复习：

掌握一门新语言基本步骤：

(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)函数对象

(9)第三方组件、库、框架

(10)实用项目

今日目标：

(1)逻辑结构：选择结构 —— 重要&简单

(2)逻辑结构：循环结构 —— 重要&稍难

难点：使用循环和选择练习“通用小程序”

1.程序的逻辑结构

程序由三种结构组成：

(1)顺序执行

(2)选择执行

(3)循环执行

2.程序逻辑结构 —— 选择执行： if

语法：

if( 条件判定 ){

条件判定为true要执行的代码;

}

|  |
| --- |
| 语句1;  if( 条件 ){  语句2;  }  语句3; |
| 条件判定为true： 语句1 => 语句2 => 语句3  条件判定为false： 语句1 => 语句3 |

练习：创建一个变量保存人的年龄，如果是老年人，就输出“欢迎您！老年卡！”，如果不是老年人，只输出“欢迎您！”

练习：创建一个变量保存用户的签名，如果用户签名为空，赋值为“主人很懒，什么都没留下”，最后输出用户的签名

3.程序逻辑结构 —— 选择执行： if...else...

语法：

if( 条件判定 ){

条件判定为true时执行的代码;

}else{

条件判定为false时执行的代码;

}

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

练习：创建一个变量保存学生的成绩，如果>=60,输出“及格”；否则输出“不及格”

练习：创建一个变量保存学生的成绩，如果>=80,输出“优秀”；否则 {如果>=60，输出“及格”，否则输出“不及格”}

午间练习：

1.判定论坛中用户积分所在的等级

>=500 元老

>=200 精英

>0 战士

2.创建一个整数变量表示一个订单的状态，输出对应的汉字提示文字

|  |  |
| --- | --- |
| if( 条件1 ){  语句1;  }else {  if( 条件2 ){  语句2;  }else{  语句3;  }  } | if( 条件1 ){  语句1;  }else if( 条件2 ){  语句2;  }else {  语句3;  } |
| 结论：else中嵌套if..else..可以简写为 if...else if...else if.... else.... | |

练习：数据库中订单的支付方式存储为数字，1-银行卡支付 2-支付宝支付 3-微信支付 4-京东白条 其它-未知方式。创建一个整数变量表示支付方式，输出为对应的汉字。

4.程序逻辑结构 —— 选择执行： switch...case....

switch：开关、切换

case：情形

switch...case....：在多种情形下进行转换，如果是情形1，执行xxx；否则如果是情形2，执行yyy；否则如何使情形3，执行zzz...再否则执行mmm

|  |  |
| --- | --- |
| if( $age === 10){  语句1;  }else if( $age===20 ){  语句2;  }else if( $age===30 ){  语句3;  }else {  语句4;  } | switch( $age ){  case 10: #$age===10  语句1;  break;  case 20: #$age===20  语句2;  break;  case 30:  语句3;  break;  default:  语句4;  } |
| switch..case...相当于一种特殊的if...else if...else... | |
| **switch..case...使用陷阱：**若某个case匹配成功，则此case以及后续case代码都会被执行！！！可以使用**break**关键字打断/终止switch的执行 | |

5.程序逻辑结构 —— 循环 : while

循环：反复多次执行一样或类似的任务

循环执行的二要素： **(1)循环体**  **(2)循环条件**

表达的含义：当循环条件仍然满足的情况下，再执行一次循环体.....直到循环条件不再满足了，就退出执行。

while: 当....的时候就做.....

语法：

while( 循环条件 ){

循环体;

}

当循环条件仍然为true，就再执行一次循环体...直到循环条件为false，退出循环

练习：使用循环打印出 0-hello、1-hello、2-hello...9-hello

练习：使用循环打印出 0、1、2、3....9

练习：使用循环打印出 0、2、4、6、8....98

练习：使用循环打印出 9、7、5、3、1

练习：使用循环打印出 1/99、 2/98、3/97、4/96.... 10/90

练习：输出1+2+3+....+100的累加和

练习：输出1\*2\*3\*...\*10的累乘积

练习：打印出一个※

练习：打印出10个※，使用循环

练习：打印出(两种方法实现) 17:10

※※※※※※※※※※

※※※※※※※※※※

※※※※※※※※※※

※※※※※※※※※※

※※※※※※※※※※

练习：打印出

※

※※

※※※

※※※※

※※※※※

练习：使用循环嵌套输出九九乘法表

1\*1=1

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

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

4\*1=4 4\*2=8 4\*3=12 4\*4=16

....

6.程序逻辑结构 —— 循环 : do....while.....

语法：

do {

循环体;

}while( 循环条件 );

示例：从0输出到9

|  |  |  |
| --- | --- | --- |
| $i = 0;  do { #先执行再判定  echo $i;  $i++;  }while( $i<10 ); | | $i = 0;  while( $i<10 ){ #先判定再执行  echo $i;  $i++;  } |
| while可以实现的功能do...while....都可以实现。 | | |
| $i = 20;  do {  echo $i;  $i++;  }while( $i<10 ); | $i = 20;  while( $i<10 ){ #先判定再执行  echo $i;  $i++;  } | |
| do...while...循环体至少执行一次 | while的循环体可能一次都无法执行 | |

课后练习：

(1)删除所有的示例代码，根据注释重写！！

(2)使用do..while输出10个※

(3)使用do..while输出5行10列个※

(4)使用do..while输出

※

※※

※※※

※※※※

※※※※※

(5)使用while和do...while分别实现九九乘法表

(6)使用while和do...while分别输出如下的三角

※※※※※

※※※※

※※※

※※

※