PHP

**Chương 1 :**

PHẦN 1 : BIẾN – KIỂU DỮ LIỆU

1. Các kiểu dữ liệu trong PHP : String , Numberic , Boolean , Null , Array , Object
2. Xác định kiểu dữ liệu
   1. Gettype() : chỉ xuất ra kiểu dữ liệu
   2. Var\_dump() : xuất ra cả kiểu dữ liệu và độ dài
3. Xét kiểu dữ liệu cho 1 biến :
   1. Settype(biến,”kiểu dữ liệu trả về”)
   2. Ép kiểu : kiểu dữ liệu đứng trước (int) tên biến
4. Kiểm tra kiểu dữ liệu của 1 biến : trả về 1 là ok , k trả về gì thì là k phải
   1. Is\_numberic();
   2. Is\_float();
   3. Is\_string();
   4. Is\_array();
   5. Is\_object();
5. Định nghĩa hằng số :
   1. Define(“PI”,3.14);

PHẦN 2 : TOÁN TỬ TRONG PHP

1. Toán tử số học : rất nhiều

PHẦN 3 : LÀM VIỆC VỚI FORM

1. Hiểu bản chất của form : dùng để nhận các giá trị mà người dùng nhập vào thì thông tin được gửi đến các file xử lý và file xử lý sẽ tiến hành xử lí
2. Khi tạo form cần chú ý :
   1. Method : cách thức dữ liệu được gửi đi (post hoặc get)
      1. Post : dữ liệu gửi gầm và k nhận thấy dữ liệu
      2. Get : dữ liệu được gửi trên đường dẫn
   2. Action : xác định trang , tập tin các dữ liệu được gửi đến để xử lý
   3. Name : tên của form
3. Gg template login-form : để tìm kiếm 1 mẫu template đẹp
   1. Vào gg tìm kiếm
   2. Dowload về và giải nén : chỉ quan tâm 2 tập tin css vs index.html
4. Lấy dữ liệu người dùng nhập vào form : theo name
   1. Đối với form có method= “post” sử dụng $\_POST
   2. Đối với form có method= “get” sử dụng $\_GET
   3. Sử dụng $\_REQUYEST cho cả 2 trường hợp trên
5. Pagination.php Gắn tham số vào đường dẫn để lấy giá trị của nó ?page=1 : nằm trên url -> ở các thẻ a gắn link vào cho nó :
   1. <a href="pagination.php?page=1">1</a>
   2. Dùng $\_get để lấy ra
   3. Sử dụng hàm isset() để kiểm tra nó có tồn tại hay k

PHẦN 4 : CÂU ĐIỀU KIỆN

* + - 1. Có 2 loại câu điều kiện IF ELSE vs swith
      2. Bài tập mô phỏng máy tính bỏ túi :
         1. Khi muốn lấy dữ liệu từ form thì lấy theo biến $\_POST hoặc $\_GET và lấy theo tên
         2. Khi lấy thì nhớ dùng mà isset để kiểm tra xem biến đó đã tồn tại hay chưa để tránh lỗi
         3. Để giữ lại các kết quả khi người dùng ấn submit form thì chúng ta dùng cách thêm value vào form và echo giá trị đó ra ở phần value nhưng khi thực hiện sẽ thấy thông báo lỗi là do chúng ta chưa gán giá trị mặc định cho nó
         4. Phải dùng nháy đôi mới ra
         5. Phải kiểm tra xem số thứ 1 vs số thứ 2 có phải là số hạng hay không thì mới thực hiện phép toán

Phần 07- Function

06- biến toàn cục và biến cục bộ : Glob al

07-08 Tham chiếu- tham trị

Tham trị : khi truyền biến vào hàm sau khi kết thúc hàm thì giá trị của biến k thay đổi

Tham chiếu : sau khi kết thúc hàm thì giá trị của biến sẽ thay đổi : thêm dấu & trước tên biến

09-Hàm mở rộng

Phân biệt giữa include và require : cùng có chức năng kéo một file nào đó vào file hiện tại. nếu gặp lỗi thì require sẽ ngừng thực hiện chương trình, Include sẽ tếp tục thực hiện

Include\_one : kiểm tra xem file đó đã được kéo vào chưa nếu rồi thì thôi chưa thì mới kéo

**Chương 2**

**01-introdute**

* Sử dụng var\_dump hoặc gettype để lấy kiểu dữ liệu của một biến
* Sử dụng is\_numberic, is\_string, is\_array , is\_date để kiểm tra kiểu dữ liệu của một biến

**02-Arr**

**01:**

* + **Mảng là một biến đăc** biệt có thể lưu trữ nhiều giá trị
  + **Có 3 loại mảng** : số nguyên , mảng kết hợp, mảng đa chiều
  + **Count đếm số phần tử trong mảng**

**02.**

**- mảng số nguyên là mảng m**à các chỉ số của các phần tử phải thuộc kiểu số nguyên(mảng số nguyên còn được gọi là mảng liên tục)

**- duyệt mảng**

**-for, foreach**

**03**

**Mảng**  kết hợp là mảng mà các chỉ số của các phần tử có thể là chuỗi hoặc số

**Chỉ dùng foreach để duyệt mảng  
04**

**Mảng lồng- mảng đa chiều** :

**Array\_sum** : hàm tính tổng trong mảng

**05**:

**Array\_value($array)** : lấy ra mảng các value

**Array\_keys($array)** : lấy ra mảng các key

**06**:

**Array\_pop($arr)**: loại bỏ phần tử cuối cùng của mảng. TRả về phần tử cuối cùng đó

**Array\_shift($arr)**: loại bỏ phần tử đầu tiên của mảng . Trả về phần tử đầu tiên đã được loại bỏ  
07:

**Array\_unique** ($arr) : loại bỏ các phần tử trùng nhau trong mảng

**08:   
unset($arr[0],$arr[1]) : xóa phần tử ở vị trí bất kỳ trong arr  
09 :**

**Array\_push($array, $val) thêm 1 hoặc nhiều phần tử vào cuối mảng**

**array\_unshift() : thêm 1 hoặc nhiều phần tử vào đầu mảng**

**10:**

**Array\_reverse($arr) : đảo ngược vị trí các phần tử mảng**

**11 :**

**Array\_flip($arr) : hoán đổi chỉ số và giá trị của mảng( đảo ngược key và $value)**

**12 :**

**Array\_sum($array): tính tổng các phần tử trong mảng**

**Min($arr) : xác định phần tử nhỏ nhât**

**Max($arr) : xác định phần tử lớn nhất**

**13 :**

**Array\_count\_value($arr) : thống kê các phần tử trong arr**

**14:**

**Array\_merge($array1, $array2) : nhập 2 hay nhiều mảng thành 1 mảng duy nhất**

**15 :**

**Array\_rand($arr,$number) : lấy ngẫu nhiên**

**16 :**

**Array\_search($value, $array) : tìm kiếm phần tử mang giá trị $value trong mảng $array. Trả về khóa của phần tử tìm được**

**17:**

**Array\_key\_exists($key, $arr) kiểm tra khóa $key có tồn tại trong mảng $arr hay ko? Nếu có trả về giá trị true.**

**In\_array($value, $arry) kiểm tra giá trị $value có tồn tại trogn amngr $array hay ko ? nếu có trả về giá trị true**

**18 :**

**Array\_change\_key\_case($course) : đưa key từ chữ hoa về chữ thường**

**Array\_change\_key\_case($course,Caseupper) : chữ hoa**

**19: chuyển đổi qua lại giữa mảng và chuỗi**

**20: truy xuất phần tử của mảng:**

**Current($array) truy xuất phần tử hiện tại của mảng**

**End($array) : truy xuất phần tử cuối cùng của mảng**

**Next($array): truy xuất phần tử sau phần tử hiện tại của mảng**

**Prev($array) truy xuất phần tử trước phần tử hiện tại của mảng**

**Reset() : quay về vị trí phần tử đầu tiên trong mảng**

**21:**

**Chuyển mảng về 1 chuỗi đặc biệt và ngược lại**

**Serialize($value) chuyển chuỗi/ mảng đối tượng thành 1 chuỗi đặc biệt để lưu vào CSDL**

**Unserialize($value) chuyển đổi chuỗi đặc biệt về trạng thái ban đầu**

**22:**

**Xáo trộn các phần tử trong arr**

**Sử dụng hàm shuffle để tạo ra mảng mới (mảng liên tục) với thứ tự các phần tử trong mảng bị thay đổi**

**23 : tạo mảng từ các biến đã có sẵn**

**Hàm compact()**

**24 : tạo mảng từ 1 khoảng nào đó**

**Range(0,10,2)**

**25 :array\_combine($key,$value):**

**Tạo 1 mảng mới có khóa được lấy từ magnr $key và giá trị được lấy từ mảng $value theo tuần tự**

**27: xử lý các   
array\_walk() : in mảng như foreach**

**28: array\_map : gửi các giá trị của một hay nhiều mảng đến 1 hàm nào đó để xử lý và nhận kết quả trả về là một mảng mới  
29 : trích xuất 1 đoạn phần tử của mảng**

**Bài tập 1 :**

**Bài tập 2 : trắc nghiệm tính cách:**

**Quetstion : chứa id và nội dung câu hỏi**

**Với mỗi câu hỏi sẽ có nhiều phương án trả lời => lưu vào bảng option**

**Result : sẽ lưu số điểm và kết quả tương tứng**

* **Vậy cần đọc các file này thôi**
* **Cần xây dựng 1 mảng chứa điểm các câu trả lời , danh sách các câu trả lời và dùng forech đổ ra**

Tham khảo:

thấy mọi người hầu hết đều thích dùng ajax nhỉ. Mình thấy như thế sẽ làm chậm đi khá nhiều. Bạn có thể thử cách sau:  
- Lấy toàn bộ danh sách dữ liệu rồi chuyển sang dạng json [{"tinh":"Hai Duong", "huyen": ["Ninh GIang", "Cam Giang"], {"tinh":"Ha Noi", "huyen": ["Dong Da", "Cau Giay"]}]  
- gán dữ liệu trên vào 1 biến javascript, tạm gọi là citydata đi  
- Tại sự kiện onchange của select tỉnh/thành, bạn dùng var city = citydata.filter({"tinh":"Ha Noi"}) để lọc ra tỉnh cần tìm, rồi lấy danh sách huyện city.huyen  
- Lấy được huyện rồi thì $("[#huyen\_selector](https://www.facebook.com/hashtag/huyen_selector?hc_location=ufi)").empty() để xóa dữ liệu cũ. Rồi dùng $.each(city.huyen, function() {//code append huyen o day}); để thêm huyện vào select