复习：

掌握一门语言的基本步骤

(1)了解背景

(2)搭建环境

(3)变量常量

$变量名 = 值; echo $变量名;

const 常量名 = 值; echo 常量名;

(4)数据类型

标量/值/基础类型

int、float、string、bool

复合类型

array、object

其它类型

null、resource

(5)运算符

算术： + - \* / % ++ --

比较： > < >= <= == != === !==

逻辑： && || !

位： << >>

赋值： = += -= \*= /= %=

拼接： . .=

三目： 式1 ? 式2 : 式3

(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

练习：数据库中保存订单状态不使用字符串(费空间/不便于实现i18n)，而使用一个数字存储：

1-等待付款 2-备货中 3-运输中

4-配送中 5-已完成 其它-未知状态

创建变量表示一个订单的状态值，输出为对应的汉字

今日目标：

(1)逻辑结构 —— 注意逻辑思考 & 练习多

程序 = 数据 + 算法

1.算法的基本结构

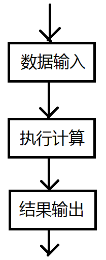
(1)顺序执行

(2)选择执行

(3)循环执行

2.程序逻辑结构 —— 顺序执行

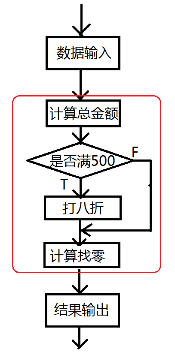
收银台程序 V1.0



练习：(1)创建一个变量表示商品单价、一个变量表示购买数量、一个变量表示用户已支付的金额 (2)计算总金额以及找零 (3)输出总金额和找零数值

3.程序逻辑结构—— 选择/分支执行之一

收银台程序 V2.0： 若购物总金额满500，则打八折



选择执行的最基础代码：

if( 条件判定 ){

//条件判定结果为true时执行的代码

}

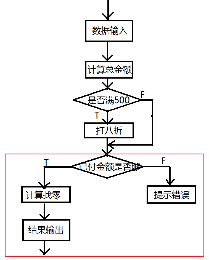
|  |
| --- |
| 语句1;  if( 条件判定 ){  语句2;  }  语句3; |
| 如果条件判定为true，执行顺序： 1=>2=>3  如果条件判定为false，执行顺序： 1=>3 |

练习：将收银台程序升级为2.0，添加“满500打八折”选择逻辑

练习：创建一个变量保存用户的个人签名，如果签名为""，就赋值为“主人很懒，什么都没留下”，输出最后的个人签名。

练习：创建变量表示员工的工资，再创建变量表示员工的政治面貌，若政治面貌为“党员”，则最后工资再+200元，输出员工最后的工资。

收银台程序 V3.0： 若已付金额足够商品总价格，计算找零并输出；否则输出错误提示



"如果...否则..."逻辑的语法结构：

if( 条件判定 ){

//条件判定为true时执行的语句

}else {

//条件判定为false时执行的语句

}

|  |
| --- |
| 语句1;  if( 条件判定 ){  语句2;  }else {  语句3;  }  语句4; |
| 条件判定为true，执行顺序： 1=>2=>4  条件判定为false，执行顺序： 1=>3=>4 |

练习：创建一个变量保存学生成绩，如果大于60分，输出“及格”，否则输出“不及格”

练习：创建两个变量保存两个学生的成绩，输出其中较大的值

练习：创建变量保存用户输入的登录名和密码，若登录为root密码为123456，输出“登录成功”，否则输出“用户名或密码错误”

练习：创建一个变量保存一个年份，若此年份是闰年输出“此年有366天”否则输出“此年有365天”

练习：创建一个变量保存学生成绩，如果小于60分，输出“不及格”，否则如果小于80输出“及格”，否则输出“优秀”

练习：创建变量保存用户的积分，若积分>1000则输出白金用户；否则如果积分大于500则输出黄金用户；否则如果大于200则输出白银用户，否则输出普通用户

练习：数据库中保存订单状态不使用字符串(费空间/不便于实现i18n)，而使用一个数字存储：

1-等待付款 2-备货中 3-运输中 14：49

4-配送中 5-已完成 其它-未知状态

创建变量表示一个订单的状态值，输出为对应的汉字

关于"if...else....的嵌套问题"有两种编写方法：

(1) if( ){

}else {

if(){

}else {

}

}

(2) if( ){

}else if( ){

}else if( ){

}else {

}

if\*1 + else if\*N + else\*1

|  |
| --- |
| 小结：  (1) if( ){ }  (2) if( ){ }else{ }  (3) if( ){ }else{ if(){}else{} }  (4) if( ){ }else if( ){ }... else{ } |

4.程序逻辑结构—— 选择/分支执行之二

**switch...case...break**选择分支结构：在多个不同“情形下”进行“切换”。

语法： switch( 变量名 ){

case 值1: #if(变量名==值1)

语句1;

case 值2: #if(变量名==值2)

语句2;

....

default: #else....

语句n;

}

含义：若指定变量等于值1，则执行语句1；否则若等于值2，则执行语句2.... 否则执行语句n。

注意：默认情况下，switch若判定了某个相等条件则执行该条件以及后续所有条件中的执行语句 —— 必须配合break关键字使用。

练习：创建一个变量表示用户在ATM机上的选项，值为：

1 - 输出“余额查询中...”

2 - 输出“存款进行中...”

3 - 输出“转账进行中...”

4 - 输出“取款进行中...”

其它 - 输出“不存在的选项！”

5.程序逻辑结构 —— 循环结构之一

循环：多次执行相同或相似的代码

循环二要素： (1)循环体 (2)循环条件

第一种循环结构：

while( 循环条件判定 ){

//循环体

}

含义：执行条件判定，若为true，则执行一次循环体；再次执行条件判定.....直到某次执行完循环体，再去执行条件判定为false，则退出循环，继续执行后面的语句

练习：打印出10/11/12/13...90/

练习：打印出5/10/15...90/

练习：打印出99/97/95...1/

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

练习：输出1+2+3+...+100的总和

$sum = 36;

1 2 3 4 5 6 7 8.... 100

练习：输出1\*2\*3\*...\*10的乘积

$result = 1;

1 2 3 4 5 6 8 ... 10

练习：输出 1/90 + 3/85 + 5/80 + 7/75 + 9/70的总和

$sum = 0;

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

练习：输出一个※

练习：输出10个※（在一行中），最后再输出一个换行

练习：输出5行10列的※，如下：

※※※※※※※※※※

※※※※※※※※※※

※※※※※※※※※※

※※※※※※※※※※

※※※※※※※※※※

课后作业：

(1)删除所有的示例代码，根据注释重写一遍！

(2)打星星

※

※※

※※※

※※※※

※※※※※

(3)打印九九乘法表

1\*1=1

2\*1=2 2\*2=4

3\*1=3 3\*2=6 3\*3=9

....

(4)打星星

※※※※※

※※※※

※※※

※※

※