HỌC PHẦN "LẬP TRÌNH WEB VỚI PHP"

(Web programming with PHP)

GVGD: ThS. Trần Mạnh Đông

PHP cơ bản

NỘI DUNG (5)

- Chuỗi (Strings)
- Array (Mảng)

Chuỗi (Strings) trong PHP

- Kiểu **strings** (chuỗi hay xâu ký tự): là chuỗi các ký tự (biểu diễn nội dung văn bản - text):
  - Ví dụ "*Lập trình Web với PHP*" là một chuỗi trong PHP
- Các cách để có được chuỗi ký tự trong PHP: 4 cách
  - Sử dụng cặp ký tự nháy đơn ' ': cách đơn giản để có được chuỗi ký tự nằm giữa cặp ' '. Khi dùng kiểu này muốn chứa ký tự ' thì phải viết \', muốn chứa \ thì viết \\

```
$message = 'LẬP TRÌNH WEB VỚI PHP';
//Viết nhiều dòng
$sql = 'SELECT name, email, address
        FROM customer ';
```

Chuỗi (Strings) trong PHP

- Các cách để có được chuỗi ký tự trong PHP:…
  - Sử dụng cặp ký tự nháy kép “ “: các ký tự nằm giữa cặp nháy kép "". Khi dùng cách tạo chuỗi này PHP sẽ tự động chuyển ký hiệu một số ***ký tự đặc biệt***.
    - Ví dụ: \n là LF - dòng mới, \r là CR - đầu dòng, \t là ký tự tab, \$ là ký tự $, \" là ký tự ", \\ là ký tự \, ...

```
$message = “LẬP TRÌNH WEB VỚI PHP”;
echo "Học phần: $a";
//ký tự đặc biệt trong chuỗi
echo "<pre>Ký tự xuống dòng cho văn bản \n khác với thẻ &#60;br&#62; </pre>"
```

Chuỗi (Strings) trong PHP…

- Các cách để có được chuỗi ký tự trong PHP:…
  - Sử dụng cú pháp *heredoc* (*tài liệu đây*): thuận tiện khi sử dụng các ký tự đặc biệt. Heredoc bao giờ cũng bắt đầu bằng ba dấu nhỏ hơn <<<, tiếp theo là một tên định danh do bạn đặt rồi phải xuống dòng ngay, và cuối cùng phải kết thúc bằng định danh; ở một dòng mới

```
<?php
$mon_hoc = 'Lập trình PHP';
$lop_hoc = ' LTWPHP.K13.10';
$str = <<<LTWPHPK1310//Chọn định danh có tên là PHP
- Học phần: $mon_hoc,
- Lớp tín chỉ: $lop_hoc,
- Thời gian học: 20 buổi (4 tín chỉ),
- Email: php@example.com.
LTWPHPK1310;
echo $str;
echo "<br />";
?>
```

Chuỗi (Strings) trong PHP…

- Các cách để có được chuỗi ký tự trong PHP:…
  - Sử dụng cú pháp *nowdoc (tài liệu hiện giờ): giống với heredoc, nhưng bắt đầu bằng <<<'Định-Danh' (có cặp " - heredoc thì không có). Nowdoc vẫn cho phép viết trực tiếp các ký tự đặc biệt nhưng sẽ không phân tích biến đề chèn vào chuỗi*

```
<?php
$mon_hoc = 'Lập trình PHP';
$lop_hoc = 'LTWPHP.K13.10';
$str = <<<'LTWPHPK1310'//Chọn định danh có tên là PHP
- Học phần: $mon_hoc,
- Lớp tín chỉ: $lop_hoc,
- Thời gian học: 20 buổi (4 tín chỉ),
- Email: php@example.com.
LTWPHPK1310;
echo $str;
?>
```

Chuỗi (Strings) trong PHP…

- **Lưu ý**:
  - Khi tạo chuỗi sử dụng dấu nháy kép “” hay cú pháp heredoc, PHP engine sẽ thực hiện phép thay thế biến (thay thế biến là tác vụ thay biến trong chuỗi giá trị của nó. Khi cần, quá trình này sẽ chuyển đổi giá trị sang kiểu chuỗi. Phép này còn gọi là phép nội suy)
  - Khi sử dụng phép thay thế biến, nếu tên biến nằm cạnh văn bản trong chuỗi thì nó sẽ hiểu đoạn văn bản đó là một bộ phận của tên biến nên phép thay thế không hoạt động. Trong trường hợp này, bạn cần phân tách rõ giữa tên biến và văn bản bằng cách đặt tên vào cặp ngoặc nhọn (không kèm ký hiệu $).
  - Khi tạo chuỗi heredoc hay nowdoc, phần tử định danh đầu phải được theo sau ngay bằng dòng mới. Tương tự, phần tử định danh cuối phải được viết trên một dòng và không được bắt đầu bằng khoảng trắng hay tab. Sau phần tử định danh cuối, ký tự duy nhất được phép sử dụng là dấu “;”. Nếu không, sẽ xảy ra lỗi cú pháp

Chuỗi (Strings) trong PHP…

- Nối các chuỗi ký tự trong PHP: sử dụng toán tử nối **.**

Toán tử	Ví dụ	Mô tả
.	$string1.$string2	Nối chuỗi $string1 và $string2
.=	$string1.=$string2	Gắn chuỗi $string2 vào $string1

Chuỗi (Strings) trong PHP…

- Hàm hay dùng về chuỗi trong PHP:
  - Hàm làm việc với độ dài chuỗi và chuỗi con
    - *empty($str)*: Nếu chuỗi rỗng trả về true. Chuỗi rỗng như "", null, '0‘
    - *strlen($str), mb_strlen($str, $encoding):* Lấy chiều dài chuỗi. mb_strlen dùng dạng nhiều byte như Unicode
    - *substr($string, $start ,$length)*: Trích xuất chuỗi, bắt đầu từ vị trí $start với chiều dài $length

Chuỗi (Strings) trong PHP…

- VD1:

Đoạn mã kiểm tra chuỗi để xem chuỗi có rỗng không

```
If (empty($first_name)) {
    $message = 'Bạn phải nhập tên.';
}
```

- VD2:

Đoạn mã lấy độ dài chuỗi và hai chuỗi con

```
$name = 'Ray Harris';
$length = strlen($name);                // $length là 10
$first_name = substr($name, 0, 3);      // $first_name là 'Ray'
$last_name = substr($name, 4);          // $last_name là 'Harris'
$last_name = substr($name, -6);         // $last_name là 'Harris'
```

Chuỗi (Strings) trong PHP…

- Hàm hay dùng về chuỗi trong PHP:
  - ***Hàm tìm chuỗi con***
    - *strpos($str, $substr)*: Kiểm tra xem chuỗi $str có chứa chuỗi $substr. Nếu có thì trả về giá trị vị trí đầu tiên bắt gặp. nếu là FALSE thì không thấy
    - stripos($str1,$str2): Phiên bản khác của strpos, không phân biệt chữ hoa chữ thường
    - strrpos($str1,$str2): Phiên bản khác của hàm strpos, tìm kiếm từ cuối ngược về đầu chuỗi
    - strripos($str1,$str2): Phiên bản khác của hàm strrops, phân biệt chữ hoa chữ thường

Chuỗi (Strings) trong PHP…

- ***Hàm tìm chuỗi con…***
  - VD:

Đoạn mã tìm khoảng trắng trong chuỗi

```
$name = 'Martin Van Buren';
$i = strpos($name,' ');       // $i là 6
$i = strpos($name, ' ', 7); // $i là 10 - dùng offset để tìm
                            // khoảng trắng thứ hai
$i = strrpos($name, ' ');     // $i là 10 - tìm kiếm chuỗi theo chiều ngược
```

Đoạn mã tìm chuỗi con

```
$name = 'Martin Van Buren'; // $i là 7
$i = strpos($name, 'Van');  // $i là FALSE - phân biệt chữ hoa, thường
$i = strpos($name, 'van');  // $i là 7 - không phân biệt chữ hoa, thường
$i = stripos($name, 'van'); // $i là 8 - tìm theo chiều ngược,
                            // không phân biệt chữ hoa, thường
$i = strripos($name, 'A');
```

Đoạn mã chia một chuỗi thành hai chuỗi con

```
$name = 'Ray Harris';
$i = strpos($name, ' ');
if ($i === false) {
    $message = 'Không tìm thấy ký tự trắng trong biến.';
} else {
    $first_name = substr($name, $i);       // $first_name = Ray
    $last_name = substr($name, $i+1);      // $first_name = Harris
}
```

Chuỗi (Strings) trong PHP…

- Hàm hay dùng về chuỗi trong PHP:
  - ***Hàm thay thế bộ phân của chuỗi***:
    - *str_replace($str1, $str_new, $str2)*: Trả về chuỗi mới trong đó tất cả $str1 trong $str2 được thay thế bằng $str_new. Hàm này phân biệt chữ hoa chữ thường
    - str_ireplace($str1, $str_new, $str2): Phiên bản khác của hàm str_replace không phân biệt chữ hoa chữ thường
    - VD:

Đoạn mã thay dấu chấm bằng dấu gạch ngang trong số điện thoại

```
$phone = '554.555.6624';
@phone = str_replace('.', '-', $phone); // $phone là '554-555-6624'
```

Đoạn mã thay thế một chuỗi bằng một chuỗi khác

```
$message = 'Hello Ray';
$message = str_ireplace('hello', 'Hi', $message); // $message là 'Hi Ray'
```

Chuỗi (Strings) trong PHP…

- Hàm hay dùng về chuỗi trong PHP:
  - ***Hàm sửa chuỗi:***
    - *trim($str)*: Trả về chuỗi mới loại bỏ các khoảng trắng thừa ở cả trái và phải
    - ltrim($str): Trả về chuỗi mới loại bỏ các khoảng trắng thừa ở bên trái chuỗi $str
    - rtrim($str): Trả về chuỗi mới loại bỏ các khoảng trắng thừa ở bên phải chuỗi $str
    - lcfirst($str): Trả về chuỗi mới với ký tự đầu được viết thường
    - ucfirst($str): Trả về chuỗi mới với ký tự đầu được viết hoa
    - ucwords($str): Trả về chuỗi mới với ký tự đầu của các từ được viết hoa
    - *strtolower($str), strtoupper($str)*: Chuyển chuỗi thành chữ thường, chuỗi thành chữ in
    - strrev($str): Trả về chuỗi mới nhưng các chữ bị đảo ngược
    - str_shuffle($str): Trả về chuỗi mới với cá ký tự bị trộn ngẫu nhiên
    - str_repeat($str,$i): Trả về chuỗi mới với chuỗi $str được lặp $i lần

Chuỗi (Strings) trong PHP…

- Hàm hay dùng về chuỗi trong PHP:
  - ***Hàm sửa chuỗi***…
    - VD:

Đoạn mã rút gọn và đệm chuỗi

```
$name = '    ray harris    ';
$name = ltrim($name); // $name = 'ray harris     '
$name = rtrim($name); // $name = 'ray harris'
$name = str_pad($name, 13);                    // $name = 'ray harris   '
$name = str_pad($name, 16, ' ', STR_PAD_LEFT ); // $name = '  ray harris  '
$name = trim($name);
```

Đoạn mã thay đổi chữ hoa, chữ thường

```
$name = ucfirst($name);        // $name = 'Ray harris'
$name = lcfirst($name);        // $name = 'ray harris'
$name = ucwords($name);        // $name = 'Ray Harris'
$name = strtolower($name);     // $name = 'ray harris'
$name = strtoupper($name);     // $name = 'RAY HARRIS'
```

Đoạn mã làm việc với thứ tự ký tự

```
$name = strrev($name);         // $name = 'SIRAH YAR'
$name = str_shuffle($name);    // ví dụ, $name = "SHYIRRR AA"
```

Đoạn mã lặp chuỗi

```
$sep = str_repeat('*', 10); // $sep = '**********'
```

Chuỗi (Strings) trong PHP…

- Hàm hay dùng về chuỗi trong PHP:
  - ***Hàm chuyển chuỗi thành mảng***
    - explode($sep, $str): Trả vê mảng các chuỗi con được phân tách thành chuỗi $sep trong chuỗi cha $str. Chuỗi con này có thể có độ dài bất kỳ. Nếu nó là chuỗi rỗng thì mỗi chuỗi con trong mảng sẽ chứa một ký tự đơn
    - VD:

Hướng dẫn chuyển chuỗi thành mảng

```
$names = 'Mike|Anne|Joel|Ray';
$names = explode('|', $name);
$name1 = $names[0]; // $name1 = 'Mike'
$name2 = $names[1]; // $name2 = 'Anne'
```

Chuỗi (Strings) trong PHP…

- Hàm hay dùng về chuỗi trong PHP:
  - ***Hàm chuyển mảng thành chuỗi***
    - implode($sep,$sa): Trả về chuỗi được nối từ các phần tử trong mảng $sa và được phân tách bằng chuỗi $sep. Chuỗi $sep là bắt buộc nhưng có thể bằng rỗng.
    - VD:

Hướng dẫn chuyển mảng thành chuỗi

```
$names = implode('|', $names);  // $names = 'Mike|Anne|Joel|Ray'
```

Hướng dẫn chuyển mảng thành chuỗi phân tách bằng tab

```
$names = implode('\t', $names);
```

Chuỗi (Strings) trong PHP…

- Hàm hay dùng về chuỗi trong PHP:
  - ***Hàm chuyển đổi giữa ký tự và giá trị ASCII***
    - chr($value): Trả về chuỗi với ký tự mà giá trị số ASCII bằng tham số $value truyền vào
    - ord($str): Trả về số nguyên giá trị ASCII của chữ cái đầu tiên trong chuỗi $str
    - VD:

Hướng dẫn chuyển số nguyên sang ký tự

```
$char = chr(65);      // $char = 'A'
$char = chr(66);      // $char = 'B'
```

Hướng dẫn chuyển ký tự sang số nguyên

```
$val = ord('A');      // $val = 65
$val = ord('B');      // $val = 66
$val = ord('Bike');   // $val = 66
```

Chuỗi (Strings) trong PHP…

- Hàm hay dùng về chuỗi trong PHP:
  - ***Hàm so sánh chuỗi***
    - strcmp($str1,$str2): So sánh hai chuỗi và trả về số nguyên chỉ thứ tự của chúng. So sánh này phân biệt chữ hoa, chữ thường (Chữ hoa đứng trước chữ thường)
      - -1: Nếu $str1 đứng trước $str2
      - 1: Nếu $str1 đứng sau $str2
      - 0: Nếu giống nhau.
    - strcasecmp($str1,$str2): Phiên bản khác của strcmp nhưng không phân biệt chữ hoa chữ thường
    - strnatcmp($str1,$str2): Phiên bản khác của strcmp nhưng sử dụng so sánh tự nhiên cho các số trong chuỗi
    - strnatcasecmp($str1,$str2): Phiên bản khác của strcmp nhưng sử dụng so sánh tự nhiên và không phân biệt chữ hoa chữ thường

Chuỗi (Strings) trong PHP…

- Hàm hay dùng về chuỗi trong PHP:
  - ***Hàm so sánh chuỗi…***
    - VD

So sánh chuỗi có phân biệt chữ hoa chữ thường

```
$result = strcmp('Anders', 'Zylka');     // $result = -1 (A trước Z)
$result = strcmp('Anders', 'zylka');     // $result = 1 (z đứng trước A)
$result = strcasecmp('Anders', 'zylka');// $result = -25 (A trước z)
```

So sánh tự nhiên số trong chuỗi

```
$result = strcmp('img06', 'img10');   // $result = -1 (img06 đứng trước)
$result = strcmp('img6', 'img10');    // $result=1 (img10 đứng trước)
$result = strnatcmp('img6', 'img10'); // $result = -1 (img6 đứng trước)
```

So sánh hai chuỗi

```
$result = strnatcasecmp($name_1, $name_2);
if ($result < 0) {
  echo $name_1 . ' đứng trước ' . $name_2;
} else if ($result == 0) {
  echo $name_1 . ' giống ' . $name_2;
} else {
  echo $name_1 . ' đứng sau ' . $name_2;
}
```

Chuỗi (Strings) trong PHP…

- Hàm hay dùng về chuỗi trong PHP:
  - *Một số hàm khác*
    - *str_word_count(string $string)*: Đếm số ***từ*** trong chuỗi (từ chứ không phải chữ)
    - *strrev(string $string)*: Đảo ngược chữ từ cuối lênh đầu
    - *substr($string, $start ,$length)*: Trích xuất chuỗi, bắt đầu từ vị trí $start với chiều dài $length
    - *is_string($var)*: Kiểm tra xem $var có phải là chuỗi
    - …

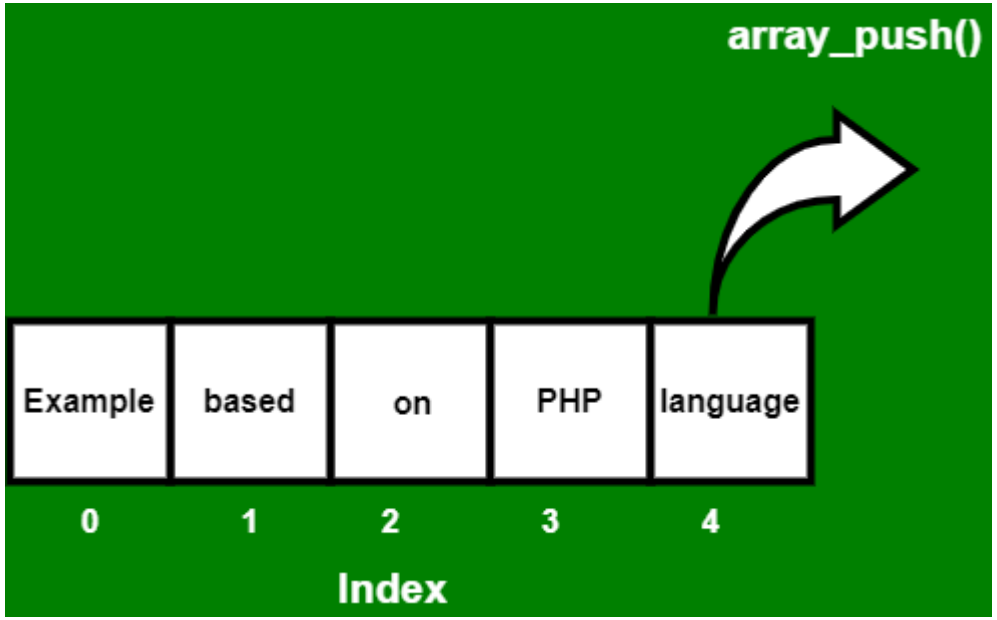
Chuỗi: Demo

Mảng (arrays)

- Mảng (Array) trong PHP là một kiểu dữ liệu chứa một hoặc nhiều thành phần được gọi là phần tử (element).
- Các loại mảng trong PHP
  - Mảng chỉ số (Numeric Array) trong PHP
  - Mảng liên kết/kết hợp (Associative Array) trong PHP
  - Mảng đa chiều (Multidimensional Array) trong PHP

Mảng (arrays)

- Mảng chỉ số (Numeric Array) trong PHP: Mỗi phần tử mảng lưu một giá trị (value) mà bạn có thể tham chiếu tới nó bằng chỉ mục (index). Kích thước của mảng là số lượng phần tử chứa trong mảng



Mảng (arrays)…

- Cú pháp tạo mảng chỉ số

```
$ten_mang = array(gia_tri1, gia_tri2, ...);
```

- Cú pháp tham chiếu tới phần tử mảng chỉ số

```
$ten_mang[index];
```

Mảng (arrays)…

- VD1

Bằng một câu lệnh

```
$names = array('Ted Lewis', 'Sue Jones', 'Ray Thomas');
```

Bằng nhiều câu lệnh

```
$names = array();                    // tạo mảng rỗng
$names[0] = 'Ted Lewis';             // gán ba giá trị vào mảng
$names[1] = 'Sue Jones';
$names[2] = 'Ray Thomas';
```

- VD2:

Bằng một câu lệnh

```
$discounts = array(0, 5, 10, 15);
```

Bằng nhiều câu lệnh

```
$discounts = array();
$discounts[0] = 0;
$discounts[1] = 5;
$discounts[2] = 10;
$discounts[3] = 15;
```

Mảng (arrays)…

- Thêm phần tử mảng chỉ số:
  - Vì mảng trong PHP có tính động nên bạn có thể thay đổi kích thước của mảng bằng cách thêm hoặc bớt các phần tử trong mảng
  - Để thêm phần tử vào cuối mảng, bạn có thể bỏ chỉ mục trong cặp dấu ngoặc vuông khi gán giá trị cho mảng. Khi đó, PHP sẽ lấy chỉ mục cao nhất, tăng lên một đơn vị và sử dụng giá trị đó làm chỉ mục cho phần tử tiếp theo

```
$ten_mang[]=$gia_tri;
```

  - VD:

Hướng dẫn thêm một giá trị vào cuối mảng

```
$letters = array('a', 'b', 'c', 'd');    //  mảng chứa bốn chữ a, b, c, d
$letters[] = 'e';                        // mảng chứa năm chữ a,b,c,d,e
```

Mảng (arrays)…

- Thêm phần tử mảng chỉ số:…
  - Có thể gán phần tử ở bất kỳ đâu trong mảng bằng cách định ra chỉ mục khi gán giá trị cho mảng:

Hướng dẫn gán giá trị cho một chỉ mục cụ thể

```
$letters = array('a', 'b', 'c', 'd');    // mảng chứa bốn chữ a, b, c, d
$letters[0] = 'e';                       // mảng e, b, c, d
$letters[3] = 'f';                       // mảng e, b, c, f
$letters[5] = 'g';                       // mảng e, b, c, f, NULL,g
```

  - Có thể lấy một phần tử mảng qua chỉ mục của mảng

Hướng dẫn lấy giá trị trong mảng

```
$letters = array('a', 'b', 'c', 'd');    // mảng chứa bốn chữ a, b, c, d
$letter1 = $letters[0];                  // lấy ra chữ a
$letter2 = $letters[1];                  // lấy ra chữ b
$letter4 = $letters[4];                  // lấy ra NULL
```

Mảng (arrays)…

- Xóa phần tử mảng chỉ số:
  - Có thể sử dụng hàm unset()

```
Unset($var1,[,var2….])
```

  - Có thể xóa tất cả các giá trị trong mảng bằng cách truyền tên mảng (không có chỉ mục index) cho hàm unset() => Tác vụ này sẽ xóa toàn bộ mảng và các phần tử trong nó

- VD:

Hướng dẫn xóa giá trị trong mảng

```
$letters = array('a', 'b', 'c', 'd');    // mảng chứa bốn chữ a, b, c, d
unset($letters[2]);                      // mảng chứa a, b, NULL, d
unset($letters);                         // $letter là NULL
```

Mảng (arrays)…

- Xóa phần tử mảng chỉ số:…
  - Để xóa các khoảng trống trong mảng: có thể dùng hàm array_values() => Hàm này dùng tên mảng làm tham số và trả về một mảng mới xóa bỏ toàn bộ các khoảng trống. Các phần tử trong mảng mới này được đánh chỉ mục tuần tự từ 0 tới con số nhỏ hơn kích thước mảng 1 giá trị

```
array_values($array)//Trả về mảng mới
```

    - VD:

Hướng dẫn loại bỏ phần tử chứa giá trị NULL và đánh lại chỉ mục

```
$letters = array('a', 'b', 'c', 'd');    // mảng chứa bốn chữ a, b, c, d
unset($letters[2]);                      // mảng chứa a, b, NULL, d
$letters = array_values($letters);       // mảng chứa a, b, d
```

Mảng (arrays)…

- Sử dụng vòng lặp để làm việc với mảng: Một số hàm dành cho các vòng lặp làm việc với mảng
  - count($array): trả về số lượng phần tử trong mảng. Hàm này không đếm khoảng trống trong mảng
  - end($array): Chuyển con trỏ về phần tử cuối cùng trong mảng
  - key($array): Trả về chỉ mục của phần tử mảng mà con trỏ đang nằm ở đó
  - isset($array): Trả về TRUE nếu biến cụ thể hoặc phần tử mảng chứa giá trị, Nếu không, trả về FALSE

Mảng (arrays)…

- Mảng liên kết (Associative Array) trong PHP
  - Mảng liên kết khác với mảng chỉ số (Numeric Array) theo nghĩa là mảng kết hợp sử dụng chỉ mục là *khóa* (key)

```
// PHP engine tự động tạo 1 mảng $array rồi gán cho nó một cặp key => value
$array["key"] = value;

// Hoặc
$array = array('key' => value);
```

Mảng (arrays)…

- Mảng liên kết…
  - VD1
  - VD2:
  - VD3:

Hướng dẫn tạo mảng liên kết chứa mức thuế

Bằng một câu lệnh

```
$tax_rates = array('NC' => 7.75,'CA' => 8.25, 'NY' => 8.875);
```

Bằng nhiều câu lệnh

```
$tax_rates = array();
$tax_rates['NC'] = 7.75;
$tax_rates['CA'] = 8.25;
$tax_rates['NY'] = 8.875;
```

Hướng dẫn cách tạo mảng liên kết chứa số mở rộng của tổng đài điện thoại

```
$ext = array();
$ext[10] = 'Sales';
$ext[13] = 'Customer Service';
$ext[16] = 'Returns';
$ext[18] = 'Warehouse';
```

Hướng dẫn cách tạo mảng chứa cả chỉ mục kiểu số và kiểu chuỗi

```
$employees = array();
$employees[0] = 'Mike';
$employees[1] = 'Anne';
$employees[2] = 'Judy';
$employees['senior'] = 'Mike';
$employees['newest'] = 'Pren';
```

Mảng (arrays)…

- Mảng liên kết…
  - Thêm và xóa phần tử mảng

Gán giá trị cho một khóa cụ thể

```
$name = array('first' => 'Ray', 'last' => 'Harris');
$name['middle'] = 'Thomas';
```

Nếu bỏ qua khóa khi thêm giá trị, điều gì xảy ra

```
$name = array('first'=>'Ray', 'last'=>'Harris');
$name[] = 'Thomas';          // khóa là 0
```

Hướng dẫn lấy giá trị của một khóa cụ thể

```
$name = array('first' => 'Ray', 'last' => 'Harris');
$first_name = $name['first'];
$last_name = $name['last'];
```

Hướng dẫn xóa giá trị khỏi mảng

```
$name = array('first'=>'Ray', 'last'=>'Harris');
unset($name['first']);       // xóa giá trị của phần tử có khóa là first
unset($name);                // xóa toàn bộ các phần tử và $name = NULL
```

Hướng dẫn sử dụng phép thay thể biến với phần tử mảng

```
$name = array('first'=>'Ray', 'last'=>'Harris');
echo "Tên: $name['first']";   // lỗi phân tích cú pháp
echo "Tên: $name[first]";     // tên: Ray
echo "Tên: {$name['first']}"; // tên: Ray
```

Mảng (arrays)…

- Vòng lặp **foreach** duyệt mảng chỉ mục
  - Cú pháp

```
foreach($ten_mang as $ten_bien){
        //các lệnh sử dụng biến $ten_bien
}
```

  - VD 1:

Vòng lặp foreach để hiển thị giá trị trong mảng thông thường

```
$numbers = array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
unset($numbers[2], $numbers[6]);
$number_string = '';
foreach ($numbers as $number) {
    $number_string .= $number . ' ';
}
echo $number_string;        // hiển thị 1 2 3 4 5 6 8 9 10
```

Mảng (arrays)…

- Vòng lặp **foreach** duyệt mảng liên kết
  - Cú pháp

```
foreach($ten_mang as [$khoa =>] $gia_tri){
        //các lệnh sử dụng hai biến $khoa và $gia_tri;
}
```

  - VD2:

Vòng lặp foreach hiển thị giá trị trong mảng liên kết

```
$tax_rates = array('NC' => 7.75,'CA' => 8.25, 'NY' => 8.875);
echo '<ul>';
foreach ($tax_rates as $rate) {
  echo "<li>$rate</li>";
}
echo '</ul>';
```

  - VD3:

Vòng lặp foreach hiển thị khóa và giá trị

```
$tax_rates = array('NC' => 7.75,'CA' => 8.25, 'NY' => 8.875);
echo '<ul>';
foreach ($tax_rates as $state => $rate) {
     echo "<li>$state ($rate)</li>";
}
echo '</ul>';
```

Mảng (arrays)…

- Thao tác với mảng trong PHP: Một số hàm hỗ trợ điền, trộn, cắt và ghép mảng

`range($lo, $hi [, $step]`	Trả về mảng được điền đầy với các giá trị trong khoảng $lo và $hi với bước nhảy giữa các giá trị là $step. Nếu bị bỏ qua, $step được mặc định là 1.
`array_fill($start, $count, $value)`	Trả về mảng được điền đầy với giá trị $value, bắt đầu từ chỉ mục $start và số lượng phần tử là $count.
`array_pad($array, $size, $value)`	Trả về mảng với giá trị $value được thêm vào cuối mảng $array cho tới khi đạt được kích thước $size. Nếu $size âm thì giá trị được thêm vào đầu chuỗi.
`array_merge($array1, $array2, ...)`	Trả về mảng với các phần tử được hợp nhất từ 2 mảng hoặc nhiều hơn. Khóa kiểu chuỗi trong mảng $array2 sẽ ghi đè khóa chữ trong mảng $array1 còn khóa kiểu số được gắn vào đuôi.
`array_slice($array, $index [, $len [, $key]])`	Trả về một phần của mảng bắt đầu từ chỉ mục $index cho tới khi có kích thước $len. Nếu $len bị bỏ qua thì nó trả về tới cuối mảng. Nếu $key là TRUE thì các khóa gốc ở mảng cũ được giữ nguyên trong mảng mới. Ngược lại, mảng cắt lát được đánh lại chỉ mục từ 0.
`array_splice($array, $index [, $len [, $new]])`	Thay đổi mảng $array bằng cách thay thế các phần tử của nó bằng các phần tử trong mảng $new bắt đầu từ chỉ mục $index và kéo dài tới hết độ dài $len. Mảng không được đánh lại chỉ mục. Lưu ý ở đây là hàm này không trả về mảng mới và thay vào đó, hàm sẽ tác động trực tiếp lên mảng $array và thay thế các phần tử của nó bằng các phần tử trong mảng $new.

Mảng (arrays)…

- Thao tác với mảng trong PHP:
  - VD

Hướng dẫn tạo mảng trong khoảng giá trị

```
$numbers = range(1, 4);                 // mảng 1, 2, 3, 4
$numbers = range(10, 22, 4);            // mảng 10, 14, 18, 22
```

Hướng dẫn điền và đệm mảng

```
$numbers = array_fill(0, 5, 1);         // mảng 1, 1, 1, 1, 1
$numbers = array_pad($numbers, 10, 0); // mảng 1, 1, 1, 1, 1, 0, 0, 0, 0, 0
```

Hướng dẫn trộn hai mảng

```
$employees = array('Mike', 'Anne');
$new_hires = array('Ray', 'Pren');
$employees = array_merge($employees, $new_hires);
echo implode(', ', $employees);         // Mike, Anne, Ray, Pren
```

Hướng dẫn cắt từ một mảng khác

```
$employees = array('Mike', 'Anne', 'Ray', 'Pren');
$new_hires = array_slice($employees, 2);
echo implode(', ', $new_hires);         // Ray, Pren
```

Hướng dẫn ghép hai mảng với nhau

```
$employees = array('Mike', 'Anne', 'Joel');
$new_hires = array('Ray', 'Pren');
array_splice($employees, 1, 2, $new_hires);
echo implode(', ', $employees);         // Mike, Ray, Pren
```

Mảng (arrays)...

- Hàng đợi (queue - FIFO) và ngăn xếp (stack -LIFO):
  - PHP cung cấp một số hàm

`array_push($array, $value)`	Thêm giá trị $value vào cuối mảng $array.
`array_pop($array)`	Loại bỏ và trả về giá trị cuối cùng trong mảng $array.
`array_unshift($array, $value)`	Thêm giá trị $value vào đầu mảng $array.
`array_shift($array)`	Loại bỏ và trả về giá trị đầu tiên của mảng $array.

  - VD:

Hướng dẫn làm việc với ngăn xếp

```
$names = array('Mike', 'Joel', 'Anne');
array_push($names, 'Ray');          // mảng Mike, Joel, Anne, Ray
$next = array_pop($names);          // mảng Mike, Joel, Anne
echo $next;                         // hiển thị Ray
```

Hướng dẫn làm việc với hàng đợi

```
$names = array('Mike', 'Joel', 'Anne');
array_push($names, 'Ray');          // mảng Mike, Joel, Anne, Ray
$next = array_shift($names);        // mảng Anne, Joel, Ray
echo $next;                         // hiển thị Mike
```

Mảng (arrays)…

- Hàm thực thi các phép toán trên mảng: Tổng và tích các phần tử
  - Các hàm

`array_sum($array)`	Trả về tổng các phần tử của mảng.
`array_product($array)`	Trả về tích các phần tử của mảng.

  - VD:

Hướng dẫn cách cộng các giá trị trong mảng

```
$prices = array(141.95, 212.95, 411, 10.95);
$sum = array_sum($prices);                    // 776.85
```

Mảng (arrays)…

- Tìm kiếm trong mảng
  - Các hàm

`in_array($value, $array[, $strict])`	Trả về TRUE nếu giá trị $value được tìm thấy trong mảng $array. Nếu $strict là TRUE thì so sánh chỉ trả về TRUE nếu kiểu được so khớp.
`array_key_exists($key, $array)`	Trả về TRUE nếu $key được sử dụng làm khóa trong mảng.
`array_searcg($value, $array[, $strict])`	Trả về khóa nếu tìm thấy $value trong $array hoặc FALSE nếu không tìm thấy. Tham số $strict hoạt động giống như trong hàm in_array.
`array_count_values($array)`	Đếm số lần xuất hiện của mỗi phần tử trong mảng $array. Trả về một mảng mới với giá trị được dùng làm khóa và số lần xuất hiện được dùng làm giá trị.

  - VD:

Hướng dẫn tìm kiếm trong mảng

```
$tax_rates = array('NC' => 7.75,'CA' => 8.25, 'NY' => 8.875);
$is_found = in_array(7.75, $tax_rates);              // TRUE
$is_found = in_array('7.75', $tax_rates);            // TRUE
$is_found = in_array('7.75', $tax_rates, true);      // FALSE
$key_exists = array_key_exists('CA', $tax_rates);    // TRUE
$key = array_search(7.75, $tax_rates);               // 'NC'
```

Hướng dẫn đếm số lần xuất hiện của giá trị trong mảng

```
$names = array('Mike', 'Mike', 'Mike', 'Anne', 'Joel', 'Joel');
$occurences = array_count_values($names);
echo $occurences['Mike'];                            // 3
echo $occurences['Anne'];                            // 1
echo $occurences['Joel'];                            // 2
```

Mảng (arrays)…

- Sắp xếp mảng
  - Các hàm

`sort($array[, $compare])`	Sắp xếp các giá trị trong mảng theo chiều tăng dần và đánh lại chỉ mục. Nếu biến $compare là SORT_REGULAR, thì theo mặc định, từng phần tử sẽ được so sánh bằng toán tử quan hệ. Nếu biến $compare là SORT_STRING, các phần tử sẽ được chuyển sang chuỗi trước khi so sánh. Nếu $compare là SORT_NUMERIC, các phần tử sẽ được chuyển sang số trước khi so sánh.
`rsort($array[, $compare])`	Sắp xếp giá trị theo chiều giảm dần và đánh lại chỉ mục.
`asort($array[, $compare])`	Sắp xếp giá trị theo chiều tăng dần và giữ nguyên chỉ mục.
`arsort($array[, $compare])`	Sắp xếp giá trị theo giảm dần và giữ nguyên chỉ mục.
`ksort($array[, $compare])`	Sắp xếp khóa theo chiều tăng dần và giữ nguyên cặp khóa/giá trị.
`krsort($array[, $compare])`	Sắp xếp khóa theo chiều giảm dần và giữ nguyên cặp khóa/giá trị.

Mảng (arrays)…

- Sắp xếp mảng…
  - VD

Hướng dẫn sắp xếp chuỗi theo thứ tự tăng dần

```
$names = array('Mike', 'Anne', 'Joel', 'Ray', 'Pren');
sort($names);                          // Anne, Joel, Mike, Pren, Ray
```

Hướng dẫn sắp xếp số theo thứ tự tăng dần

```
$numbers = array(520, '33', 9, '199');
sort($numbers, SORT_NUMERIC);      // 9, 33, 199, 520
```

Hướng dẫn sắp xếp theo thứ tự giảm dần

```
$names = array('Mike', 'Anne', 'Joel', 'Ray', 'Pren');
rsort($names);                         // Ray, Pren, Mike, Joel, Anne
```

Hướng dẫn sắp xếp mảng liên kết

```
$tax_rates = array('NC' => 7.75,'CA' => 8.25, 'NY' => 8.875);
asort($tax_rates);                     // sắp xếp theo giá trị (tăng dần)
ksort($tax_rates);                     // sắp xếp theo khóa (tăng dần)
arsort($tax_rates);                    // sắp xếp theo giá trị (giảm dần)
krsort($tax_rates);                    // sắp xếp theo khóa (giảm dần)
```

Mảng (arrays)…

- Thay đổi mảng
  - Các hàm

`array_unique(`*`$array`*`[, `*`$compare`*`])`	Trả về mảng với các giá trị trùng nhau bị bỏ. Tham số $compare hoạt động giống như các hàm sắp xếp nhưng giá trị mặc định là SORT_STRING.
`array_reverse(`*`$array`*`[, `*`$key`*`])`	Trả về mảng với các phần tử bị đảo ngược vị trí. Nếu tham số $key TRUE thì các khóa gốc được giữ nguyên, còn không sẽ bị đánh lại chỉ mục.
`shuffle(`*`$array`*`[,`*`$key`*`])`	Tráo ngẫu nhiên các giá trị trong mảng $array. Mảng được đánh lại chỉ mục.
`array_rand(`*`$array`*`[, `*`$count`*`])`	Trả về khóa ngẫu nhiên trong mảng. Nếu $count lớn hơn 1, nó trả về mảng có số lượng khóa ngẫu nhiên = $count.

Mảng (arrays)…

- Thay đổi mảng…
  - VD:

Hướng dẫn thay đổi mảng

```
$names = array('Mike', 'Mike', 'Mike', 'Anne', 'Joel', 'Joel');
$names = array_unique($names);          // Mike, Anne, Joel
$names = array_reverse($names);         // Joel, Anne, Mike
shuffle($names);                        // Mike, Joel, Anne (ví dụ)
```

Hướng dẫn thay đổi mảng liên kết

```
$tax_rates = array('NC' => 7.75,'CA' => 8.875, 'NY' => 8.25);
$tax_rates = array_reverse($tax_rates, true);
```

Hướng dẫn lấy khóa ngẫu nhiên từ mảng

```
$names = array('Mike', 'Anne', 'Joel', 'Ray', 'Pren');
$key = array_rand($names);              // 2 (ví dụ)
$names_rand = array_rand($names, 3);    // 0, 1, 3 (ví dụ)
```

Hướng dẫn tráo và rút bài từ bộ bài

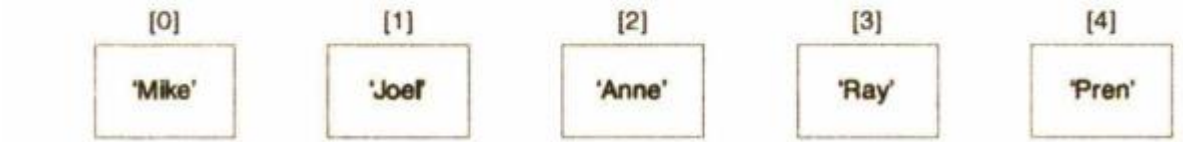
```
// tạo bộ bài
$faces = array('2', '3', '4', '5', '6', '7', '8', '9', 'T', 'J', 'Q', 'K', 'A');
$suits = array('h', 'd', 'c', 's');
$cards = array();
foreach ($faces as $face) {
  foreach ($suits as $suit) {
     $cards[] = $face . $suit;
  };
}
// tráo và rút bài
shuffle($cards);
$hand = array();
for ($i = 0; $i < 5; $i++) {
  $hand[] = array_pop($cards);
}
echo implode(', ', $hand);              // 3c, Js, Qs, Jc, Qc (ví dụ)
```

Mảng (arrays)…

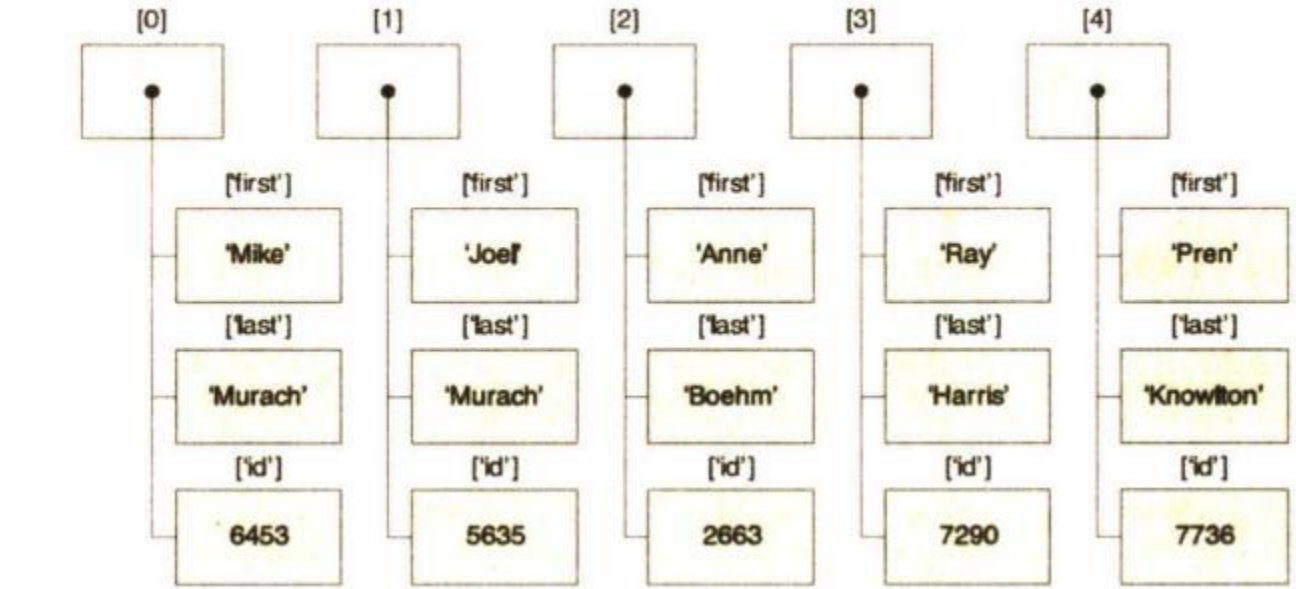
- Mảng đa chiều (Multidimensional Array) trong PHP: Mảng đa chiều trong PHP là mảng lưu trữ các mảng khác dưới dạng các phần tử của chúng. Mỗi chiều làm tăng thêm độ phức tạp, đòi hỏi nhiều chỉ mục để truy cập các phần tử. Các dạng phổ biến bao gồm mảng hai chiều (như bảng) và mảng ba chiều, hữu ích để sắp xếp dữ liệu phức tạp, có cấu trúc.

Mảng (arrays)...

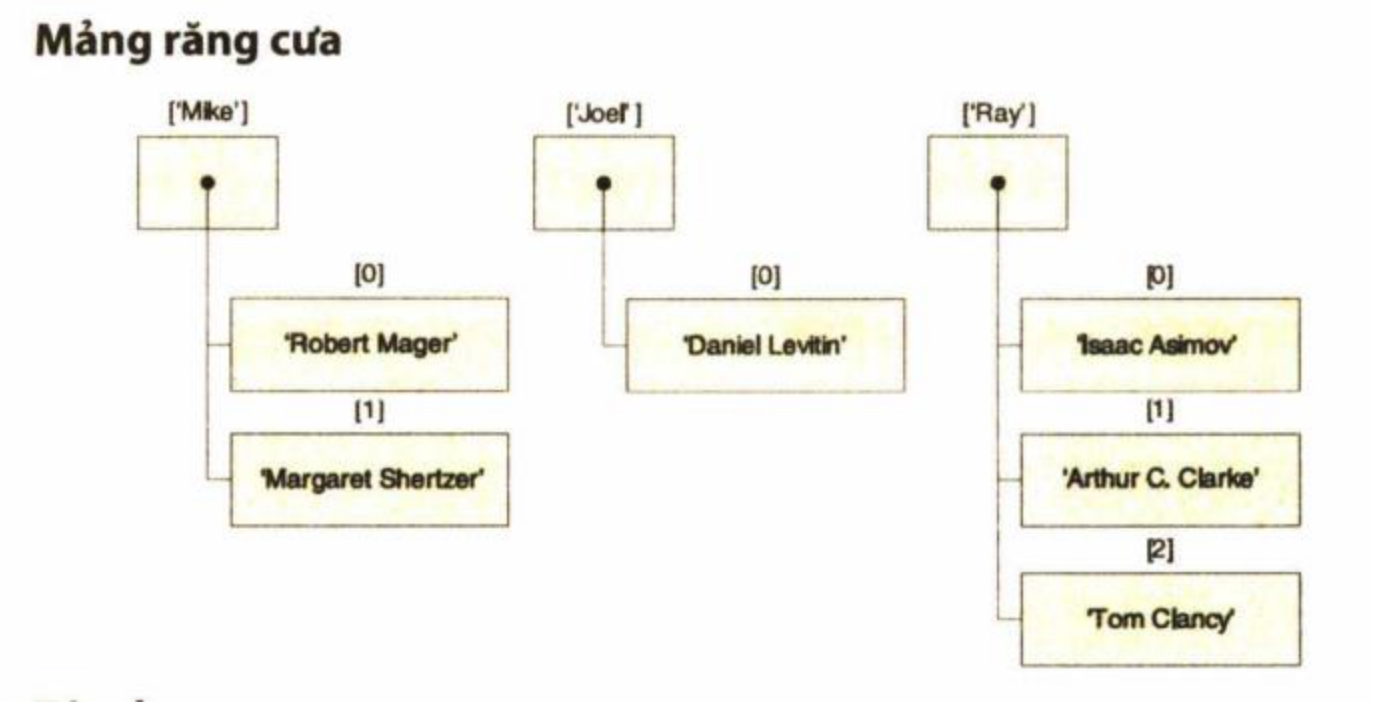
Mảng đơn



Mảng chữ nhật



Mảng (arrays)...


Mảng răng cưa
['Mike']
['Joel']
['Ray']
[0]
'Robert Mager'
[1]
'Margaret Shertzer'
[0]
'Daniel Levitin'
[0]
'Isaac Asimov'
[1]
'Arthur C. Clarke'
[2]
'Tom Clancy'

Mảng (arrays)…

- Tạo và sử dụng mảng hai chiều

Mã tạo mảng hai chiều

```
$times_table = array();                 // tạo mảng rỗng
for ($i = 0; $i <= 12; $i++) {          // thêm 13 phần tử vào mảng
  $times_table[$i] = array();           // các mảng con đều rỗng
}
```

Mã thêm giá trị cho mảng hai chiều

```
for ($i = 0; $i <= 12; $i++) {
  for ($i = 0; $i <= 12; $i++) {
    $times_table[$i][$j] = $i * $j;
  }
}
```

Mã tham chiếu tới từng phần tử trong mảng

```
echo $times_table[4][3];                // 12
echo $times_table[7][6];                // 42
```

Mảng (arrays)…

- Tạo và sử dụng mảng của mảng liên kết

Mã tạo mảng giỏ hàng

```
$cart = array();                    // tạo giỏ hàng rỗng
```

Mã tạo và thêm mảng liên kết vào mảng giỏ hàng

```
$item = array();                    // tạo mảng sản phẩm rỗng
$item['itemCode'] = 123;
$item['itemName'] = 'Visual Basic 2010';
$item['itemCost'] = 52.5;
$item['itemQuantity'] = 5;
$cart[]= $item;                     // thêm sản phẩm vào giỏ hàng
```

Mã tạo và thêm mảng liên kết khác vào giỏ hàng

```
$item = array();                    // tạo mảng sản phẩm rỗng
$item['itemCode'] = 456;
$item['itemName'] = 'C++ 2010';
$item['itemCost'] = 52.5;
$item['itemQuantity'] = 2;
$cart[]= $item;                     // thêm sản phẩm vào giỏ hàng
```

Mã tham chiếu tới các phần tử trong mảng của mảng liên kết

```
echo $cart[0]['itemCode'];          // hiển thị 123
echo $cart[1]['itemName'];          // hiển thị C++ 2010
```

Mảng: Demo

- Khởi tạo mảng
- Các loại mảng
- Thêm/xóa phần tử trong mảng
- Duyệt mảng