FPT Education

FPT POLYTECHNIC

cdio™

Conceive Design Implement Operate

THỰC HỌC – THỰC NGHIỆP

LẬP TRÌNH PHP1

PHP CƠ BẢN 2

http://www.poly.edu.vn

Mục tiêu

- Hàm (function)
- PHP và HTML

PHẦN 1

HÀM (FUNCTION)

- ***Hàm***: là các khối mã (block code) có thể tái sử dụng và chỉ thực thi khi được gọi.
- Hàm có sẵn và hàm tự định nghĩa
- Tạo hàm: sử dụng từ khoá function, theo sau là tên hàm, dấu ngoặc đơn và một khối mã.

```
function ten_ham(){
    //code here
}
```

- Gọi hàm:

```
ten_ham();
```

HÀM (FUNCTION)

- Được phép gọi hàm trước khi khai báo hàm thực hiện.

```
foo(); // allowed
function foo() {}
```

- Ví dụ

```php
<?php

$x = 4;
function assignx () {
    $x = 0;
    print "\$x inside function is $x.";
}

assignx();
print "\$x outside of function is $x.";

?>
```

THAM SỐ HÀM (FUNCTION PARAMETER)

- Hàm với điều kiện if: hàm chỉ được xác định khi điều kiện nhất định được đáp ứng.

```
bar(); // error
if (true) { function bar() {} }
bar(); // ok
```

HÀM (FUNCTION)

❑ *Hàm có tham số và trả về kết quả*

```
function ten_ham(tham_so){
    return gia_tri;
}
```

❑ Ví dụ

```php
<?php

function area($width, $height)
{
    return width * height;
}
echo "Area: " . area(10,20);

?>
```

- ***Hàm có tham số tuỳ chọn (optional parameter)***: có thể chỉ định các giá trị mặc định cho các tham số bằng cách gán cho chúng một giá trị bên trong danh sách các tham số. Sau đó, nếu đối số đó không được chỉ định khi hàm được gọi, thì giá trị mặc định sẽ được sử dụng thay thế.

```
function myFunc($x, $y = ' Earth')
{
  echo $x . $y;
}

myFunc('Hello'); // "Hello Earth"
```

HÀM (FUNCTION)

- *Named arguments (đối số được đặt tên)*: cho phép một đối số được truyền theo bấy kỳ thứ tự nào bằng cách chỉ định tên của tham số tương ứng với nó.
- Tính năng này bổ sung cho optional parameter bằng cách cho phép bỏ qua các giá trị mặc định

```
function myNamed($a, $b = 2, $c = 4)
{
  echo "$a $b $c";
}

myNamed(c: 3, a: 1); // "1 2 3"
```

- *Variable parameter lists (Danh sách các tham biến):* Hàm có thể được khai báo với nhiều đối số hơn.
- `func_get_arg` : để nhận một đối số tại một thời điểm. Hàm này nhận một đối số duy nhất, là tham số được trả về, bắt đầu bằng 0

```php
function myArgs()
{
    $x = func_get_arg(0);
    $y = func_get_arg(1);
    $z = func_get_arg(2);
    echo $x . $y . $z;
}
myArgs('Fee', 'Fi', 'Fo'); // "FeeFiFo"
```

HÀM (FUNCTION)

- `func_num_args:` nhận số lượng đối số được truyền vào
- `func_get_args:` trả về một mảng chứa tất cả các đối số đó.

```
function myArgs2()
{
  $num = func_num_args();
  $args = func_get_args();
  for ($i = 0; $i < $num; $i++)
    echo $args[$i];
}

myArgs2('Fee', 'Fi', 'Fo'); // "FeeFiFo"
```

- Từ PHP 5.6, danh sách các tham số có thể là một *variadic parameter (tham số bất định)*, được biểu thị bằng dấu chấm lửng (...)
- *Variadic parameter* hoạt động như một mảng và phải luôn là tham số cuối cùng trong danh sách.

```
function myArgs3(...$args)
{
  foreach($args as $v) {
    echo $v;
  }
}

myArgs3(1, 2, 3); // "123"
```

ANOMYNOUS FUNCTIONS

- Theo mặc định, bất kỳ biến nào được sử dụng bên trong hàm đều bị giới hạn ở phạm vi local.
- Ví dụ

```
$x = 'Hello'; // global $x

function myFunc()
{
  global $x; // use global $x
  $x .= ' World'; // change global $x
}

myFunc();
echo $x; // "Hello World"
```

ANONYMOUS FUNCTION

- ***Anonymous function (hàm ẩn danh):*** cho phép hàm được truyền dưới dạng đối số và được gán cho một biến. Biến sau đó có thể được gọi như một hàm.
- Được định nghĩa như một hàm thông thường, không có tên cụ thể

```
$say = function($name)
{
  echo "Hello " . $name;
};

$say("World"); // "Hello World"
```

ANONYMOUS FUNCTION

- Anonymous function được sử dụng như callback function.

```
function myCaller($myCallback)
{
  echo $myCallback();
}

// "Hello"
myCaller( function() { echo "Hello"; } );
```

ANONYMOUS FUNCTION

- ***Array_map()*** trả về một mảng chứa tất cả các phần tử của mảng arr1 sau khi đã áp dụng hàm callback
- Ví dụ

```php
function cube($n)
{
    return($n * $n * $n);
}

$a = array(1, 2, 3, 4, 5);
$b = array_map("cube", $a);

print_r($b);
```

ANONYMOUS FUNCTION

- ***Array_map() và anonymous function***
- Ví dụ

```php
$a = array(1, 2, 3, 4, 5);
$b = array_map(function($n)
{
  return $n * $n * $n;
}, $a);
print_r($b);
```

CLOSURE

- ***Closure*** là một anonymous function và nó có thể truy cập các biến bên ngoài phạm vi mà nó được tạo ra. Một đặc điểm nhận dạng **Closure** function là nó sẽ có từ khoá ***use*** phía sau tên của hàm.

```
$x = 1, $y = 2;

$myClosure = function($z) use ($x, $y)
{
  return $x + $y + $z;
};

echo $myClosure(3); // "6"
```

ARROW FUNCTION

- ***Arrow function***: là một cú pháp ngắn gọn hơn của anonymous function (được giới thiệu từ PHP7.4).
- Dùng từ khoá **fn** thay cho **function**
- Chỉ cho phép một biểu thức trả về duy nhất và không sử dụng từ khoá **return**.

```
$x = 1, $y = 2;
$myClosure = fn($z) => $x + $y + $z;
echo $myClosure(3); // "6"
```

GENERATOR

- ***Generator***: là một hàm được sử dụng để tạo ra một dãy các giá trị. Mỗi giá trị trả về với lệnh *yield*. Không giống như return, yield lưu trạng thái của hàm, cho phép tiếp tục (từ vị trí đã dùng) khi được gọi lại.
- Ví dụ

```
function getNum()
{
  for ($i = 0; $i < 5; $i++) {
    yield $i;
  }
}

foreach(getNum() as $v)
  echo $v; // "01234"
```

GENERATOR

- Generator: Trong PHP7, với lệnh *yield from*, cho phép yield giá trị từ generator khác, vòng lặp hoặc mảng khác.
- Ví dụ

```
function countToFive()
{
  yield 1;
  yield from [2, 3, 4];
  yield 5;
}

foreach (countToFive() as $v)
  echo $v; // "12345"
```


demo

PHẦN 2

STRING FUNCTION

- PHP tích hợp sẵn một số hàm để xử lý và thao tác trên chuỗi.

```
$a = 'String';

// Search and replace characters in a string
$b = str_replace('i', 'o', $a); // Strong

// Insert text at specified position
$b = substr_replace($a, 'My ', 0, 0); // My String

// Get part of string specified by start and length
$b = substr($a, 0, 3); // Str

// Convert to uppercase
$b = strtoupper($a); // STRING

// Find position of the first occurrence of a substring
$i = strpos($a, 'i'); // 3 (starts at 0)

// Get string length
$i = strlen($a); // 6
```

RETURN TYPE DECLARATIONS

- PHP7 hỗ trợ khai báo ***kiểu trả về của hàm***.
- Kiểu trả về của hàm được khai báo sau danh sách tham số, bắt đầu bằng dấu hai chấm.

```php
<?php
function sum($a, $b): float {
    return $a + $b;
}

// Note that a float will be returned.
var_dump(sum(1, 2));
?>
```

NULLABLE TYPE

- Từ PHP7.1, một kiểu khai báo được đánh dấu là ***nullable*** bằng cách *đặt trước kiểu đó bằng dấu chấm hỏi*.
- Kiểu nullable cho phép giữ null ngoài bất kỳ giá trị kiểu nào được chỉ định

```
function nullOrInt(?int $i)
{
  echo $i === null ? 'null' : 'int';
}

echo nullOrInt(3); // "int"
echo nullOrInt(null); // "null"
```

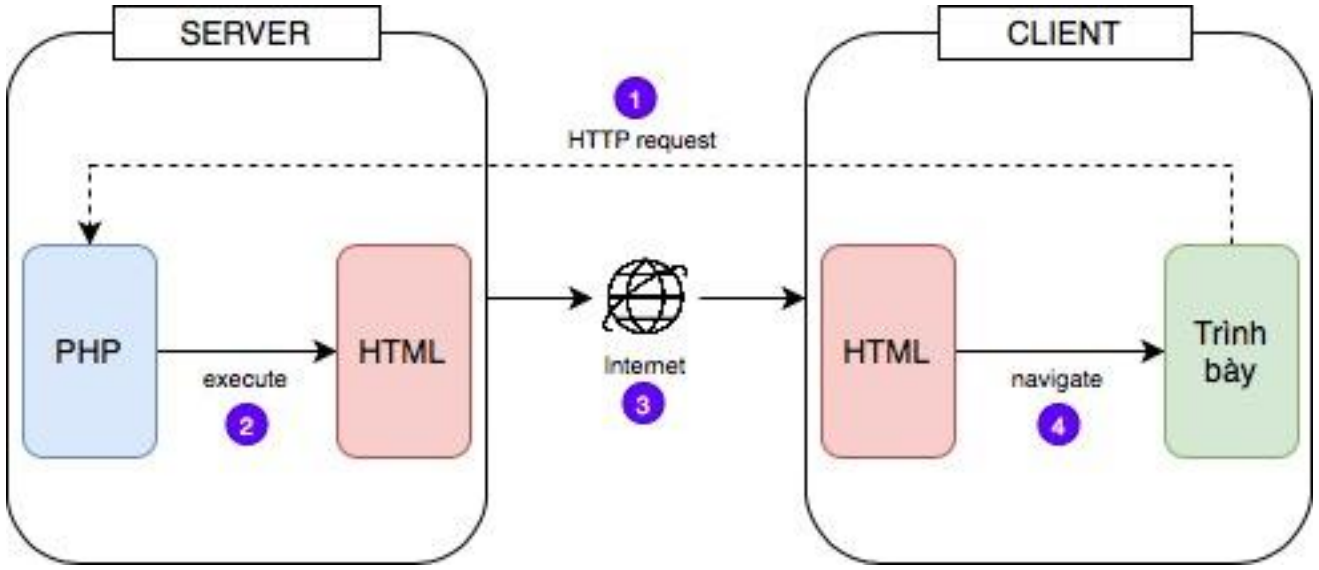
UNION TYPE

- PHP8 giới thiệu ***union type*** cho phép *khai báo kiểu dữ liệu với nhiều kiểu hơn* thay vì 1 kiểu dữ liệu
- Khi khai báo kiểu union, mỗi kiểu được phân tách bằng 1 thanh dọc (|)

```
function squared(int|float $i): int|float
{
  return $i ** 2;
}

echo squared(2); // "4"
echo squared(1.5); // "2.25"
```

PHP VÀ HTML



1. Người dùng yêu cầu 1 trang (page) bằng cách nhập vào URL.
Ví dụ: localhost://php/index.php

2. Máy chủ chạy mã index.php (root của site).
Kết quả của quá trình này là HTML code

3. HTML code được gửi đến client thông qua internet

4. HTML code được thông dịch, sau đó kết xuất được hiển thị trong trình duyệt (browser)

IF...ELSE... VÀ HTML

- Tạo trang HTML bằng code PHP

```php
<?php
    print "<html>";
    print "<head><title>My Program</title></head>";
    print "<body>";
    print "<h1>Hello World</h1>";
    print "</body>";
    print "</html>";
?>
```

IF...ELSE... VÀ HTML

- Code PHP nằm trong HTML

```
<html>
<head><title>My Program</title></head>
<body>
<?php
        print "<h1>Hello World</h1>";
?>
</body>
</html>
```

IF...ELSE... VÀ HTML

- Câu lệnh điều kiện xuất văn bản đến webpage: có thể chuyển đổi chế độ giữa HTML giữa các khối mã PHP.
- Ví dụ:

```
<?php if ($x == 1) : ?>
  This will show if $x is 1.
<?php else : ?>
  Otherwise this will show.
<?php endif; ?>
```

PHP code

HTML

IF...ELSE... VÀ HTML

- HTML và lệnh If trong PHP.
- Ví dụ:

```
<html>
<head><title>My Program</title></head>
<body>
<?php
    $a = 25; $b = 36;
    if( $a > $b) {
        print "$a is greater than $b";
    }
        else {
        print "$b is greater than $a";
    }
?>
</body>
</html>
```

- HTML (heading) và For trong PHP.
- Ví dụ:

```
<html>
<head><title>My Program</title></head>
<body>
<?php
    for($i=1; $i<=6; $i++){
        print '<h'.$i.'>Hello World</h'.$i.'>';
    }
?>
</body>
</html>
```


demo

SUMARRY

- ☑ Hàm (function)
- ☑ PHP và HTML


thank
you!