**PHP BUỔI 1: CÀI ĐẶT, BIẾN, HẰNG, KIỂU DỮ LIỆU, TOÁN TỬ**

I.CÀI ĐẶT:

Tải phần mềm Xampp: <https://www.apachefriends.org/download.html>

Cài đặt xong chạy phần mềm:

A screenshot of a cell phone

Description automatically generated

Tại thanh địa chỉ gõ: <http://localhost>/thư\_mục\_web (hoặc: 127.0.0.1)

\*/ Tùy chỉnh đổi port: (tùy máy)

A screenshot of a cell phone

Description automatically generated

A screenshot of text

Description automatically generated

Thanh địa chỉ gõ: localhost:8080/tên\_thư\_mục\_web

II.LÝ THUYẾT: BIẾN, HẰNG, KIỂU DỮ LIỆU, TOÁN TỬ:

1. **Biến:**

* Biến( *variable*) Trong Lập trình có thể hiểu nôm na là các giá trị có thể thay đổi được.
* Để khai báo biến trong PHP thì nó cũng có một số nguyên tắc sau:
* Biến phải được bắt đầu bằng ký tự $.
* Tên biến phải được bắt đầu bằng các **chữ cái** hoặc dấu \_ **không được bắt đầu bằng số**, và các ký tự khác ngoài.

\***Chú ý**:

* PHP là một ngôn ngữ phân biệt chữ hoa thường. VD: **$vietpro** khác với **$Vietpro**
* Trong PHP chúng ta sử dụng dấu = để gán giá trị cho biến.
* Các bạn cũng có thể dùng echo để in ra biến

Ví dụ:

|  |
| --- |
| <?php  $vietpro = 'Học viện đào tạo lập trình viên số 1 Hà Nội';  echo $vietpro;  // out put: Học viện đào tạo lập trình viên số 1 Hà Nội  ?> |

1. **Hằng:**

*Hằng bản chất là* ***một loại biến****nhưng****không thể****thay đổi giá trị được. Tuy rằng xét về bản chất là giống nhau nhưng cách khai báo thì hoàn toàn khác nhau.*

Cú pháp khai báo hằng như sau:

define('tenHang', 'giaTri');

Trong đó:

* **define**: là cú pháp mặc định để tạo hàm trong PHP.
* **tenHang**: là tên của hằng các bạn muốn đặt.
* **giaTri**: là giá trị của hằng các bạn vừa đặt tên.

Ví Dụ: Mình tạo một hằng **NAME** có giá trị là **Nguyen Van Troi**.

Và các bạn cũng có thể in ra hằng với echo.

|  |
| --- |
| define('NAME', 'Nguyen Van Troi');  echo NAME; |

1. **Kiểu dữ liệu:**

PHP là một ngôn ngữ khá là hay, không giống như C,Java,..  Khi khai báo biến thì chỉ cần khai báo tên biến và giá trị của nó còn kiểu dữ liệu thì tự PHP có thể nhận diện được.

**\*/Trong PHP tổng cộng có 7 kiểu dữ liệu cơ bản như sau:**

* INT (Số nguyên).
* Kiểu số thực (float, double,real).
* Kiểu boolean.
* Chuỗi.
* Mảng.
* Null.
* Đối tượng.

a. Kiểu INT(số nguyên).

Kiểu INT(viết tắt của integer) là một kiểu dữ liệu dạng số nguyên và nó có thể hoạt động dưới nhiều kiểu cơ số khác nhau.

Để khai báo biến dưới dạng INT thì giá trị của biến không được chứa dấu ' hoặc ".

Kiểu INT có kích thước 32bit (khi vượt quá thì nó sẽ tự động chuyển sang kiểu khác).

VD: Mình sẽ khai báo biến với kiểu INT qua các dạng số mà PHP hỗ trợ.

|  |
| --- |
| $a = 1234; // hệ số thập phân  $a = -123; // số âm  $a = 0123; // Bát phân  $a = 0x1A; // Hệ thập lục  $a = 0b11111111; // Hệ nhị phân |

b, Kiểu số thực.

Kiểu số thực các bạn có thể hiểu nôm na là số có phần dư,.. như: 5.5 hoặc 8.9.

Trong PHP kiểu số thực tòn tại ở 2 dạng là **float** và **double**. Cách khai báo cũng tương tự như kiểu INT.

VD:

|  |
| --- |
| $float = 5.586; |

c, Kiểu boolean.

Kiểu boolean trong PHP là một kiểu dữ liệu mà giá trị của nó chỉ tồn tại 2 giá trị TRUE,FALSE (có thể viết hoa, thường cũng được).

|  |
| --- |
| $is\_Male = true; |

d, Kiểu chuỗi.

Kiểu chuỗi trong PHP tồn tại ở hai dạng là String và Char. Để khai báo chuỗi thì giá trị của chuỗi phải được đặt trong cặp dấu ngoặc ' hoặc ".

VD:

|  |
| --- |
| $string = 'vietpro';  //đúng  $string = "vietpro";  // đúng  $string = vietpro;  // sai |

Riêng đối với chuỗi, trong PHP có cung cấp cho chúng ta hàm **is\_string()**kiểm tra xem phần tử có phải chuỗi hay không. Hàm này kết quả sẽ trả về **TRUE** nếu đúng và **FALSE** nếu sai.

VD:

|  |
| --- |
| $a = 'vietpro';  //khai bao bien a  $b = 95;  //khai bao bien b  is\_string($a); // true  is\_string($b); // false |

e, Kiểu mảng.

Kiểu mảng hay còn gọi là array **trong PHP** là một danh sách các phần tử có cùng, hoặc không cùng kiểu dữ liệu. Nó gồm có 2 loại mảng là **mảng một chiều** và **mảng đa chiều** và trong mỗi loại mảng đó nó lại phân ra thành **mảng tuần tự** và **mảng bất tuần tự**. Và để truy xuất phần tử trong mảng thì chúng ta cần phải dự vào vị trí của nó.

***Khai báo mảng***

Để khai báo mảng trong PHP chúng ta có 2 cách khởi tạo:

* Cách 1:

|  |
| --- |
| $array = array(); |

* Cách 2 (Đối với PHP>=5.4):

|  |
| --- |
| $array = []; |

Bên cạnh đó các bạn cũng có thể khai báo kèm thêm với giá trị của mảng tương ứng với 2 cách trên.

|  |
| --- |
| $array = array('giá trị 1', 'giá trị 2'); |

hoặc

|  |
| --- |
| $array = ['giá trị 1', 'giá trị 2']; |

***Hiển thị mảng.***

Trong PHP có cung cấp cho chúng ta hàm **print\_r()** để in ra tất cả các phần tử và vị trí của nó trong mảng.

VD: Mình sẽ khởi tạo một mảng $ten gồm 3 tên Tài, Tuấn, Hà và đồng thời cho in ra giá trị của các mảng luôn.

|  |
| --- |
| $ten = array('Tài', 'Tuấn', 'Hà');  print\_r($ten); |

Khi chạy hàm trên lên thì các bạn sẽ thấy.

|  |
| --- |
| Array ( [a] => Tài [0] => Tuấn [1] => Hà ) |

Nhìn vào kết quả trên các bạn cũng đã thấy được vị trí của mảng sẽ bắt đầu từ 0 cho đến n đúng không (chú ý: chỉ áp dụng với mảng tuần tự).

Giờ nếu muốn in ra một phần tử thì các bạn sẽ phải dựa vào vị trí của phần tử đó.

VD:

|  |
| --- |
| $ten = array('Tài', 'Tuấn', 'Hà');  // khởi tạo mảng  echo $ten[0]; //Tài  echo $ten[1]; //Tuấn  echo $ten[2]; // Hà |

***Thêm phần tử vào mảng.***

Để thêm phần tử vào mảng các bạn làm theo cú pháp:

|  |
| --- |
| $TenMang['key'] = 'value'; |

Trong Đó:

* **Tenmang**: Là tên của mảng.
* **Key**: là key của mảng, nếu không điền thì php sẽ tự thêm vào cuối mảng.
* **Value**: là giá trị phần tử.

|  |
| --- |
| $arr = array('Nguyen', 'Thanh');  $arr[2] = Duy;  // kết quả: ['Nguyen','Thanh','Duy'] |

hoặc

|  |
| --- |
| $arr = array('Nguyen', 'Thanh');  $arr[] = Duy;  // kết quả: ['Nguyen','Thanh','Duy'] |

***Sửa phần tử trong mảng***

Về bản chất thì mảng là một biến nên đương nhiên các bạn có thể chỉnh sửa được. Và để sửa được nó thì bạn cần truy cập vào biến mảng đó giống như hiển thị mảng.

Cú Pháp:

|  |
| --- |
| $mang['index'] = 'giá trị mới'; |

Trong đó:

* **mang**: Là tên của mảng.
* **index**: là key của mảng, nếu không điền thì php sẽ tự thêm vào cuối mảng.
* **giá trị mới**: là giá trị phần tử.

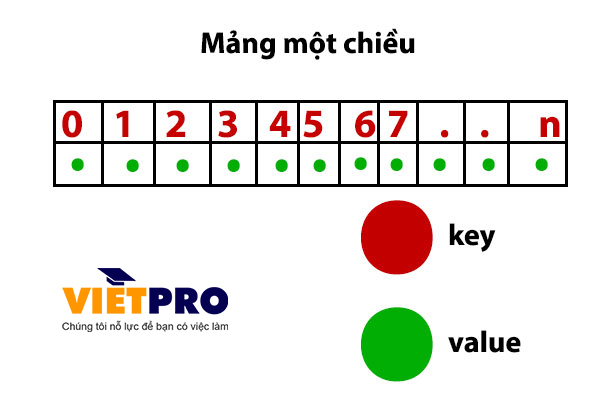
***Mảng bất tuần tự***

Đây là một loại mảng mà có key('Khóa') không theo một tuần tự nào có thể là số, có thể là chữ. Mọi truy vấn cũng giống như mảng tuần tự:

|  |
| --- |
| // khai báo mảng  $mang = [];  // Thêm mảng  $mang[] = 'giá trị';  $mang['key'] = 'giá trị';  // sửa mảng  $mang['key'] = 'giá trị mới'; |

***Mảng một chiều***

Phần trên kiến thức về mảng một chiều và mình xin tóm gọn nó bằng hình ảnh sau:



***Mảng nhiều chiều***

Mảng nhiều chiều(hay mảng đa chiều) thực ra là mảng một chiều mà phần tử của nó là các mảng một chiều hoặc nhiều chiều khác.

A screenshot of a cell phone

Description automatically generated

1. **Kiểu NULL.**

Đây là một kiểu dữ liệu rỗng. Khi khai báo biến kiểu này bộ nhớ sẽ tiếp nhận tên biến mà không tốn thêm bất kỳ một ô nhớ nào.

VD:

|  |
| --- |
| $a = null; |

1. **Kiểu Object.**

Kiểu dữ liệu này mình chỉ nêu tên cho các bạn biết thôi, còn về cách khai báo hay sử dụng nó thì mình sẽ nói thêm ở series PHP hướng đối tượng.

1. **Ép kiểu.**

Phần trên mình đã giới thiệu với mọi người về cách khai báo cũng như hoạt động của nó. Bây giờ Mình sẽ giới thiệu với mọi người cách ép kiểu trong PHP.

Để ép kiểu trong PHP mọi người sử dụng cú pháp:

|  |
| --- |
| $bienMoi = (kieudulieu) $bien; |

Trong đó:

* **$bienMoi**: là biến mới chứa dữ liệu sau khi đã ép.
* **$kieudulieu**: là một trong các kiểu dữ liệu mình nêu ở trên.
* **$bien**: là biến mà bạn muốn ép kiểu (có thể là dữ liệu sẵn).

VD:

|  |
| --- |
| $a = 9; //khai báo biến a kiểu int  $b = (string) $a; // ép kiểu sang string |

Để kiểm tra xem dữ liệu đang ở kiểu nào các bạn có thể dùng hàm **var\_dump();**

1. **Toán tử:**

a. Toán tử gán.

Đây là một loại toán tử thông dụng nhất trong tất cả các ngôn ngữ lập trình. Mà chính các bạn cũng đã biết rồi nhưng các bạn chưa biết thuật ngữ của nó thôi.

Loại toán tử này dùng dấu = để gán giá trị.

VD:

|  |
| --- |
| $a = 5;  $b = 'vietpro'; |

b. Biểu thức.

Dạng toán này các bạn gặp rất nhiều trong khi học toán và trong lập trình cũng được thực hiện như vậy (cũng có khái niệm nhân chia trước cộng trừ sau).

VD:

|  |
| --- |
| $a = 5 + 6; //11  $b = $a + 9; // 20  $c = $b / 5; // 4  $d = $c \* 2; // 8 |

c. Toán tử quan hệ.

Đây là một dạng toán tử sẽ phải sử dụng rất nhiều trong khi lập trình, mà theo như bên toán học thì đây là toán so sánh. Kết quả trả về của toán tử này sẽ là kiểu **boolean** (**TRUE FALSE**).

Bảng Biểu Thức.

|  |  |
| --- | --- |
| Toán tử | Mô Tả |
| > | Lớn hơn |
| >= | Lớn hơn hoặc bằng |
| < | Nhỏ hơn |
| <= | Nhở hơn hoặc bằng |
| == | Bằng |
| != | Khác (Không bằng) |

\*Chú ý:

* Các toán tử có từ 2 ký tự trở lên như <=,>= phải viết liền không được chứa khoảng trắng ở giữa.

VD:

|  |
| --- |
| $a = 5;  $b = 6;  $result = ($a == $b); //False  $result = ($a != $b); //True  $result = ($a < $b); //True  $result = ($a <= $b); //True  $result = ($a > $b); //False  $result = ($a >= $b); //False |

d. Toán tử logic.

Đây là một dạng toán tử kết hợp giữa hai hoặc nhiều giá trị **boolean** hay nói cách khác là kết hợp của hai hay nhiều toán tử quan hệ lại và nó cũng trả về dữ liệu dạng **True False**.

Bảng Logic

|  |  |  |  |
| --- | --- | --- | --- |
| **Toán tử** | **Chú thích** | **Ví dụ** | **Kết quả** |
| && | Trả về true nếu cả hai vế là true | True&&True | True |
| || | Trả về true nếu một trong hai vế là true | True||False | True |
| ! | Trả về true nếu phủ định của nó là true | !False | True |

VD:

|  |
| --- |
| $a = 4;  $b = 4;  $result = ((($a + 7) == 4) && (5 == 0); //false |

e. Toán tử kết hợp.

Toán tử này thực ra là các biểu thức rút gọi của biểu thức.

|  |  |  |
| --- | --- | --- |
| **Toán tử** | **Ví dụ** | **Chú thích** |
| ++ | $a++ | Bằng $a=$a+1; |
| -- | $a-- | Bằng $a=$a-1; |
| \*= | $a\*=$b | Bằng $a=$a\*$b; |
| /= | $a/=$b | Bằng $a=$a/$b; |
| += | $a+=$b | Bằng $a=$a+$b; |
| -= | $a-=$b | Bằng $a =$a-$b; |

VD:

|  |
| --- |
| $a = 5;  $b = 7;  $a \*= $b; // 35 |

---/\*Hết\*/---