

2026 CakePHP-Day1

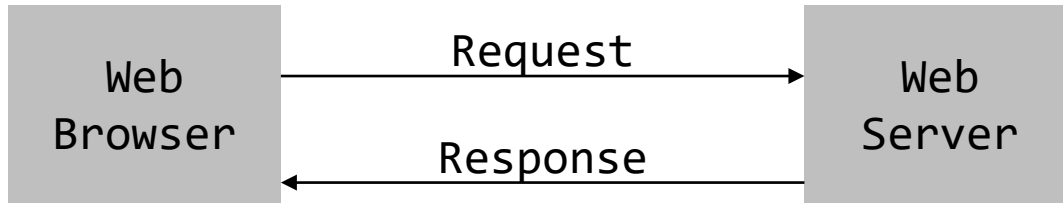
PHP 複習

PHP

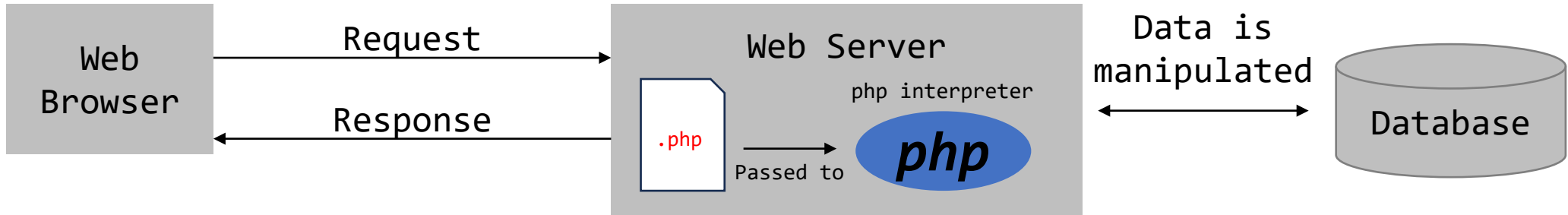
- PHP : Hypertext Preprocessor , 超文字預處理器
- PHP 是一種被廣泛使用在網站後端的伺服器端腳本語言
- 在 Server 上執行 , 結果以純 HTML 形式回傳給瀏覽器
- 與 JS 相比 , PHP 是快速製作動態和互動式網頁的強大工具
- PHP 文件副檔名為 .php , 內容包含文本、HTML、CSS、JavaScript

Web pages

- 靜態網頁 (Static Web pages)



- 動態網頁 (Dynamic Web pages)



Docker Run

- 執行一個新的 Container:
- `docker run --name php -p 8080:80 php:8.5-apache`
- F1 >Dev Containers: Attach To Running Container，選擇剛剛生成的 Container
- 進入 `/var/www/html`，在裡面進行練習。
- 如果不小心開錯工作區，使用快捷鍵 `Ctrl + KF` 來關閉資料夾。
 - 說明：`Ctrl + KF` 是先同時按 `Ctrl` 鍵和 `K` 鍵，放開後再按 `F` 鍵。

Syntax

- PHP 編寫時以 `<?php` 開頭，`?>` 結尾
- `<?php ?>` 可以隨意嵌入在 `html` 中
- 關鍵字不分大小寫

```
<!DOCTYPE html>
<html lang="zh-tw">
<head>
    <title> Hello World! </title>
    <meta charset='utf-8'>
</head>
<body>
    <?php
    echo "Hello World from echo!";
    ECHO"<br>";
    printf("%s", "Hello World from printf!");
    ?>
</body>
</html>
```

01.php

Variable declare

- 以 \$ 開頭
- 可以使用 a-z、A-Z、0-9、_
- 不能以數字開頭、不能用保留字
- 區分大小寫 \$a、\$A
- 字串由「.」連接 e.g. \$a = "Hello" . "World"; // "HelloWorld";
- 有區域、全域範圍之分
- 資料型態：
int, float, bool, null, string, array, object

Output

- `echo` : 可輸出多個變數
- `var_dump` : 輸出變數資訊
- `print` : 只能有一個參數
- `printf` : 可輸出格式化的字串
 - `%d` : 整數
 - `%u` : 無符號整數
 - `%f` : 浮點數
 - `%s` : 字串
- `print_r` : 可輸出陣列
- `""` : 會進行字串插值
- `' '` : 不會解析變數

Output

02.php

```
$a = 3;  
$b = 3.0;  
$c = "Hello" . "World";  
$d = true;  
$e = null;  
$f = [1, 2, 3];  
$g = ["dog" => "狗", "cat" => "貓", "bee" => "蜂"];
```

```
var_dump($a); echo "<br>";  
var_dump($b); echo "<br>";  
var_dump($c); echo "<br>";  
var_dump($d); echo "<br>";  
var_dump($e); echo "<br>";  
var_dump($f); echo "<br>";  
var_dump($g);
```

Output

03.php

```
$test1 = "Hello";  
$test2 = "World!";  
$test3 = $test1 . $test2;  
$arr= [1, 2, 3];  
$arr[] = 4;  
  
echo "$test3"; echo "<br>";  
  
var_dump($arr); echo "<br>";  
  
print 'This is a $test1 {$arr[0]}, {$arr[1]}';  
echo "<br>";  
  
print "This is a $test1 $arr[0], $arr[1]";  
echo "<br>";  
  
printf("This is a %s %d", $test1, $arr[0]);
```

Scope

- `$GLOBALS[]` :
在指令中引用已定義的全域變數
- `global` :
在函數中引用已定義的全域變數
- `static` :
在函數中定義靜態變數

```
$a = 3;  
function test1() {  
    $a = 5;  
    echo "<br>In test1, local a = $a, ";  
    echo "global a=" . $GLOBALS['a'];  
}  
function test2() {  
    global $a;  
    echo "<br>In test2, a = $a";  
    ++$a;  
}  
function test3() {  
    static $a = 0;  
    echo "<br>In test3, a = $a";  
    ++$a;  
}  
test1();  
test2();  
test2();  
test1();  
test3();  
test3();
```

04.php

Constant

- 在程式執行期間無法變更
- 名稱之前沒有 **\$** 符號
- 有兩種定義方法：
 - `define`：可在函式中定義，但會自動成為全域變數
 - 在 `define` 函數中定義時，名稱中**不應該有空格**
 - `const`：必須在全域範圍定義不能在函式中定義

```
const con_g = "const in golbal";
define("def_g", "define in golbal");
function test() {
    // Error
    // const con_l = "const in local";
    define("def_l", "define in local");
    echo con_g . "<br>";
    echo def_g . "<br>";
    echo def_l . "<br>";
};

test();

echo con_g . "<br>";
echo def_g . "<br>";
echo def_l . "<br>";
```

05.php

Operator

- `+`, `-`, `*`, `/`, `%`, `**`, `++`, `--`
- `<`, `<=`, `>`, `>=`, `==`, `!=`, `===`, `!==`
 - `"3.14" == 3.14 // True`
 - `"3.14" === 3.14 // False`
 - `"3.14" != 3.14 // False`
 - `"3.14" !== 3.14 // True`
- `&&`, `||`, `!`
- `=`, `+=`, `-=`, `*=`, `/=`, `%=`
- `<=>` : 太空船運算子
 - `return -1 ( 左邊小 )`, `0 ( 相等 )`, `1 ( 右邊小 )`

Array

- 兩種語法：
 - `array(...)`、`[...]`
- 實際上是一種關聯陣列(key value)
- `key` 必須是整數或字串，若是 `Null`，會強制轉成空字串
- `key` 若為非整數或類似整數的字串會轉成整數
 - e.g. `1.5` → `1`, `true` → `1`, `"1"` → `1`
- `key` 是可選的，會自動分配

```
$array1 = array(
    2 => "a",
    "3" => "b",
    4.5 => "c",
    true => "d",
    null => "e",
);
$array2 = [
    "fruit" => "kiwi",
    "Str1",
    "kiwi" => "fruit",
    "Str2",
    100 => -100,
    -100 => 100,
    "Str3",
];
print_r($array1); echo "<br />";
var_dump($array1); echo "<br />"; echo "<br />";
var_dump($array2);
```

06.php

Branch

- 用法與 C/C++ 相同
 - if, elseif, else
 - switch(...) case ... 可以用字串做標籤

```
$coler = "red";

if ($coler == "red") {
    echo "#ff0000";
} elseif ($coler == "green") {
    echo "#00ff00";
} elseif ($coler == "blue") {
    echo "#0000ff";
} else {
    echo "#ffffff";
};
```

```
$coler = "red";

switch ($coler) {
    case "red":
        echo "#ff0000";
        break;
    case "green":
        echo "#00ff00";
        break;
    case "blue":
        echo "#0000ff";
        break;
    default:
        echo "#ffffff";
        break;
};
```

07.php

Loop

- 用法與 C/C++ 相同
 - for (...)
 - **foreach (...)**
 - while (...)
 - do {...} while (...);
 - break;
 - continue;

```
$a = [  
    "dog" => 1,  
    "cat" => 2,  
    "mouse" => 3,  
    "horse" => 4,  
];  
  
foreach ($a as $k => $v) {  
    echo "Key: $k, Value: $v <br />";  
}  
  
foreach ($a as $v) {  
    echo "$v";  
}
```

08.php

Functions

```
// 無參數
function func1() {
    echo "Hello World!";
};
func1();
```

```
// 有參數
function func2($arg1, $arg2) {
    echo $arg1 . $arg2;
};
func2("Hello", "World!");
```

09.php

```
// 預設參數
function func3(int $arg1 = 10, int $arg2 = 5) {
    echo $arg1 + $arg2;
    echo "<br />";
};
func3(4, 1);
func3(5);
func3(arg2: 10);
func3();
```

```
// 匿名函式
$func4 = function ($name) {
    echo "Hi " . $name;
};
$func4('John');
```

Functions

```
// 回傳值
function addNumbers1(float $num1, float $num2) {
    return $num1 + $num2;
}
echo addNumbers1(1.2, 5.2); // Output: 6.4
```

```
// 繼承變數
$name = "John";
$func5 = function () use ($name) {
    echo "Hi " . $name;
};
$func5();
$name = "Tom";
$func5();
```

09.php

```
// 宣告回傳型態
function addNumbers2(float $num1, float $num2): int {
    return $num1 + $num2;
};
echo addNumbers2(1.2, 5.2); // Output: 6
```

```
// 傳參考
function add_five(&$value) {
    $value += 5;
};
$num = 2;
add_five($num);
echo $num;
```

Form Processing

- PHP 的一種流行用途是處理 HTML 表單的資料
- 有兩種方法可以將表單資料從瀏覽器傳輸到 Web 伺服器：
 - GET、POST
- PHP 中可以透過兩個全域變數來取得傳輸的資料：
 - `$_GET`：取得 `method="GET"` 提交的表單資料
 - `$_POST`：取得 `method="POST"` 提交的表單資料
- `$_SERVER`：取得伺服器和執行環境資訊 (`headers`, `paths`, `script locations`, ...)
- `$_FILES`：取得來自表單的上傳檔案
- `$_COOKIE`：取得 cookies 內容
- `$_SESSION`：取得 `session` 內容
- `$_REQUEST`：取得以 `POST` 和 `GET` 方法提交的資料
- `$_ENV`：取得伺服器端環境變數

Form Processing

```
<!DOCTYPE html>  
<html lang="zh-tw">
```

10.html

```
<head>  
  <meta charset="UTF-8">  
  <title> File Upload </title>  
</head>
```

```
<body>  
  <form action="10.php" method="post" enctype="multipart/form-data">  
    <input type="file" name="myFile">  
    <input type="submit" value="Upload">  
  </form>  
</body>
```

```
</html>
```

Form Processing

```
mkdir uploads
chown -R www-data:www-data uploads
chmod -R 755 uploads
```

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if ($_FILES["myFile"]["error"] == 0) {
        $target_dir = "uploads/";
        $fileName = $_FILES["myFile"]["name"];
        $target_file = $target_dir . basename($fileName);
        if (move_uploaded_file($_FILES["myFile"]["tmp_name"], $target_file)) {
            echo "檔案 " . basename($_FILES["myFile"]["name"]) . " 已上傳。";
            print_r($_FILES["myFile"]);
        } else {
            echo "抱歉，檔案上傳失敗。<br>";
            print_r($_FILES["myFile"]);
        }
    } else {
        echo "請選擇檔案";
    }
}
```

10.php

Form Processing

11.html

```
<!DOCTYPE html>
<html lang="zh-tw">

<head>
  <meta charset="UTF-8">
  <title> Form Test </title>
</head>

<body>
  <form id="form" action="11.php" method="post">
    <fieldset>
      <legend> 帳單資訊 </legend>
      <label for="username"> 帳號 </label>
      <input type="text" name="username" id="username" placeholder="請輸入帳號" required>
      <label for="password"> 密碼 </label>
      <input type="password" name="password" id="password" placeholder="請輸入密碼" required>
      <button type="submit"> 送出 </button>
    </fieldset>
  </form>
  <div id="output"></div>
</body>

</html>
```

Form Processing

11.php

```
<?php
$username = $_POST['username'];
$password = $_POST['password'];

if ($username == "guest") {
    echo "歡迎，訪客！";
} elseif ($username == "admin" && $password == "1234") {
    echo "歡迎，管理員！";
    var_dump($_SERVER);
    var_dump($_GET);
    var_dump($_POST);
} elseif ($username == "user" && $password == "5678") {
    echo "歡迎，使用者！";
} else {
    echo "無效的帳號或密碼！";
}
```

Form Processing

<!-- 將 11.html 拷貝至 12.html，並在 </form> 標籤後加上以下 JS 程式碼 -->

12.html

```
<script>
  const form = document.getElementById('form');
  form.onsubmit = function (event) {
    event.preventDefault();
    fetch('12.php', {
      method: 'POST',
      body: new FormData(form)
    }).then(function (response) {
      return response.json();
    }).then(function (text) {
      document.getElementById('output').innerHTML = text.msg;
      if (text.detail) console.log(text.detail);
    });
  };
</script>
```

Form Processing

12.php

```
<?php
$username = $_POST['username'];
$password = $_POST['password'];

if ($username == "guest") {
    $output = ["msg" => "歡迎，訪客！"];
} elseif ($username == "admin" && $password == "1234") {
    $output = [
        "msg"=> "歡迎，管理員！",
        "detail"=> $_SERVER
    ];
} elseif ($username == "user" && $password == "5678") {
    $output = ["msg" => "歡迎，使用者！"];
} else {
    $output = ["msg" => "無效的帳號或密碼！"];
}
echo json_encode($output);
```

Exercise：統一發票兌獎系統

請使用 **HTML**、**JavaScript** 與 **PHP** 製作一個簡易的「統一發票兌獎系統」。

- 建立一個發票兌獎表單，讓使用者輸入 **8 碼發票號碼**。
- 本練習**不需要連接資料庫**，也不需要從網路取得開獎資料。
- 系統至少需要包含下列獎項：
 - 特別獎、特獎、頭獎、二獎、三獎、四獎、五獎、六獎、未中獎。
- 請建立兩個檔案：
 - `exercise.html`：負責顯示表單與結果。
 - `exercise.php`：負責判斷發票是否中獎。
- `exercise.html` 不可以使用傳統表單送出後重新載入頁面的方式顯示結果。
- 請參考 `12.html` 的作法：
 - 使用 JavaScript `fetch()` 將表單資料傳送至 `exercise.php`。
 - 使用 `POST` 方法傳送資料。
 - 使用 `FormData` 傳送表單內容。
 - 使用 `response.json()` 解析 PHP 回傳的資料。
 - 將兌獎結果顯示在目前頁面中。

Exercise：統一發票兌獎系統

請使用 **HTML**、**JavaScript** 與 **PHP** 製作一個簡易的「統一發票兌獎系統」。

- `exercise.php` 請使用 `$_POST` 取得使用者輸入的發票號碼。
- 請自行建立 **PHP Array** 儲存開獎號碼，例如：
 - 特別獎號碼
 - 特獎號碼
 - 三組頭獎號碼
- 中獎號碼可自行設定，不需要使用真實開獎資料。

Exercise：統一發票兌獎系統

兌獎規則

- 假設頭獎號碼為：12345678
- 則：
 - 12345678 → 頭獎
 - 2345678 → 二獎
 - 345678 → 三獎
 - 45678 → 四獎
 - 5678 → 五獎
 - 678 → 六獎
- 請依照發票號碼的**末幾碼**判斷是否符合各獎項。

Exercise：統一發票兌獎系統

PHP 程式要求

- 至少使用一個 **Array** 儲存中獎號碼。
- 使用 `foreach` 遍歷頭獎號碼並進行比對。
- 至少撰寫一個 **Function** 負責判斷中獎結果。
- 使用 `if`、`elseif` 或其他條件判斷完成兌獎邏輯。
- 判斷時應注意獎項順序，應優先判斷較高獎項，避免同一張發票被判斷成較低獎項。
- 使用 `json_encode()` 將結果轉換成 JSON 格式後回傳。

Exercise：統一發票兌獎系統

JSON 回傳格式

- PHP 回傳結果至少應包含：

```
{  
    "success": true,  
    "invoice": "12345678",  
    "prize": "頭獎",  
    "msg": "恭喜中頭獎！"  
}
```

- 若輸入資料格式錯誤，例如不是 8 碼數字，可回傳：

```
{  
    "success": false,  
    "msg": "請輸入正確的 8 碼發票號碼"  
}
```

Exercise：統一發票兌獎系統

完成後應達成

- 使用者輸入發票號碼並按下「兌獎」後：
 - JavaScript 將資料傳送至 `exercise.php`。
 - PHP 根據寫死的開獎號碼進行判斷。
 - PHP 使用 JSON 格式回傳兌獎結果。
 - JavaScript 接收 JSON。
 - 網頁直接顯示「中獎獎項」或「未中獎」，不重新整理整個頁面。

Bonus

- 完成基本功能後，可以嘗試加入：
 - 顯示各獎項的獎金。
 - 顯示本期所有中獎號碼。
 - 一次輸入多張發票進行兌獎。
 - 使用 CSS 將中獎與未中獎顯示成不同樣式。