**Biến:**

- dung từ khóa global để khai báo biến toàn cục

- global x = $\_GLOBAL[‘x’]

-dùng từ khóa static để lưu trữ biến cục bộ trong hàm sau khi thực thi sẽ k bị reset/xóa.

-var\_dump($x) dung để kiểm tra kiểu dữ liệu của biến

**String:**

-strlen(‘’): in ra độ dài chuỗi

-str\_word\_count(‘’): in ra số lượng từ

-strrev(‘’): đảo ngược chuỗi

-strpos(‘str’,’a’) : trả về index của a

-str\_replace(‘a’,’b’,’string’) : thay thế b vào a

-explode(kí tự phan cách chuỗi, $string): phan cách chuỗi thanh mảng

-implode() : chuyển mảng thanh chuỗi

-str\_word\_count(): đếm số từ trong chuỗi

-str\_repeat($str,n): lặp lại mảng n lần

-htmlentities(): chuyển chuỗi html thanh thực thể

-substr(string,index) Lấy chuỗi con từ chuỗi cha

-strpos(string, a) : tìm index của a trong chuỗi

-trim() loại bỏ khaongr trắng

**Number:**

-is\_int()

is\_float(),

is\_finite(), : kiểm tra có phải số vô hạn (>float)

is\_nan() : kiểm tra có phải NaN

is\_numberic(): kiểm tra có phải số kiểu chuỗi

-Dùng (int)$x để ép kiểu

-round(a,2): làm tròn với 2 là số kí tự sau dấu phẩy

-floor() : làm tròn xg

-ceil(): làm tròn lên

-number\_format(): định dạng số theo kiểu tùy ý

**Math:**

- pi()

-min(),max(…)

-rand(1,10)

-sqrt(): căn bậc hai

-abs()

**Hằng số:**

Define(name,value);

**Toán tử:**

- $x **xor** $y : x đúng khi x hoặc y đúng nhưng k phải cả 2

- Nối chuỗi: $a.$b : nối chuỗi a với chuỗi b

$a.=$b: nối chuỗi b vào chuỗi a và thay đổi giá trị của a

-Nối mảng: $a+$b:

-So sánh mảng: ==, === , !==, !=

-Toán tử 3 ngôi: $x = expr1 ? expr2 : expr3

- giá trị mặc định : $x = expr1 ?? expr2

**Function:**

**-** php là 1 ngôn ngữ đánh máy khá lỏng lẻo: tham số truyền vào của hàm là kiểu int, nhưng ta cố tinh truyền kiểu string vào thì nó sẽ tự động đổi giá trị thanh kiểu int:

- để khắc phục ta khai báo declare(strict\_types=1); vào đầu tệp

-function addNumbers(float $a, float $b) : float {}: dung dấu : để set kiểu cho giá trị trả về của hàm

-function add\_five(&$value) {} dung dấu & trước tham số để tham chiếu tới biến local và có thể thay đổi giá trị của nó

**ARRAY:**

-$cars = array("Volvo", "BMW", "Toyota"); Tạo mảng

-count($array) : để in ra độ dài mảng

- mảng đơn:

-mảng kết hợp: $age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

Foreach($age as $x => $value){}

-Mảng đa chiều: $cars = array (  
  array("Volvo",22,18),  
  array("BMW",15,13),  
  array("Saab",5,2),  
  array("Land Rover",17,15)  
);

* sort()- sắp xếp các mảng theo thứ tự tăng dần
* rsort()- sắp xếp các mảng theo thứ tự giảm dần
* asort()- sắp xếp các mảng kết hợp theo thứ tự tăng dần, theo giá trị
* ksort()- sắp xếp các mảng kết hợp theo thứ tự tăng dần, theo khóa
* arsort()- sắp xếp các mảng kết hợp theo thứ tự giảm dần, theo giá trị
* krsort()- sắp xếp các mảng kết hợp theo thứ tự giảm dần, theo khóa

-array\_values(): đưa mảng tích hợp về mảng đơn

-array\_keys(): đưa….

-array\_pop(). Xóa phần tử cuối ủa mảng và trả về ptu đã xóa

-array……..

**OOP:**

-class:

- $a instanceof A : kiểm tra a có thuộc class A hay k

-contrucsor: function \_\_constructor(…){}

-destructor: đc gọi ngay sau khi khởi tạo đối tượng

Function \_\_destructor(){}

-access modifier:

Public:

Protect: truy cập trong lớp và trong lớp con

Private: truy cập chỉ trong lớp

- kế thừa: class a extends b

- hằng số trong lớp: const MESSAGE = ‘abc’;

Sử dụng trong lớp: self::MESSAGE

Sử dụng bên ngoài lớp: Tên lớp::MESSAGE

-abstract class: là lớp có 1 phương thức trừu tượng( phương thức được đặt tên nhưng chưa triển khai mã) sẽ triển khai mã trong lớp con

Khai báo: abstract class A {abstract protect function(){} }

- interface: khai báo: interface A

-Kế thừa: Class B implements A {}

-Abstract class thì có thể có thuộc tinh và phương thức phải có từ khóa abstract còn interface thì k cần và k có thuộc tinh

-trait: khai báo các phương thức được sử dụng trong nhiều lớp( giải pháp thay thế cho việc PHP ko thể kế thừa từ nhiều class

Khai báo: trait A

Sử dụng trong class khác: use A

-static method: phương thức tinh k cần tạo đối tượng mà vẫn có thể gọi tới

Dùng ngoài class(public): ClassName::methodname

dùng trong mothod khác: self::methodname

Dùng trong method class khác(phải là public): classname::methodname

Dùng trong method class con(public & protect):parent::methodname

-thuộc tinh static: static $a = 3.14 : sử dụng giống static method

- Đa hình: thể hiện qua interface: khi 1 hoặc nhiều lớp sử dụng cùng 1 interface

-Kế thừa:

-Trừu tượnga; Thể hiện qua class, thay vì chúng ta tạo ra 1 lớp cụ thể thì chúng ta tạo ra 1 class bao quát có các thuộc tinh và hành động chung mà nhiều đối tượng cụ thể cần chúng

-Đóng gói: thể hiện qua access modifier