1. PHP操作MySQL数据库的练习题：（1）写SQL语句创建数据库ustb；创建一个表userinfo，表中添加字段id、name、class，属性不做统一要求，可自行设计。

创建ustb：CREATE DATABASE ustb;

创建userinfo: CREATE TABLE userinfo (

id INT,

name VARCHAR(50),

class VARCHAR(50)

);

（2）写insert语句，利用PHP执行语句，向表userinfo中添加两条记录，记录具体信息不统一要求，自行添加即可。根据语句执行情况，输出提示信息，如“添加成功……”之类，自行设计输出效果。（3）从表中查询某记录的数据，写select语句，利用PHP执行，并根据情况处理结果集，不统一要求自行设计输出。

<?php

$servername = "localhost";

$username = "your\_username";

$password = "your\_password";

$dbname = "ustb";

$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$sql = "INSERT INTO userinfo (id, name, class)

VALUES (1, 'John Doe', 'Class A'),

(2, 'Jane Smith', 'Class B')";

if ($conn->query($sql) === TRUE) {

echo "插入记录成功";

} else {

echo "Error: " . $sql . "<br>" . $conn->error;

}

$conn->close();

?>

1. 流程控制语句练习题：使用for循环输出100以内的偶数。

<?php

// 使用for循环输出100以内的偶数

for ($i = 0; $i <= 100; $i += 2) {

echo $i . " ";

}

?>

1. web数据交互练习题：制作一个简易的登录表单页面，要求输入用户名、密码，有登录、重置按钮。接收用户输入的数据，输出到页面。

<!DOCTYPE html>

<html>

<head>

<title>登录</title>

</head>

<body>

<h2>登录</h2>

<form action="" method="post">

<label for="username">用户名:</label>

<input type="text" id="username" name="username"><br><br>

<label for="password">密码:</label>

<input type="password" id="password" name="password"><br><br>

<input type="submit" value="登录" name="login">

<input type="reset" value="重置">

</form>

<?php

if (isset($\_POST["login"])) {

$username = $\_POST["username"];

$password = $\_POST["password"];

echo "<h3>登录结果:</h3>";

echo "用户名: " . $username . "<br>";

echo "密码: " . $password;

}

?>

</body>

</html>

4.     复习教材的第6章内容，完成PHP实现文件上传等相关操作的练习题：写一个form表单，添加一个上传文件的表单控件，加一个提交按钮。写一个脚本，接收上述表单控件上传的文件，并保存到指定目录下。注意：（1）表单中含上传文件控件，则form表单的属性，要定义enctype属性，如 :

<form action="" method="post" enctype="multipart/form-data">

</form>

（2）保存文件之前一般需要进行一些判断，比如文件类型、大小等是否符合要求，具体要求可自行设计，最后再使用

move\_uploaded\_file()函数，将上传的文件保存到指定目录下。

<?php

if (isset($\_POST["submit"])) {

$targetDir = "uploads/";

$targetFile = $targetDir . basename($\_FILES["file"]["name"]);

$uploadOk = 1;

$fileType = strtolower(pathinfo($targetFile, PATHINFO\_EXTENSION));

// 检查文件类型

if ($fileType != "txt" && $fileType != "pdf" && $fileType != "doc") {

echo "只允许上传txt, pdf, doc文件.";

$uploadOk = 0;

}

// 检查文件大小

if ($\_FILES["file"]["size"] > 500000) {

echo "文件大小超过限制.";

$uploadOk = 0;

}

// 检查上传状态

if ($uploadOk == 0) {

echo "文件上传失败.";

} else {

if (move\_uploaded\_file($\_FILES["file"]["tmp\_name"], $targetFile)) {

echo "文件已上传: " . basename($\_FILES["file"]["name"]);

} else {

echo "文件上传失败.";

}

}

}

?>

5. PHP中三元运算符的使用练习题：

$number = 100;

$msg = $number > 0 ? '大于0' : '不大于0';

echo $msg;

以上代码的输出结果是什么，三元运算符的执行过程要理解，很多编程中用处比较多，它可以让代码更简洁。

大于0，条件是 $number > 0，由于 $number 的值为 100，大于0，所以条件为真。因此，返回表达式1的值 "大于0"，并将其赋给变量 $msg。

最后，通过 echo 输出变量 $msg 的值，即 "大于0"。

6. PHP中的包含语句在平时作业中练习过几次了，用过最多的是require，除此之外，有相似功能的函数还有include、require\_once、include\_once。这些语句的共同作用是：“请求包含目的资源”，他们的区别要清楚，他们的语法格式要了解。

以下是包含语句的不同类型及其区别：

1. `include`：用于将指定文件包含到当前脚本中。如果包含文件不存在或出现错误，那么脚本会继续执行，并显示一个警告。

语法：`include 'filename';`

2. `require`：与 `include` 作用相同，也用于将指定文件包含到当前脚本中。唯一的区别是，如果包含文件不存在或出现错误，那么脚本会停止执行，并显示一个致命错误。

语法：`require 'filename';`

3. `include\_once`：用于包含指定文件，只有在之前没有被包含过的情况下才会执行包含操作。即便在同一个脚本中多次调用 `include\_once`，文件也只会被包含一次。

语法：`include\_once 'filename';`

4. `require\_once`：与 `include\_once` 的作用相同，用于包含指定文件，只有在之前没有被包含过的情况下才会执行包含操作。

语法：`require\_once 'filename';`

需要注意的是，包含语句中的文件路径是相对于当前脚本的。如果文件不在同一目录下，可以使用相对路径或绝对路径来指定文件的位置。

总体来说，`include` 和 `require` 是最常用的包含函数，其中 `require` 更严格，如果文件不存在或出错会导致脚本停止执行。而 `include` 只会发出警告，脚本会继续执行。

另外，使用 `\_once` 可以确保文件只被包含一次，避免重复包含导致的错误。这在引入类、函数或配置文件时很有用。

7. PHP的赋值方式有哪两种？一种传值赋值，如 $a=1; $b=$a;   另一种是引用赋值，如$a=1; $b=&$a;

理解和区分两种方式的数据传递原理。

在PHP中，赋值操作有两种方式：传值赋值和引用赋值。

1. 传值赋值：通过将一个变量的值复制给另一个变量来实现赋值。例如，`$a = 1; $b = $a;` 将 `1` 的值赋给变量 `$a`，然后将 `$a` 的值 `1` 再次复制给变量 `$b`。

在传值赋值中，每个变量都有自己的内存空间，它们之间是独立的，任何一个变量的修改都不会影响其他变量。传值赋值：在传值赋值中，每个变量都在内存中有自己的存储空间，存储着其对应的值。当一个变量的值被赋给另一个变量时，实际上是将值从一个存储空间复制到另一个存储空间。这样，两个变量之间就是独立的。

2. 引用赋值：通过将一个变量的引用赋给另一个变量来实现赋值。例如，`$a = 1; $b = &$a;` 将变量 `$a` 的引用赋给变量 `$b`。引用赋值：在引用赋值中，两个变量实际上指向同一个存储空间，它们引用的是同一个值。当一个变量的引用被赋给另一个变量时，它们实际上指向了同一个内存地址。这样，无论通过哪个变量进行修改，都会反映在另一个变量上。

在引用赋值中，两个变量实际上指向同一个内存空间，它们是相互关联的。因此，任何一个变量的修改都会影响到另一个变量。

8.  PHP内置函数很多，一些上课时讲过的常用函数要掌握他们的作用，编程时会灵活使用：时间日期函数time()  date()等、查看数据类型的函数gettype()、字符串函数explode()、implode()、判断变量是否已经被定义或是否存在的函数isset()、判断文件是否是文件上传过程中产生的临时文件的函数is\_uploaded\_file()、PHP操作数据库的mysqli扩展中的一系列函数等。

1. 时间日期函数：

- `time()`：返回当前的Unix时间戳（自1970年1月1日以来经过的秒数）。

- `date($format, $timestamp)`：将Unix时间戳格式化为指定的日期时间格式。

2. 数据类型相关函数：

- `gettype($var)`：返回变量的数据类型。

- `is\_int($var)`, `is\_string($var)`, `is\_float($var)`, `is\_bool($var)`：检查变量是否是整数、字符串、浮点数、布尔类型。

- `is\_array($var)`, `is\_object($var)`, `is\_null($var)`：检查变量是否是数组、对象、空值。

3. 字符串函数：

- `explode($delimiter, $string)`：将字符串拆分为数组，根据指定的分隔符。

- `implode($glue, $array)`：将数组元素连接为一个字符串，使用指定的粘合剂（分隔符）。

- `strlen($string)`, `strpos($haystack, $needle)`, `substr($string, $start, $length)`：获取字符串长度、查找子字符串的位置、截取子字符串。

4. 变量相关函数：

- `isset($var)`：检测变量是否已经被定义并且具有非 NULL 值。

- `empty($var)`：检测变量是否为空，包括零、空字符串、空数组、空对象、NULL和未定义的变量。

- `unset($var)`：销毁指定的变量。

5. 文件函数：

- `is\_uploaded\_file($filename)`：检查文件是否是通过HTTP POST上传的临时文件。

- `file\_exists($filename)`：检查文件或目录是否存在。

6. 数据库相关函数（使用mysqli扩展）：

- 连接数据库：`mysqli\_connect()`

- 执行查询：`mysqli\_query()`

- 获取结果集：`mysqli\_fetch\_assoc()`, `mysqli\_fetch\_array()`, `mysqli\_fetch\_row()`

- 插入新记录：`mysqli\_insert\_id()`