|  |  |
| --- | --- |
| Understand php syntax like: variable, operator, loop,.... | *Variable* : $var\_name = value;  Operator :   * **Toán tử số học** : + - \* / % \*\* * **Toán tử gán** : =, += , -=,….. * **Toán tử so sánh** : ==, ===, !=,… * **Toán tử tăng, giảm** : * ++$x(tăng $x 1 lần, rồi trả về $x) * $x++(trả về $x, rồi tăng) * **Toán tử logic** : and, or , &&, ||,.. * **Toán tử chuỗi** : . (ghép), .=(nối) * **Toán tử mảng** : +, ==, !=,.. * **Toán tử điều kiện** : ?: , ??   Loop : for, while. Do..while , foreach |
| Data type in php | String, Booleans , Int, Float(Double), Array, Object, Null |
| Diffrent between: include, require, include\_one, require\_one | * **Để truyền code từ 1 file sang 1 file khác mà ko cần copy lại** * **Include** thì sai tên địa chỉ code vẫn chạy, **require** báo lỗi * **Include once** **và require once** : Khai báo một lần duy nhất và lần khai báo sau ko tính nữa và code vẫn chạy |
| Super global in php: $\_SERVER, $\_SESSION? | * **$\_SERVER :** chứa thông tin header, đường dẫn và vị trí tập lệnh * **$\_SESSION** : là lưu trữ thông tin sẽ được sử dụng trên nhiều trang trong 1 ứng dụng, các biến php thông thường chỉ làm việc tại trang đó, sang trang khác sẽ mất, còn SESSION thì không * **Một phiên làm việc :** mở ra, thực hiện thay đổi , đóng lại * **Tạo Session bằng hàm :** session\_start() * **Biến SESSION được gắn với biến toàn cục $\_SESSION** |
| What is OOP and the advantage of OOP? | **OPP xoay quanh đối tượng, bọc bởi các class**   * **Tính Đóng gói :** Che dấu những tính chất xủ lý bên trong đối tượng, class A viết thuộc tính, class B viết hàm main để thực thi class A, từ đó hàm B là hàm main chính và sẽ thấy dc tổng quát * **Tính Kế Thừa :** kế thừa lại những **tính năng** mà đối tượng khác đã có , giúp việc tránh lặp code dư thừa * **Tính Đa hình :**  **Method Overloading** (Nạp chồng các method cùng tên nhưng khác tham số * **Method overriding :**  ghi đè **method** ảo của 1 lớp cha( sử dụng từ khóa virtual và overide * **Tính Trừu tượng :** che dấu hoạt động bên trong và chỉ hiển thị những tính năng thiết yếu của đối tượng tới người dùng. **Ví dụ** : người dùng nhắn tin : nhập thông tin, thông tin người nhận và gửi. Còn quá trình gửi đi như nào thì người dùng ko biết mà chỉ biết kế quả đã gửi đi🡺 **quá trình gửi tin nhắn dc ẩn đi mà chỉ hiển thị những chức năng người dùng cần như nhập thông tin….** |
| How to define class, interface and abstract class in PHP? | **Class : chứa các hàm và biến và đối tượng**  **Interface :** **Là 1 bản thiết kế** cho 1 chức năng trong class ( từ khóa : implements)   * Object implements lại interface Phải overwrite lại tất cả phương thức của nó đã định nghĩa còn class bình thường chỉ cần extends * Interface có thể kế thừa lẫn nhau. * Một class có thể implements nhiều interface, * Chỉ khai báo hằng ko khai báo biến * **Quan hệ giữa 1 class khi thực hiện(implements) interface là “can - do ”** * **Interface coi như 1 plugin(phần mềm thứ 3 ) cho class để thêm vào(cài phần mềm)**   **Abstract : là 1 class cha cho tất cả class cùng bản chất(kiểu, loại, nhiệm vụ của class)( bản thiết kế đầy đủ)**   * Có thể khai báo biến, * Sẽ đinh nghĩa hàm(phương thức) và các lớp con sẽ kế thừa và overwrite lại (tính đa hình) và có mức protected và public * **Quan hệ giữa 1 class khi kế thừa(extends) abstract là “is - a ”** * **Ví dụ :** * abstract class máy móc( ô tô, máy bay) * Abstract class độngvật (chó,chim) * Interface chạy,bay * Chim và máy bay đều interface bay * Ô tô và chó đều interface chạy * Nhưng chúng thuộc 2 loại khác nhau và 1 class có thể có cả 2 interface chạy và bay(kế thừa) * **Tóm lại : 1 class có thể có nhiều interface ăn,ngủ…** * **1 class abstract cha là animal có các interface, sau đó abstract kế thừa lại thằng cha sẽ có thể dùng interface dc** |
| How to create new instance from class? | Tạo một lớp bằng từ khóa **NEW**  Tạo bằng **New Instance :** tạo nếu biết tên class và hàm tạo mặc định công khai |
| Diffrerent between Class and Object? | **Object** là 1 thể hiện của 1 Class  **Class** là 1 bản thiết kế từ đó các đối tượng được tạo, là một nhóm các đối tuowgj giống nhau  Ví dụ :  **Class:** Human **Object:** Man, Woman  **Class:** Fruit **Object:** Apple, Banana, Mango, Guava wtc.  **Class:** Mobile phone **Object:** iPhone, Samsung, Moto  **Class:** Food **Object:** Pizza, Burger, Samosa |
| Constructor and Destructor method? | **Constructor :**  hàm tạo có nhiệm vụ khởi tạo giá trị cho các thuộc tính của đối tượng của lớp đó trước khi đối tượng đc sử dụng  **Destructor :** được gọi đến nếu như 1 object ko dc tham chiếu đến nữa hoặc chương trình trong quá trình tự tắt( để giải phóng bộ nhớ,đogs file) |
| What is Trait, how to use it? | **Trait** là một nhóm các phương thức mà bạn muốn include nó trong 1 class khác , giảm việc trùng code khi k dùng tính đa kế thừa phức tạp  Khác với abstract là ko cần dựa vào sự kế thừa |
| member scope: public, protected, private? | **Private :** truy cập mức nội bộ , dùng trong chính class của nó chứ ko ra ngoài  **Protected:** truy xuất mức nội bộ ở lớp đó và lớp kế thừa nó  **Public :** truy cập thuộc tính và phương thức ở bất cứ đâu |
| Inheritance in php | **Kế thừa :** Class B kế thừa từ Class A thì Class B sẽ mang tất cả thuộc tính và phương thức của A  **Ví dụ** : **object động vật** có   * **Phương thức :** ăn , ngủ chạy * **Thuộc tính :** chân mắt mũi   **Object con chó extends động vật**   * **Con chó có các phương thức và thuộc tính của object động vật**   **Nạp Chồng ( Override) :** overide lại các phương thức của class cha  Ví dụ : **ghi đè thuộc tính(**biến)  Class dongvat {  Var $chan = 0;  Function an(){  Return “đang ăn”;  }  }  Class concho extends dongvat{  Var $chan = 4;  Function an(){  Return “ăn xong”;  }  }  **$concho = new concho();**  **Echo $concho->chan;**  **Echo $concho->an();**   * **Kết quà : 4** * **Kết quả : ăn xong** |
| What is array? How Many type | **Array :**  mảng chứa nhiều hơn 1 giá trị  **Type :**   * Mảng kiểu số nguyên * Mảng kiểu liên kết “=>” * Mảng đa chiều( nhiều hơn 1 mảng) |
| Difference between asocial vs squences array | **Array squences : mảng nối tiếp,** mảng là 1 đối tượng có độ dài cố định và sẽ có khe cho mỗi ptu của nó  **Sử dụng hàm : range() tạo ra :** một mảng chứa một phạm vi phần tử(trả về ptu từ thấp đến cao )  **Ví dụ :**  <?php $number = range(0,5); print\_r ($number); ?>   * Array ( [0] => 0 [1] => 1 [2] => 2 [3] => 3 [4] => 4 [5] => 5 ) |
| How to define multidimension array( mảng đa chiều) | **Mảng đa chiều :** là 1 mảng chứa 1 hoặc nhiều hơn 1 mảng  **Ví dụ : mảng 2 chiều**  $cars = array   (   array("Volvo",22,18),   array("BMW",15,13),   array("Saab",5,2),   array("Land Rover",17,15)   );   * **$cars** chứa 4 array và 2 số(giá trị) * Để trỏ tới phần tử của $car array thì phải trỏ tới vị trí của 2 số   <?php echo $cars[0][0].": In stock: ".$cars[0][1].", sold: ".$cars[0][2].".<br>"; echo $cars[1][0].": In stock: ".$cars[1][1].", sold: ".$cars[1][2].".<br>"; echo $cars[2][0].": In stock: ".$cars[2][1].", sold: ".$cars[2][2].".<br>"; echo $cars[3][0].": In stock: ".$cars[3][1].", sold: ".$cars[3][2].".<br>"; ?>   * **Kết quà : Volvo** in stock : **22**, sold : **18** * **…** |
| How to add/remove/edit data in array? | **Add :** array\_push(array, value1, value2, ...);  **Remove:** unset($biến cần xóa);  **Edit :** array\_replace(array1, array2, array3, ...);  **Ví dụ Edit :**  <?php  $a1=array("red","green");  $a2=array("blue","yellow");  print\_r(array\_replace($a1,$a2));  ?>   * **Kết quả :** Array ( [0] => blue [1] => yellow ) |
| Array functions: count, array\_keys, array\_values, array\_filter, array\_column, array\_map, array\_merge, list, implode, explode,... | **Count :** đếm số lượng ptu mảng  **Array\_keys** : trả về các từ khóa mảng  **Array\_values :** trả về giá trị của mảng **(** lọc theo số lẻ, chẵn…)  **Array\_column :** in ra giá trị cột  **Array\_map:** gửi giá trị của mảng vào 1 function và sẽ trả về kết quả mảng đó sau khi áp dụng function  Ví dụ : 1,2,3 có function phép nhân => 1\*2\*3 = 6  **Array\_merge :** gộp 2 hoặc nhiều mảng thành 1 mảng duy nhất  **List():** gán giá trị cho danh sách các biến trong 1 thao tác(dùng mảng số)   * **Ví dụ** :   $my\_array = array(“dog”,”cat”,”pig”);  List($a,$b,$c)=$my\_array;  Echo “I have $a, $b, $c”   * I have dog, cat , pig.   **Implode(“”,$array) :** in ra string cho các ptu mảng <https://www.w3schools.com/php/func_string_implode.asp>  **Explode(“”,$array) :** từ 1 string parse ra kiểu mảng  <https://www.w3schools.com/php/func_string_explode.asp> |
| How many way to loop through all element in array? | **5 way :** While,For,Foreach,Do While, Array Iterator(trình lặp mảng) |
| Sort array: the utilities functions? | **Sắp xếp mảng:**   * **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 * **arsort** khóa ) - 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 * **Mảng kết hợp là mảng liên kết** |
| What is Date Time object in php? | **Object Date Time :** đối tượng đại diện cho ngày giờ |
| How to format Date/Time? | **Hàm date\_format() :** trả về 1 ngày được định đạng theo định dạng đã chỉ định  <?php $date=date\_create("2013-03-15"); echo date\_format($date,"Y/m/d H:i:s"); ?>  Trả về biến date phải là date\_create |
| How to parse Date/Time string into date/time object? | **Chuyển đổi date/time dạng string về object :**  $dateString = 'Wed, 28 Dec 2011 13:04:30 GMT';  $dateTime = datetime::createfromformat('D, d M Y H:i:s e',$dateString);  echo $dateTime->format('d-M-Y H:i:s e');  keys : createfromformat(string,$biến string) |
| Operators with date/time like: add, subtract, diff,...? | **Strtotime($date1 - $date2)**  **diff** |
| What is TimeZone, how to use it on PHP? How to configure default timezone in php? | **Time zone : múi giờ**  **Ví dụ :**  <?php date\_default\_timezone\_set("Asia/Bangkok"); echo date\_default\_timezone\_get(); ?>  **Configure :**   * Go to your phpinfo() page and search for Loaded Configuration File and open the php.ini file mentioned under that section. * 2) Change the default timezone settings by adding your new timezone by modifying this line: date.timezone=Asia/Kolkata. * 3) Save the php.ini file and stop the Apache server. * 4) Restart the Apache server. The timezone settings should now be modified. |
| What is String in php? | **Một tập hợp các ký tự** |
| Diffrerent between use single-quote and double quote to define string in php? | **Single-quotes :** tất cả hiểu là string  **Double-quotes :** có thể hiểu cả $biến và sẽ echo ra giá trị của $biến đó |
| Special character in string like \n, \r, \t,...? | **\n :** xuống dòng  **\t** : tab  **\r** : về đầu dòng  … |
| How to manipulation with string like: concat, replace, reverse, implode, explode,...? | **Concat** : nối các chuỗi lại với nhau, dùng dấu “.”  **Ví dụ :** <?php  // First String  $a = 'Hello';  // Second String  $b = 'World!';  // Concatenation Of String  $c = $a.$b;  // print Concatenate String  echo " $c \n";  ?>  **Str\_replace() :** thay 1 từ trong chuỗi bằng từ khác  <?php echo str\_replace("world","Peter","Hello world!"); ?>   * Hello Peter!   **Array\_Reserve():** đảo ngược ký tự chuỗi  **Ví dụ :**  <?php $a=array("a"=>"Volvo","b"=>"BMW","c"=>"Toyota"); print\_r(array\_reverse($a)); ?>   * Array ( [c] => Toyota [b] => BMW [a] => Volvo ) |
| What is regular expression? Why do we need it? | **Regex :** biểu thức chính quy, chuỗi ký tự tạo thành 1 định dạng cụ thể |
| Regex syntax in PHP? | **preg\_match($parten,$subject,&$matches)**   * $parttern : regex // * $subject : chuỗi cần kiểm tra * $matches : kết quả trả về, dạng tham số   **Ví dụ :**  $subject = "freetuts.net";  $pattern = '/net/';  preg\_match($pattern, $subject, $matches);  echo '<pre>';  print\_r($matches);  echo '</pre>';   * Array( [0] => net ) |
| How to extract all numbers from string use regex? | **Sử dụng pattern /d+ :** in ra toàn bộ số (digit) |
| How to replace String using regex? | **Preg\_replace($pattern,$replacement,$subject)**   * $replacement: chuỗi replace thành * $subject : string muốn duyệt và replace   **Ví dụ :**  $partern = '/hi/';  $subject = 'hi everybody, hi there';  $replacement = 'hello';  echo preg\_replace($partern, $replacement, $subject);   * **Hello everybody, hello there** |
| How to open and read a file in php?  write content to file? | **fopen($path,$option) : mở**   * $**path** : đường dẫn * $**option**: quyền cho phép thao tác ví dụ như r(read only) , r+(read and write)…   **Fread($fp,$size) : đọc file**   * **$fp :** đối tượng lúc mở file * **$size :** kích cỡ file cần đọc( dùng filesize($path) để lấy size)   **Fwrite($fp,$content) : ghi file**   * **$fp :** đói tượng lúc mở file * **$content :** nội dung muốn ghi   **Fclose ($fp) : đóng file** |
| What is HTTP method? How many method do you know? | **Truyền tải siêu văn bản HTTP method :** là cách client(trang web) yêu cầu server ( database) làm gì với request của mình và ngược lại  9 method nhưng phổ biến là  GET, POST, PUT, DELETE |
| How to submit HTML form into php? | **Form action :** đường dẫn truyền đến  **Phương thức truyền:** get,post…..   * **Và nút submit sẽ trỏ tới** |
| Diffrent between GET and POST? | **GET : lấy dữ liệu từ server (không bảo mật)**  **POST : truyền dữ liệu lên server(bảo mật)** |
| How to get Form variables? | **$\_GET[‘tên của form cần lấy giá trị biến’]**  **$\_POST[‘tên của form cần lấy giá trị biến’]** |
| Diffrenent between $\_GET, $\_POST and $\_REQUEST | **$\_GET :** lấy biến từ một chuỗi truy vấn ( câu lệnh sql)  **$\_POST :** lấy biến từ phương thức POST, như là trong 1 form  **$\_REQUEST** : là gộp của 2 phương thức trên nơi mà **$\_POST** có thể override **$\_GET** |
| The different between Form has file upload vs normal form? | **Form upload thì dc mã hóa các thông số vì post truyền username và mật khẩu cần dc bảo mật** |
| How to upload files? | **Tạo form có thuộc tính** enctype="multipart/form-data"  Dùng phương thức POST  Input sẽ có type =”file”  …  Viết php ( code trong htdocs) : keys : **move\_uploaded\_file()**  **SAVE : $\_FILE** |
| What is COOKIES/SESSION? | **Cookies :** đoạn văn bản mà một web server có thể lưu trên ổ cứng, **cho phép website lưu thông tin trên pc người dùng** và sau đó lấy lại nó như đăng nhập mật khẩu….  **SESSION** : **là cách giao tiếp của client vs server**, **bắt đầu** khi client gửi request đến server, **tồn tại** từ trang này tới trang khác của ứng dụng và chỉ **kết thúc** khi hết timeout hoặc đóng ứng dụng . Giá trị lưu vào 1 tệp tin tạm trên server   * Bộ lưu trữ tập tin tạm |
| How to read data from cookies and session? |  |
| How to set data into cookies/session? | **COOKIE:**  **Tạo bằng hàm setcookie(name, value, expire, path, domain, secure, httponly);**  **Ví dụ :**  *<?php $cookie\_name = “user”; $cookie\_value = “Đức Anh”; setcookie($cookie\_name, $cookie\_value, time() + (86400 \* 15), “/”); // 86400 giây tương đương 1 ngày ?>*  *Hàm cookies phải tạo trc tag <html>*  */ : có nghĩa là cookies có giá trị trên toàn bộ website*  ***SESSION***  *Tạo session bằng hàm* ***session\_start()(*** *hồi phục or tạo session mới)*  *Session lưu trữ trong biến toàn cục* ***$SESSION***  *Lưu giá trị SESSION :* $\_SESSION['session\_name'] = $session\_value  Hủy session\_unset() |
| When session/cookie is deleted? | **Cookie :** nếu không có thuộc tính hết hạn thì sẽ giữ lại cookies cho đến khi phiên hiện tại kết thúc ( do người dùng tự tác động)  **Session :** tắt trình duyệt là xóa session => dùng cookies để lưu trữ sẽ tối ưu |
| How to set expire time for cookies/session? | **Cấu trúc cookies là**  **setcookie(name, value, expire, path, domain, secure, httponly);**  mặc định 1 ngày là 86400 giây  keys : **time() (86400 \* 15)** : nghĩa là 15 ngày |