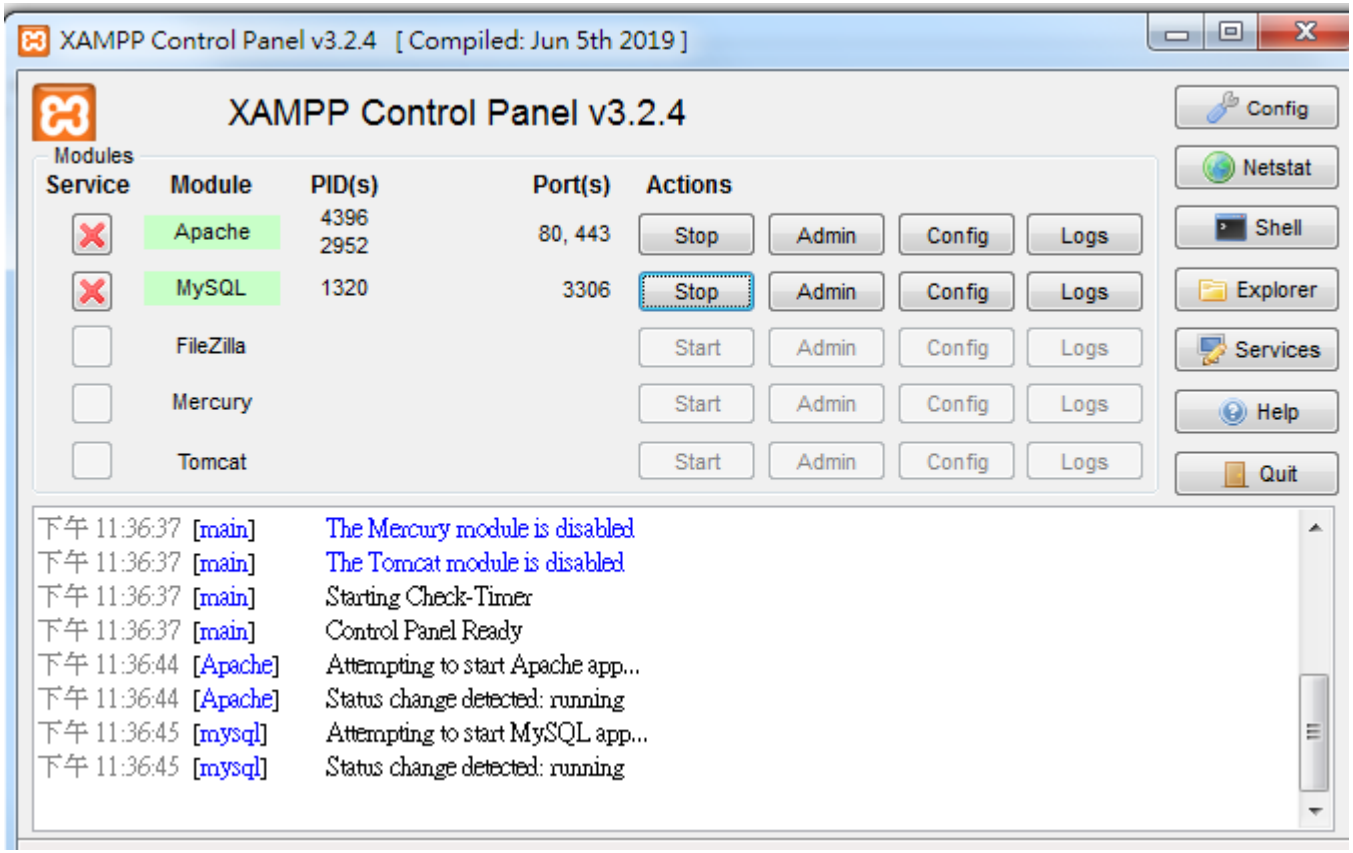
PHP 基本程式設計 I

Running Environment

Step 1


XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]
XAMPP Control Panel v3.2.4
Modules
Service
Module
PID(s)
Port(s)
Actions
Apache
4396
2952
80, 443
Stop
Admin
Config
Logs
MySQL
1320
3306
Stop
Admin
Config
Logs
FileZilla
Start
Admin
Config
Logs
Mercury
Start
Admin
Config
Logs
Tomcat
Start
Admin
Config
Logs
Config
Netstat
Shell
Explorer
Services
Help
Quit
下午 11:36:37 [main] The Mercury module is disabled
下午 11:36:37 [main] The Tomcat module is disabled
下午 11:36:37 [main] Starting Check-Timer
下午 11:36:37 [main] Control Panel Ready
下午 11:36:44 [Apache] Attempting to start Apache app...
下午 11:36:44 [Apache] Status change detected: running
下午 11:36:45 [mysql] Attempting to start MySQL app...
下午 11:36:45 [mysql] Status change detected: running

Step 2

- Save as ex52-1.php

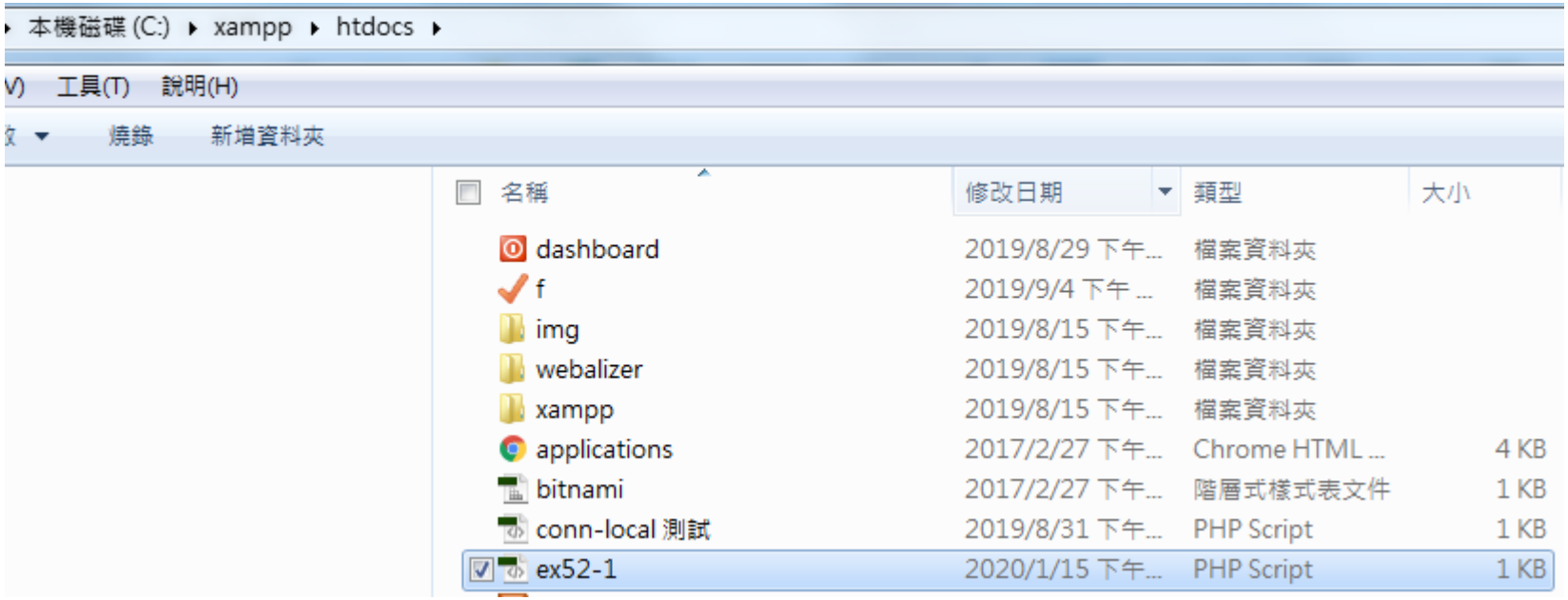
```
<!DOCTYPE html>
<html>
<body>

<?php
echo "My first PHP script!";
?>

</body>
</html>
```

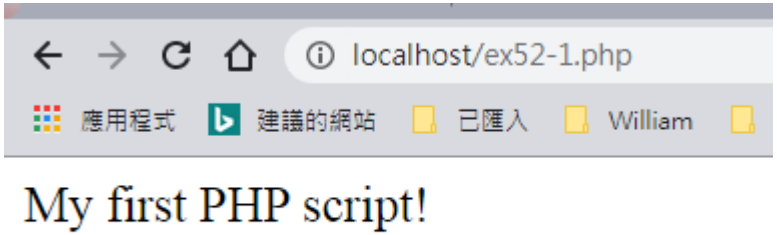
Step 3

- Copy ex52-1.php to the following folder



Step 4

- Run ex52-1.php under localhost on your browser



About php

php?

- HP的全名為Hypertext Preprocessor，它是個被廣泛運用在網頁程式撰寫的語言，尤其是它能適用於網頁程式的開發及能夠嵌入HTML文件之中，它的語法和C、Java及Perl等語法相似，且學習起來更容易上手。PHP的目地是為了能
- PHP是在伺服器端執行的程式語言，所以任何其它的CGI程式所能做得到的，它都能做到。像是從表單中收集資料，或是產生動態的網頁內容，或者是傳送及接收cookies等等，PHP都能做得到!使網站開發者可以快速地撰寫動態網頁。
- PHP物件導向程式設計. PHP雖說是在HTML中的一種Script語言，但他也是一個物件導向語言。

php?

- php能使用在大多數的作業系統，像是Linux、HP-UX、Solaris、OpenBSD、Microsoft Windows、Mac OS X、 RISC OS等等都能使用。PHP也能在大多數的網站伺服器上執行，像是Apache、Microsoft Internet Information Server、Personal Web Server、Netscape and iPlanet servers、Oreilly Website Pro server、Caudium、Xitami、 OmniHTTPd。在大多數的伺服器中，PHP被編譯其中的一個模組，但PHP也能編譯成CGI模式，使PHP成為一個CGI處理程序。
- PHP的功用不單單只是輸出HTML文件而已，它的功能還包括了輸出圖形、PDF檔、及Flash檔。你當然也可以讓它輸出一些文字，像是
XHTML及任何其它的XML檔，PHP可以產生出以上這些檔案，並且將它們儲存在伺服器上。PHP也提供了相當多的協定，像是LDAP、IMAP、SNMP、NNTP、POP3、HTTP、COM等等，還有其它相當多的擴充模組可以使用。

優點

- **持續的更新-**PHP提供豐富的函數，而且往後還會不斷地有新的函數庫加入，以及不停地更新，這使得在程式設計方面有著更好的資源，同時還能在幾乎所有平台上良好地工作，這使得php 成為了開發者喜愛的熱門語言
- **快捷性-**程序開發快，運行快，技術本身學習快。因為PHP可以被嵌入於HTML語言，它相對於其他語言。**編輯簡單，實用性強，更適合初學者**
- **跨平台性強-**於PHP是運行在服務器端的腳本，可以運行在**UNIX、LINUX、WINDOWS、Mac OS、Android等平台。**
- **語法簡單-**如果先有學習C和Perl的人, 很容易上手，並且跟ASP有部分類似

優點

- **支援主流技術-**目前主流技術都支援，比如WebService、Ajax、XML等等，足夠應用。
- **成熟物件導向體系-** PHP已經有成熟的＊物件導向體系，能夠適應基本的物件導向要求，適合開發大型專案。
- **龐大的社群-**有成熟且龐大的社群來支援PHP的開發，如果在開發上遇到什麼問題，向php社群求援會是你解決問題的一個好方法。
- **應用在許多知名網站-**目前使用PHP語言進行網站建設的大型應用有很多，目前全球有2000多萬個網站使用PHP，**包括雅虎、Google、百度、YouTube、新浪、騰訊等知名網際網路公司均採用PHP語言來開發自身的系統，PHP 已成為了最熱門的開發語言之一。**

缺點

- 語法不太嚴謹，比如變數不需要定義就可以使用，像是在C，Java語言中的變數是必須先定義以後才可以使用的，所以在協作上(與他人共同管理上)，需要更多的資源投入與管理。
- 目錄結構混亂，相比其他框架目錄結構要差一點。
- PHP不適合密集型（大數據量）運算場景。PHP的語言特性決定PHP不適合做大型數據樣運算，PHP語言由C語言寫的，PHP處於C基礎之上，PHP的所有運算處理流程需要轉化為C語言，**並且PHP語言還有一些環境問題，語言特型，相比於C而言程式碼會冗長許多。**

PHP的前景是如何呢?

- 根據Techrepublic調查，2018年十大最被需求的程式語言中，PHP排名第9，而且Techrepublic認為PHP應用廣泛，並且擁有HTML不能完成的功能，還能跟MySQL數據庫互動。看來PHP還是有一定的影響力，使用簡單而且強大的功能讓PHP前景還是可見的~

What's new in PHP 7

- PHP 7 is much faster than the previous popular stable release (PHP 5.6)
- PHP 7 has improved Error Handling
- PHP 7 supports stricter Type Declarations for function arguments
- PHP 7 supports new operators (like the spaceship operator: <=>)

The `<=>` ("Spaceship") operator will offer combined comparison in that it will :

```
Return 0 if values on either side are equal
Return 1 if the value on the left is greater
Return -1 if the value on the right is greater
```

```
//Comparing Integers

 echo 1 <=> 1; //output  0
 echo 3 <=> 4; //output -1
 echo 4 <=> 3; //output  1

 //String Comparison

 echo "x" <=> "x"; //output  0
 echo "x" <=> "y"; //output -1
 echo "y" <=> "x"; //output  1
```

Set Up PHP on Your Own PC

- However, if your server does not support PHP, you must:
  - install a web server
  - install PHP
  - install a database, such as MySQL
- The official PHP website (PHP.net) has installation instructions for PHP: http://php.net/manual/en/install.php

PHP Syntax

Basic PHP Syntax

- A PHP script is executed on the server, and the plain HTML result is sent back to the browser.
- A PHP script starts with <?php and ends with ?>
- The default file extension for PHP files is ".php".

```
<?php
// PHP code goes here
?>
```

PHP structure

- A PHP file normally contains HTML tags, and some PHP scripting code.
- Below, we have an example of a simple PHP file, with a PHP script that uses a built-in PHP function "echo" to output the text "Hello World!" on a web page:

```
<!DOCTYPE html>
<html>
<body>

<h1>My first PHP page</h1>

<?php
echo "Hello World!";
?>

</body>
</html>
```

PHP Case in-sensitivity

- In PHP, keywords (e.g. if, else, while, echo, etc.), classes, functions, and user-defined functions are case-insensitive.
- In the example below, all three echo statements below are equal and legal:

```
<!DOCTYPE html>
<html>
<body>

<?php
ECHO "Hello World!<br>";
echo "Hello World!<br>";
EcHo "Hello World!<br>";
?>

</body>
</html>
```

Case-sensitive

- However; all variable names are case-sensitive!
- Look at the example below; only the first statement will display the value of the $color variable! This is because $color, $COLOR, and $coLOR are treated as three different variables:

```
<!DOCTYPE html>
<html>
<body>

<?php
$color = "red";
echo "My car is " . $color . "<br>";
echo "My house is " . $COLOR . "<br>";
echo "My boat is " . $coLOR . "<br>";
?>

</body>
</html>
```

Comments in PHP

- A comment in PHP code is a line that is not executed as a part of the program. Its only purpose is to be read by someone who is looking at the code.

```
<!DOCTYPE html>
<html>
<body>

<?php
// This is a single-line comment

# This is also a single-line comment
?>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<?php
/*
This is a multiple-lines comment block
that spans over multiple
lines
*/
?>

</body>
</html>
```

PHP Variables and Scopes

PHP Variables

- Variables are "containers" for storing information.
- In PHP, a variable starts with the $ sign, followed by the name of the variable:

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

- After the execution of the statements above, the variable $txt will hold the value ”Hello world!”, the variable $x will hold the value 5, and the variable $y will hold the value 10.5.
- When you assign a text value to a variable, put quotes around the value.
- Unlike other programming languages, PHP has no command for declaring a variable. It is created the moment you first assign a value to it.

PHP Variables

- A variable can have a short name (like x and y) or a more descriptive name (age, carname, total_volume).
- Rules for PHP variables:
  - A variable starts with the `$` sign, followed by the name of the variable
  - A variable name must start with a letter or the underscore character
  - A variable name cannot start with a number
  - A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
  - Variable names are case-sensitive (`$age` and `$AGE` are two different variables)
- Remember that PHP variable names are case-sensitive!

Output Variables

- The PHP echo statement is often used to output data to the screen.
- The following example will show how to output text and a variable:

```php
<?php
$txt = "W3Schools.com";
echo "I love $txt!";
?>
```

way 1

- The following example will produce the same output as the example above:

way 2

```php
<?php
$txt = "W3Schools.com";
echo "I love " . $txt . "!";
?>
```

Output Variables

- The following example will output the sum of two variables:

```php
<?php
$x = 5;
$y = 4;
echo $x + $y;
?>
```

PHP Variables Scope

- In PHP, variables can be assigned a value treated as a declaration anywhere in the script.
- The scope of a variable is the part of the script where the variable can be referenced/used.
- PHP has three different variable scopes:
  - local
  - global
  - static

Global and Local Scope

- A variable declared **outside** a function has a GLOBAL SCOPE and can only be accessed outside a function:

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 5; // global scope

function myTest() {
    // using x inside this function will generate an error
    echo "<p>Variable x inside function is: $x</p>";
}
myTest();

echo "<p>Variable x outside function IS: $x</p>";
?>

</body>
</html>
```

Variable x inside function is:

Variable x outside function IS: 5

Different from C; The global variable cannot be accessed by a local scope; Namely, cannot Use it as a local variable.

```
<!DOCTYPE html>
<html>
<body>

 <?php
  $x = 5; // global scope

  function myTest() {
    // using x inside this function will generate an error
    echo "<p> myTest()-Variable x inside function is: $x</p>";
  }

  myTest();
  echo "<p>Variable x outside function is: $x</p>";
 ?>

</body>
</html>
```

myTest()-Variable x inside function is:

Variable x outside function is: 5

Global variable setting

A variable declared **within** a function has a *LOCAL SCOPE* and can only be accessed within that function:

```
<!DOCTYPE html>
<html>
<body>

 <?php
  function myTest() {
   $x = 5;
   // using x inside this function will generate an error
   echo "<p> myTest()-Variable x inside function is: $x</p>";
  }

  myTest();
  echo "<p>Variable x outside function is: $x</p>";
 ?>

</body>
</html>
```

myTest()-Variable x inside function is: 5

Variable x outside function is:

Local variable setting

Global and Local Scope

- You can have local variables with the same name in different functions, because local variables are only recognized by the function in which they are assigned.

PHP The global Keyword

- The global keyword is used to access a global variable from within a function for global variable sharing.
- To do this, use the global keyword before the variables (inside the function):

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 5;
$y = 10;

function myTest() {
    global $x, $y;
    $y = $x + $y;
}

myTest();  // run function
echo $y; // output the new value for variable $y
?>

</body>
</html>
```

15

PHP The global Keyword: $GLOBALS[*index*]

- PHP also stores **all global variables** in an array called $GLOBALS[*index*]. The *index* holds the name of the variable. This array is also accessible from within functions and can be used to update global variables directly.

```
<!DOCTYPE html>
<html>
<body>

<?php
$x = 5;
$y = 10;

 function myTest() {
    $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
 }

 myTest();
 echo "variable \$y is ?". $y;
 echo "<p>variable \$GLOBALS['y'] is".  $GLOBALS['y'];

?>

</body>
</html>
```

variable $y is ?15

variable $GLOBALS['y'] is15

另一方法,存取global scope上的變數值

PHP The static Keyword

- Normally, when a function is completed/ executed, all of its variables are deleted. However, sometimes we want a local variable NOT to be deleted. We need it for a further job.
- To do this, use the static keyword when you first declare the variable:

```
<!DOCTYPE html>
<html>
<body>

 <?php
 function myTest() {
    static $x = 0;
    echo $x;
    $x++;
 }

 myTest();
 echo "<br>"; myTest();
 echo "<br>"; myTest();
 ?>

</body>
</html>
```

```
0
1
2
```

Then, each time the function is called, that variable will still have the information it contained from the last time the function was called.

Note: The variable is still local to the function.

PHP echo and print Statements

Display Text

- echo and print are more or less the same. They are both used to output data to the screen.

```
<!DOCTYPE html>
<html>
<body>

<?php
echo "<h2>PHP is Fun!</h2>";
echo "Hello world!<br>";
echo "I'm about to learn PHP!<br>";
echo "This " . "string ", "was ", "made ", "with multiple parameters.";
?>

</body>
</html>
```

PHP is Fun!

Hello world!
I'm about to learn PHP!
This string was made with multiple parameters.

Display Variables

- The following example shows how to output text and variables with the echo statement:

```
<!DOCTYPE html>
<html>
<body>

<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

echo "<h2>" . $txt1 . "</h2>";
echo "Study PHP at " . $txt2 . "<br>";
echo $x + $y;
?>

</body>
</html>
```

Learn PHP

Study PHP at W3Schools.com
9

Display Text

- The print statement can be used with or without parentheses: print or print().

```
<!DOCTYPE html>
<html>
<body>

<?php
print "<h2>PHP is Fun!</h2>";
print "Hello world!<br>";
print "I'm about to learn PHP!";
?>

</body>
</html>
```

PHP is Fun!

Hello world!
I'm about to learn PHP!

Display Variables

- The following example shows how to output text and variables with the print statement:

```
<!DOCTYPE html>
<html>
<body>

<?php
$txt1 = "Learn PHP";
$txt2 = "W3Schools.com";
$x = 5;
$y = 4;

print "<h2>" . $txt1 . "</h2>";
print "Study PHP at " . $txt2 . "<br>";
print $x + $y;
?>

</body>
</html>
```

Learn PHP

Study PHP at W3Schools.com
9

Connect to DB by using SELECT

FYI: https://www.w3schools.com/pHP/php_ref_overview.asp

查詢表格

localhost/ksu_select.html

應用程式 建議的網站 已匯入 William English Android 計畫 研究+競賽 技術移轉

ksu select operation

按查詢鍵, 查詢 ksu_std_table 中, 各系的學生人數.

查詢

ksu_std_table 學生於各系人數顯示如下:

系別	學生人數
CS	3
IE	3
IM	2

records found!

返回

```
SELECT ksu_std_department, count(1) FROM ksu_std_table group by ksu_std_department
```

效能分析 [行內編輯] [編輯] [SQL 語句分析] [建立 PH

全部顯示 | 資料列數： 25 篩選資料列: 搜尋此資料表

+ 選項

ksu_std_department	count(1)
CS	3
IE	3
IM	2

ksu_select3.html

```html
<!doctype html>
<html lang="zh_tw">
<head>
  <meta charset="utf-8">
  <title>Hello</title>
</head>
<body>
  <h3> ksu select operation </h3>
  <!--不對字符編碼 -->
  <form enctype="multipart/form-data" method="post"
        action="ksu_select3.php">
    按查詢鍵, 查詢 ksu_std_table 中, 各系的學生人數.
    <br/>
    <br/>
    <input type="submit" name="sub" value="查詢"/>
  </form>
</body>
</html>
```

ksu_std_table 資料表

```
SELECT ksu_std_department, count(1) FROM ksu_std_table group by ksu_std_department
```

效能分析 [行內編輯] [編輯] [SQL 語句分析] [建立 PH

全部顯示 | 資料列數： 25 篩選資料列: 搜尋此資料表

+ 選項

ksu_std_department	count(1)
CS	3
IE	3
IM	2

#	名稱	類型	編碼與排序
1	ksu_std_id	varchar(6)	latin1_swedish_ci
2	ksu_std_name	varchar(20)	utf8_unicode_ci
3	ksu_std_age	int(2)	
4	ksu_std_department	char(2)	latin1_swedish_ci
5	ksu_std_signin	date	
6	ksu_std_grade	int(1)	

```
echo "<table border='1'>
<tr>
  <th> 系別 </th>  <th>學生人數 </th>
</tr>";

//使用 mysqli_fetch_array() 取回資料庫資料
while($row = mysqli_fetch_array($result))
{
  echo "<tr>";
  echo "<td>" . $row['ksu_std_department'] . "</td>";
  echo "<td>" . $row['count(1)'] .   "</td>";
  echo "</tr>";
}
echo "</table>";
echo "records found!"."<br/><br/>";
```

ksu_select3.php

```php
<?php
 $db_host = "localhost";
 $db_name = "ksu_database";
 $db_table = "ksu_cstd_table";
 $db_user = "root";
 $db_password = "";

 // 連結檢測
 $conn = mysqli_connect($db_host, $db_user, $db_password);
 if(empty($conn)){
    print  mysqli_error ($conn);
    die ("無法對資料庫連線！" );
    exit;
 }
 if(!mysqli_select_db( $conn, $db_name)){
    die("資料庫不存在!");
    exit;
 }
```

ksu_select3.php

```
20  //自型設定
21  mysqli_set_charset($conn,'utf8');
22
23  echo "ksu_std_table  學生於各系人數顯示如下:". "<br/><br/>";
24  $result = mysqli_query($conn,
25                          "SELECT ksu_std_department, count(1) FROM ksu_std_table
                            group by ksu_std_department");
26  echo "<table border='1'>
27  <tr>
28    <th> 系別 </th>  <th>學生人數 </th>
29  </tr>";
30
31  //使用 mysqli_fetch_array() 取回資料庫資料
32  while($row = mysqli_fetch_array($result))
33  {
34    echo "<tr>";
35    echo "<td>" . $row['ksu_std_department'] . "</td>";
36    echo "<td>" . $row['count(1)'] .   "</td>";
37    echo "</tr>";
38  }
39  echo "</table>";
40  echo "records found!"."<br/><br/>";
41  ?>
42  <form enctype="multipart/form-data"  method="post" action="ksu_select3.html">
43  <input type="submit" name="sub" value="返回"/>
44  </form>
```

Exercise

- 寫出好程式: 1) 程式碼照公司規定撰寫: 註解+變數定義+含數定義+程式名稱定義 2) 程式中考慮效能

localhost/ksu_select4.php

應用程式 建議的網站 已匯入 William English Android

ksu_std_table 學生於各系人數顯示如下:

系別	學生人數
CS	3
IE	3
IM	2

3 records found!

返回

ksu_select4.html +ksu_select4.php

End