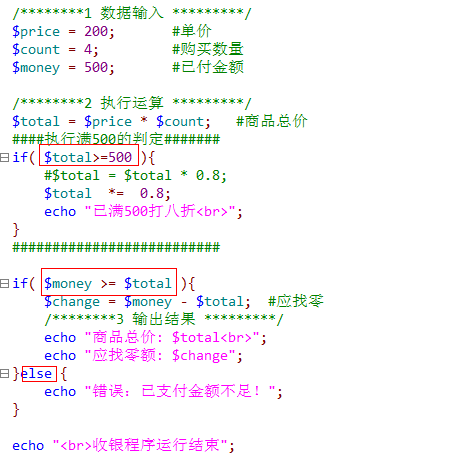
if( 条件判定 ){

#条件为true时执行的代码

}else {

#条件为false时执行的代码

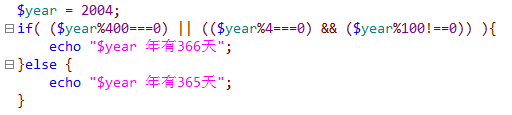
}



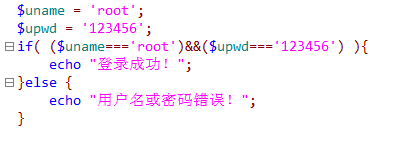
|  |
| --- |
| 语句1;  if( $total >= 500 ){  语句2;  }else {  语句3;  }  语句4;  如果条件为true： 语句1=>语句2=>语句4  如果条件为false： 语句1=>语句3=>语句4 |

练习：创建变量表示考试成绩，如果满60则输出及格，否则输出不及格。

练习：创建一个变量表示一个年份，如果是闰年，则输出“该年份有366天”，否则输出“该年份有365天”。



练习：创建变量表示用户输入的用户名和密码，如果用户名为root且密码为123456，输出“登录成功”，否则输出“用户名或密码错误”



练习：购物完整流程：



练习:创建变量保存学生成绩，输出：优秀、及格、不及格

练习：学子商城中，订单状态可能取值和对应编码：

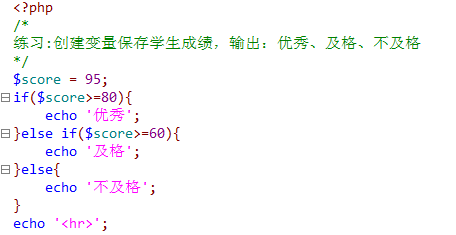
等待付款——10 备货中——20 运输中——30

派送中—— 40 已签收——50 未知状态—— 其它

在数据库中真实存储的是每个状态的编码，创建一个变量表示某个订单的状态编码，输出为对应的汉字

|  |
| --- |
| if选择结构可能的形式：  if( 条件 ){  } |
| if( 条件 ){  }else {  } |
| if( 条件 ){  }else {  if(){  }else{  }  } #情形较多时可能导致代码可读性差 |
| if( 条件1 ){  }else if( 条件2 ){  }else if( 条件3 ){  }else {  } #情形较多时推荐的写法 |

练习：创建变量保存学生成绩，输出：优秀、及格、不及格

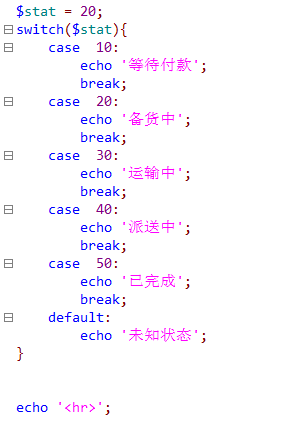


练习：学子商城中，订单状态可能取值和对应编码：

等待付款——10 备货中——20 运输中——30

派送中—— 40 已签收——50 未知状态—— 其它

在数据库中真实存储的是每个状态的编码，创建一个变量表示某个订单的状态编码，输出为对应的汉字

5.程序逻辑结构——选择执行（switch...case...） —— 了解

switch：开关、切换

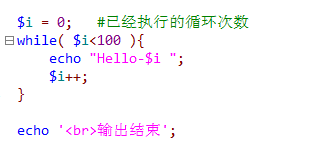
case：情形、场景

switch..case...：在多个不同的情形间切换，语法：

|  |  |
| --- | --- |
| $stat = 30;  switch( $stat ){  case 10: #stat==10  语句1;  **break; #打断，不再执行**  case 20: #stat==20  语句2;  **break;**  ....  default:  语句n;  } | $stat = 30;  if( $stat==10){  语句1;  }else if($stat==20){  语句2;  }else if(...){  .....  }else{  语句n;  } |
| switch..case...可以被if...else...代替，且只能进行相等判定，功能没有if...else...强大 | |

6.程序逻辑结构——循环结构 —— 重点&难点

循环结构：反复多次执行一样或类似的操作。如：打印100遍Hello、从Hello1打印到Hello100。



循环执行的二要素：

**(1)循环体：** 要反复执行的内容

**(2)循环条件：** 循环可以继续进行的条件

循环执行的第一种语法：

while( 判定条件 ){

#循环主体：要执行的语句

}

含义：当循环条件仍然满足时，就再次执行一次循环主体；直到循环条件为false，退出循环。

|  |  |
| --- | --- |
| $count = 0;  if( $count<10 ){  echo '需要再跑一圈';  }  echo '结束'; | $count = 0;  while( $count<10 ){  echo '需要再跑一圈';  }  echo '结束'; |
| 大括号最多执行一次 | 大括号可能执行很多遍 |

课后任务：—— 一起讨论完成！

(1)输出：1/2/3.../100

(2)输出：2/4/6.../98 —— 2星题

(3)输出：95/90/85/80..../5 —— 3星题

（8）循环计算1+2+3+...+100的整数和（5050），输出该整数和；

（9）循环计算1\*2\*3\*...\*10的整数乘积（362880），输出该整数积；

（10）循环计算90/5+80/10+70/15+60/20+50/25的和（），输出该和；

--------------------------------

(4)输出：※

(5)输出：十个※+<br>

※※※※※※※※※※<br>

(6)输出：五行十列个※ —— 4星题

※※※※※※※※※※<br>

※※※※※※※※※※<br>

※※※※※※※※※※<br>

※※※※※※※※※※<br>

※※※※※※※※※※<br>

(7)输出： —— 5星题

※

※※

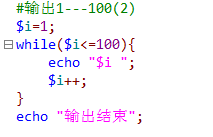
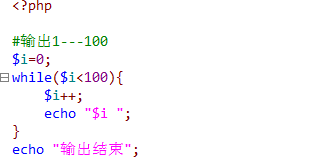
※※※

※※※※

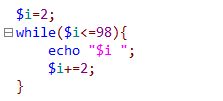
※※※※※

## （11）★99乘法表

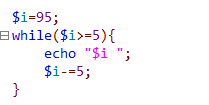
（1）



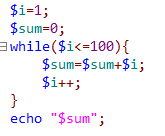
（2）



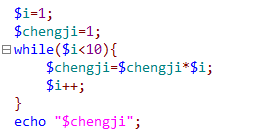
（3）



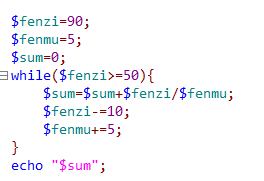
（8）



（9）

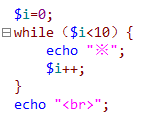


（10）

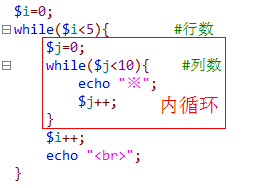


（4）

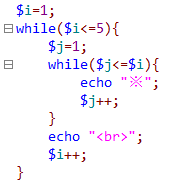
（5）



（6）



（7）



（11）

