Mục lục

[TUẦN 1: PHP – link github: https://github.com/hungn2k/TT\_CleverGroup.git 3](#_Toc104733943)

[I. PHP Tutorial 3](#_Toc104733944)

[1. Syntax 3](#_Toc104733945)

[2. Comments 3](#_Toc104733946)

[3. Variables 3](#_Toc104733947)

[4. Echo/print 4](#_Toc104733948)

[5. Data Types 4](#_Toc104733949)

[6. String 5](#_Toc104733950)

[7. Numbers 5](#_Toc104733951)

[8. Math 5](#_Toc104733952)

[9. Constants 5](#_Toc104733953)

[10. Operator 5](#_Toc104733954)

[11. If…else….Elseif 10](#_Toc104733955)

[12. Switch 10](#_Toc104733956)

[13. Loops 10](#_Toc104733957)

[14. Functions 11](#_Toc104733958)

[15. Arrays 11](#_Toc104733959)

[16. Superglobals 11](#_Toc104733960)

[17. Regex 12](#_Toc104733961)

[II. PHP Form 12](#_Toc104733962)

[1. Form Handing 12](#_Toc104733963)

[2. Form Validation 13](#_Toc104733964)

[3. Form Required 13](#_Toc104733965)

[4. Form URL/E-mail 13](#_Toc104733966)

[5. Form Complete 14](#_Toc104733967)

[III. PHP Advanced 14](#_Toc104733968)

[1. Date and Time 14](#_Toc104733969)

[2. Inclue Files 14](#_Toc104733970)

[3. File Handling 15](#_Toc104733971)

[4. File Open/Read 15](#_Toc104733972)

[5. File Creat/Write 15](#_Toc104733973)

[6. File Upload 15](#_Toc104733974)

[7. Cookies 15](#_Toc104733975)

[8. Sessions 16](#_Toc104733976)

[9. Filters 16](#_Toc104733977)

[10. Callback Functions 16](#_Toc104733978)

[11. JSON 16](#_Toc104733979)

[12. Exceptions 16](#_Toc104733980)

[IV. PHP OOP 17](#_Toc104733981)

[1. Class/Objects 17](#_Toc104733982)

[2. Constructor 17](#_Toc104733983)

[3. Destructor 17](#_Toc104733984)

[4. Access Modifiers 17](#_Toc104733985)

[5. Inheritance 17](#_Toc104733986)

[6. Constants 17](#_Toc104733987)

[7. Abstract Classes 18](#_Toc104733988)

[8. Interfaces 18](#_Toc104733989)

[9. Traits 18](#_Toc104733990)

[10. Static Methods 18](#_Toc104733991)

[11. Static Properties 18](#_Toc104733992)

[12. Namespaces 19](#_Toc104733993)

[V. PHP AJAX 19](#_Toc104733994)

[1. Ajax php 19](#_Toc104733995)

[2. Ajax database 19](#_Toc104733996)

[3. Ajax xml 19](#_Toc104733997)

# TUẦN 1: PHP – link github: https://github.com/hungn2k/TT\_CleverGroup.git

## PHP Tutorial

### Syntax

* Phần mở rộng mặc định của các tệp PHP là “.php”
* Một tệp php thường chứa các thẻ HTML và một số mã kịch bản php

<?php  
// PHP code goes here  
?>

### Comments

* **Mục đích:** 
  + Để người khác hiểu code
  + Nhắc nhở bản thân về những gì đã làm
* **Chú thích 1 dòng:** 
  + // this is a single-line comment
  + # this is also a single-line comment
* **Chú thích nhiều dòng**:

/\*  
This is a multiple-lines comment block  
that spans over multiple  
lines  
\*/

### Variables

* **Tạo khai báo biến**: $txt = “hello”;
* **Biến PHP:**
  + Một biến bắt đầu với dấu $, tiếp theo là tên của biến
  + Một tên biến phải bắt đầu bằng một chữ cái hoặc ký tự gạch dưới
  + Tên biến không thể bắt đầu bằng một số
  + Một tên biến chỉ có thể chứa các ký tự và dấu gạch dưới chữ số alpha (A-z, 0-9 và \_)
  + $age và $AGE là hai biến khác nhau
* **Biến đầu ra:** 
  + echo: xuất dữ liệu lên màn hình
  + <?php  
    $x = 5;  
    $y = 4;  
    echo $x + $y;  
    ?>
* **Phạm vi biến:**
  + 3 phạm vi biến khác nhau:
    - **Local**: biến được khai báo trong một hàm có phạm vi cục bộ và chỉ có thể truy cập trong hàm đó
    - **Global**: biến đưuọc khai báo bên ngoài một hàm có phạm vi toàn cầu và chỉ có thể truy cập bên ngoài một hàm
    - **Static**
  + **Từ khóa global** được dùng để truy cập biến toàn cầu từ trong 1 hàm
  + PHP cũng lưu trữ tất cả các biến toàn cầu trong một mảng có tên $Globals [index]. Mảng này cũng có thể truy cập từ bên trong các chức năng và có thể được sử dụng để cập nhật các biến toàn cầu trực tiếp.
  + **Từ khóa static**: thông thường, khi một hàm được hoàn thành/thực thi, tất cả các biến của nó sẽ bị xóa. Tuy nhiên, đôi khi chúng ta muốn một biến cục bộ không bị xóa. Chúng ta cần nó cho một công việc tiếp theo🡪 sử dụng từ khóa tĩnh khi khai báo biến lần đầu tiên

function Test() {  
  static $x = 0;  
  echo $x;  
  $x++;  
}

🡪Sau đó, mỗi lần hàm được gọi, biến đó vẫn sẽ có thông tin mà nó chứa từ lần cuối cùng hàm được gọi

### Echo/print

* **Echo** và **Print** cơ bản là giống nhau. Cả hai đều được sử dụng để xuất dữ liệu vào màn hình. Sự khác biệt là: **Echo** không có giá trị trả về trong khi **Print** có giá trị trả về là 1 để nó có thể được sử dụng trong các biểu thức. **Echo** có thể lấy nhiều tham số (mặc dù việc sử dụng như vậy là rất hiếm) trong khi **Print** có thể lấy một đối số. **Echo** nhanh hơn một chút so với **Print**.

### Data Types

* String
* Integer:  -2,147,483,648 and 2,147,483,647
* Float:
* Boolean: True/ False
* Array
* Object
* Null value : Null là một loại dữ liệu đặc biệt chỉ có một giá trị: NULL. Một biến của loại dữ liệu null là một biến không có giá trị được gán cho nó. Nếu một biến được tạo mà không có giá trị, nó sẽ tự động được gán giá trị null. Các biến cũng có thể được làm trống bằng cách đặt giá trị thành NULL.

### String

* Hàm **strlen()** trả về độ dài của chuỗi
* Hàm **str\_word\_count ()** đếm số lượng từ trong một chuỗi
* Hàm **strrev()** đảo ngược một chuỗi
* Hàm **strpos ()** tìm kiếm một văn bản cụ thể trong một chuỗi. Nếu tìm thấy, hàm trả về vị trí ký tự đầu tiên. Nếu không tìm thấy, nó sẽ trả về FALSE
* Hàm **str\_replace ()** thay thế một số ký tự bằng một số ký tự khác trong một chuỗi

### Numbers

* **Integer:** là số không có phần thập phân. Kiểu dữ liệu số nguyên là số không thập phân giữa

-2147483648 và 2147483647 trong các hệ thống 32 bit và giữa -9223372036854775808 và 9223372036854775807 trong 64 hệ thống bit. Một giá trị lớn hơn (hoặc thấp hơn) so với điều này, sẽ được lưu trữ dưới dạng Float, bởi vì nó vượt quá giới hạn của một số nguyên.

* **Float:** Kiểu dữ liệu float thường có thể lưu trữ giá trị lên tới 1.7976931348623E+308 (phụ thuộc vào nền tảng) và có độ chính xác tối đa là 14 chữ số
* **Infinity**: vô cực
* **NaN:** là viết tắt của không phải là một số. NAN được sử dụng cho các hoạt động toán học không thể.

### Math

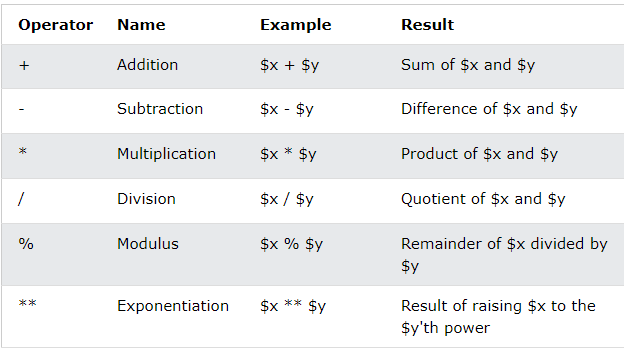
* Pi(), max(), min(), abs(), sqrt(), round(), rand(), …

### Constants

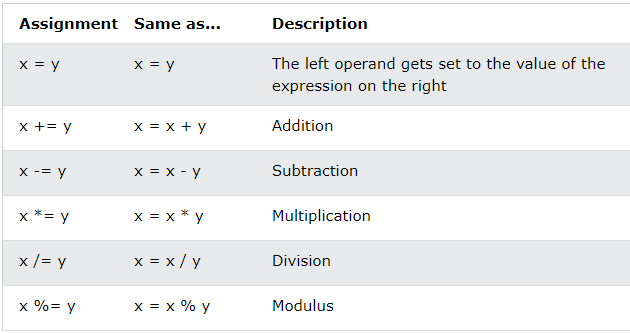
* là một định danh (tên) cho một giá trị đơn giản. Giá trị không thể được thay đổi trong tập lệnh. Một tên hằng số hợp lệ bắt đầu bằng chữ cái hoặc dấu gạch dưới (không có giá nào trước tên hằng số).
* define(*name*, *value*, *case-insensitive*)

### Operator

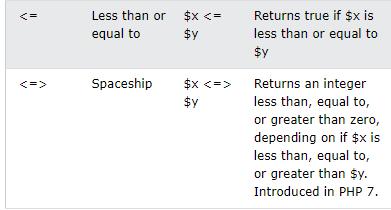
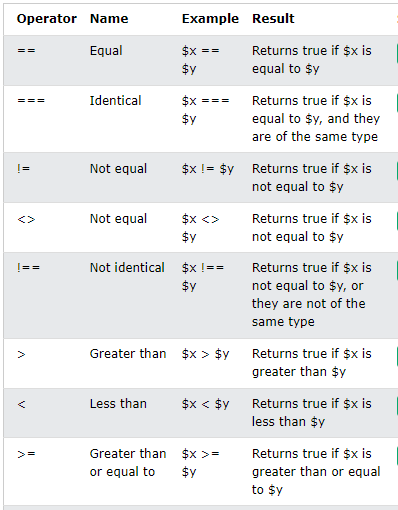
* Toán tử số học:



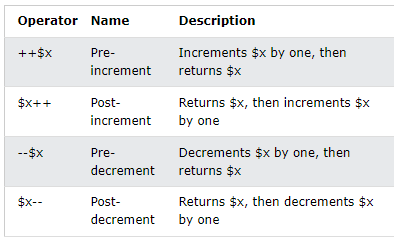
* Toán tử gán



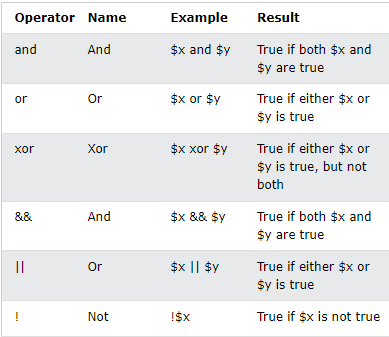
* Toán tử so sánh



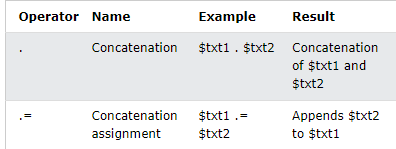
* Toán tử tăng/ giảm



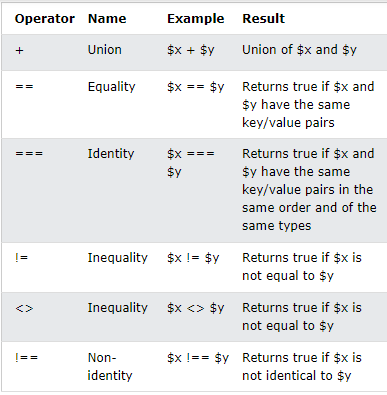
* Toán tử Logic:



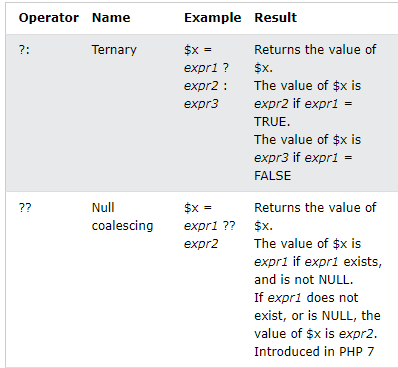
* Toán tử chuỗi



* Toán tử mảng



* Toán tử gán có điều kiện



### If…else….Elseif

* if (*condition*) {  
    *code to be executed if this condition is true;*} elseif (*condition*) {  
   *code to be executed if first condition is false and this condition is true;*} else {  
    *code to be executed if all conditions are false;*}

### Switch

* switch (*n*) {  
    case *label1:*  
    *code to be executed if n=label1;*  
      break;  
    case *label2:*  
    *code to be executed if n=label2;*  
      break;  
    case *label3:*  
    *code to be executed if n=label3;*  
      break;  
      ...  
    default:  
    *code to be executed if n is different from all labels;*  
  }

### Loops

* **While:**

while (*condition is true*) {  
*code to be executed*;  
}

* **Do while:**

do {  
*code to be executed;*} while (*condition is true*);

* **For:**

for (*init counter; test counter; increment counter*) {  
  *code to be executed for each iteration;*  
}

* **Foreach:**

foreach ($*array*as$*value*) {  
  *code to be executed;*  
}

* **Break/Continue**
  + **Break:** thoát khỏi vòng lặp
  + **Continue:** thoát khỏi lần lặp hiện tại, sang lần lặp mới

### Functions

function *functionName*() {  
*code to be executed*;  
}

### Arrays

* array();
* chỉ số bắt đầu là 0
* mảng liên kết: sử dụng các khóa có tên để gán cho chúng
* mảng đa chiều: chứa một hoặc nhiều mảng
* 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 liên kết theo thứ tự tăng dần, theo giá trị
  + ksort () - Sắp xếp các mảng liên kết theo thứ tự tăng dần, theo khóa
  + arsort( ) - Sắp xếp các mảng liên kết theo thứ tự giảm dần, theo giá trị
  + krsort () - Sắp xếp các mảng liên kết theo thứ tự giảm dần, theo khóa

### Superglobals

* Các biến có thể truy cập, bất kể phạm vi - và bạn có thể truy cập chúng từ bất kỳ chức năng, lớp hoặc tệp nào mà không phải làm bất cứ điều gì đặc biệt. Các biến Superglobal PHP là:
  + $GLOBALS
  + $\_SERVER
  + $\_REQUEST
  + $\_POST
  + $\_GET
  + $\_FILES
  + $\_ENV
  + $\_COOKIE
  + $\_SESSION
* $ Globals là một biến siêu toàn cầu PHP được sử dụng để truy cập các biến toàn cầu từ mọi nơi trong tập lệnh PHP
* $ \_Server là một biến siêu toàn cầu PHP, chứa thông tin về các tiêu đề, đường dẫn và vị trí tập lệnh.
* $ \_Request là một biến siêu toàn cầu PHP được sử dụng để thu thập dữ liệu sau khi gửi biểu mẫu HTML
* $ \_POST là một biến siêu toàn cầu PHP được sử dụng để thu thập dữ liệu biểu mẫu sau khi gửi biểu mẫu HTML với Phương thức = "Post". $ \_Post cũng được sử dụng rộng rãi để vượt qua các biến
* $ \_GET là một biến siêu toàn cầu PHP được sử dụng để thu thập dữ liệu biểu mẫu sau khi gửi biểu mẫu HTML với Phương thức = "GET". $ \_Get cũng có thể thu thập dữ liệu được gửi trong URL

### Regex

* Một biểu thức chính quy là một chuỗi các ký tự tạo thành một mẫu tìm kiếm. Khi bạn tìm kiếm dữ liệu trong một văn bản, bạn có thể sử dụng mẫu tìm kiếm này để mô tả những gì bạn đang tìm kiếm
* Preg\_match(): Trả về 1 nếu mẫu được tìm thấy trong chuỗi và 0 nếu không tìm thấy
* Preg\_match\_all (): Trả về số lần mẫu được tìm thấy trong chuỗi, cũng có thể là 0
* Preg\_repalce() : Trả về một chuỗi mới trong đó các mẫu phù hợp đã được thay thế bằng một chuỗi khác
* [abc]: Tìm một ký tự từ các tùy chọn giữa các dấu ngoặc
* [^abc]: Tìm bất kỳ ký tự nào không giữa các dấu ngoặc
* [0-9]: Tìm ký tự trong phạm vi 0-9
* |, . , $, \d, \s, \b ….

## PHP Form

### Form Handing

* Khi người dùng điền vào biểu mẫu ở trên và nhấp vào nút gửi, dữ liệu biểu mẫu sẽ được gửi để xử lý đến tệp PHP có tên "welcome.php". Dữ liệu biểu mẫu được gửi bằng phương pháp HTTP POST.

<html>  
<body>  
  
<form action="welcome.php" method="post">  
Name: <input type="text" name="name"><br>  
E-mail: <input type="text" name="email"><br>  
<input type="submit">  
</form>  
  
</body>  
</html>

* <html>  
  <body>  
    
  Welcome <?php echo $\_POST["name"]; ?><br>  
  Your email address is: <?php echo $\_POST["email"]; ?>  
    
  </body>  
  </html>

GET vs POST: Cả GET và POST đều tạo ra một mảng (ví dụ: mảng( key1 => value1, key2 => value2, key3 => value3,...)). Mảng này giữ các cặp khóa / giá trị, trong đó các phím là tên của các điều khiển biểu mẫu và giá trị là dữ liệu đầu vào từ người dùng.

Cả GET và POST đều được coi là $\_GET và $\_POST. Đây là những siêu toàn cầu, có nghĩa là chúng luôn có thể truy cập được, bất kể phạm vi - và bạn có thể truy cập chúng từ bất kỳ chức năng, lớp hoặc tệp nào mà không cần phải làm bất cứ điều gì đặc biệt.

$\_GET là một loạt các biến được chuyển đến tập lệnh hiện tại thông qua các tham số URL.

$ \_POST là một loạt các biến được chuyển sang kịch bản hiện tại thông qua phương pháp HTTP POST.

* GET: Thông tin được gửi từ biểu mẫu với phương pháp GET **hiển thị cho tất cả mọi người** (tất cả các tên và giá trị biến được hiển thị trong URL). GET cũng có giới hạn về số lượng thông tin cần gửi. Giới hạn là khoảng 2000 ký tự. Tuy nhiên, vì các biến được hiển thị trong URL, có thể đánh dấu trang. Điều này có thể hữu ích trong một số trường hợp
* POST: Thông tin được gửi từ biểu mẫu với phương pháp POST là **vô hình đối với người khác** (tất cả tên / giá trị được nhúng trong nội dung của yêu cầu HTTP) và **không có giới hạn** về số lượng thông tin cần gửi. Hơn nữa POST hỗ trợ chức năng nâng cao như hỗ trợ đầu vào nhị phân nhiều phần trong khi tải tệp lên máy chủ.

### Form Validation

* Biến $ \_Server ["php\_SELF"] là một Biến toàn cầu trả về Tên tệp của tập lệnh hiện Đang thực thi🡪 gửi dữ liệu biểu mẫu được gửi đến chính trang thay vì nhảy sang một trang khác🡪 người dùng sẽ nhậ được thông báo lỗi trên cùng một trang với biểu mẫu
* Htmlspecialchars(): chuyển đổi các ký tự đặc biệt thành các thực thể HTML

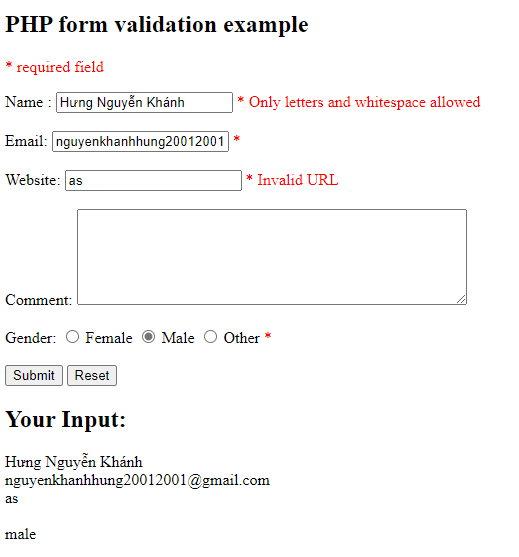
### Form Required

* Các trường bắt buộc: kiểm tra xem biến $ \_POST có trống không. Nếu nó trống, thông báo lỗi được lưu trữ trong các biến lỗi khác nhau và nếu nó không trống, nó sẽ gửi dữ liệu nhập liệu của người dùng
* Hiển thị lỗi:

### Form URL/E-mail

* Xác thực tên: kiểm tra xem trường tên chỉ chứa chữ cái, dấu gạch ngang, dấu nháy đơn và khoảng trắng. Nếu giá trị của trường tên không hợp lệ, thông báo lỗi
* Xác thực E-mail: filter\_var()
* Xác thực URL: preg\_match()

### Form Complete



## PHP Advanced

### Date and Time

* Date(): date(*format*,*timestamp*)
  + d - Đại diện cho ngày trong tháng (01 đến 31)
  + m - Đại diện cho một tháng (01 đến 12)
  + Y - Đại diện cho một năm (tính bằng bốn chữ số)
  + l (chữ thường 'L') - Đại diện cho ngày trong tuần
  + Các ký tự khác, như "/", ".", hoặc "-" cũng có thể được chèn vào giữa các ký tự để thêm định dạng bổ sung
* Get a Time: **echo "The time is " . date("h:i:sa");**
  + H - Định dạng 24 giờ của một giờ (00 đến 23)
  + h - định dạng 12 giờ của một giờ với số 0 hàng đầu (01 đến 12)
  + i - Phút với số 0 hàng đầu (00 đến 59)
  + s - Giây với số 0 hàng đầu (00 đến 59)
  + a - Lowercase Ante meridiem và Post meridiem (am or pm)
* get your Time Zone: date\_default\_timezone\_set(“America/New\_York”)
* tạo ngày: mktime()
* tạo ngày từ chuỗi : strtotime()

### Inclue Files

* require sẽ tạo ra một lỗi nghiêm trọng (e\_compile\_error) và dừng tập lệnh
* include sẽ chỉ tạo ra một cảnh báo (e\_warning) và tập lệnh sẽ tiếp tục

### File Handling

* Thao tác với Tệp:
  + Readfile(): đọc 1 tệp và ghi nó vào bộ đệm đầu ra

### File Open/Read

* Một phương thức tốt hơn để mở các tệp là với hàm fopen(). Hàm này cung cấp cho bạn nhiều tùy chọn hơn hàm readfile().
* Tham số đầu tiên của fopen () chứa tên của tệp sẽ được mở và tham số thứ hai chỉ định trong chế độ mà tệp sẽ được mở: fopen("name.txt", "r")
* Fread(): Tham số đầu tiên của fread () chứa tên của tệp để đọc và tham số thứ hai chỉ định số lượng byte tối đa để đọc: fread($myfile,filesize("name.txt"));
* Fclose(): đóng file
* Fgets(): đọc 1 dòng từ 1 tệp
* Feof(): check End-of-File
* Fgetc(): đọc 1 ký tự duy nhất từ 1 tệp

### File Creat/Write

* Hàm fopen () cũng được sử dụng để tạo một tệp: để viết (w) hoặc nối thêm (a)
* Hàm fwrite() được sử dụng để ghi vào tệp. Tham số đầu tiên của fwrite () chứa tên của tệp để ghi vào và tham số thứ hai là chuỗi được viết.
* Overwriting
* Nối văn bản: thêm dữ liệu vào tệp bằng cách sử dụng chế độ "a"

### File Upload

* Cấu hình tệp “php.ini”: file\_uploads = On
* Tạo biểu mẫu HTML:
  + method = post
  + enctype = “multipart/ form-data”
* Php script:
  + $target\_dir = "uploads/" - chỉ định thư mục nơi tệp sẽ được đặt
  + $target\_file chỉ định đường dẫn của tệp được tải lên
  + $uploadOk = 1
  + $imageFileType giữ phần mở rộng tệp của tệp (trong trường hợp thấp hơn)
* Kiểm tra xem tệp tồn tại chưa
* Giới hạn kích cỡ tệp
* Giới hạn loại tệp

### Cookies

* Cookie thường được sử dụng để xác định người dùng
* Tạo cookie với PHP: setcookie(name, value, expire, path, domain, secure, httponly); // chỉ cần tham số tên, cácc thông số khác là tùy chọn
* Tạo/truy xuất cookie: sử dụng biến toàn câu $\_COOKIE và dùng isset() để xem cookie có được đặt hay không
* Sửa đổi giá trị Cookie: setcookie()
* Xóa Cookie: để xóa cookie thì sử dụng hàm setcookie() có ngày hết hạn trong quá khứ

### Sessions

* một cách để lưu trữ thông tin (trong các biến) , thông tin được lưu trữ trên máy tính người dùng
* bắt đầu session: session\_start() , $\_SESSIOn
* lấy giá trị của biến session
* sửa đổi biến session: ghi đè lên nó
* hủy session: session\_unset() và session\_destroy()

### Filters

* Filter\_list() : xác thực dữ liệu (xem dữ liệu có ở dạng thích hợp k) và làm sạch dữ liệu ( loại bỏ các ký tự bất hợp pháp khỏi dữ liệu)
* Nhiều trang web nhận được đầu vào bên ngoài, dữ liệu được gửi không hợp lệ có thể dẫn đến các vấn đề bảo mất và phá vỡ trang web 🡪 bằng cách sử dụng Filter PHP: nhận được đầu vào chính xác
* Làm sạch chuỗi: filter\_var($str, FILTER\_SANITIZE\_STRING)
* Xác thực số nguyên: filter\_var($int, FILTER\_VALIDATE\_INT)
* Xác thực địa chỉ IP: filter\_var($ip, FILTER\_VALIDATE\_IP)
* Email: filter\_var($email, FILTER\_SANITIZE\_EMAIL),filter\_var($email, FILTER\_VALIDATE\_EMAIL
* URL: filter\_var($url, FILTER\_SANITIZE\_URL); filter\_var($url, FILTER\_VALIDATE\_URL)

### Callback Functions

* là một hàm được chuyển như một đối số vào một hàm khác.

### JSON

* một cú pháp để lưu trữ và trao đổi dữ liệu.
* json\_encode(): mã hóa giá trị
* json\_decode(): giải mã 1 đối tượng json thành 1 đói tượng php hoặc 1 mảng liên kết
* truy cập các giá trị đã giải mã

### Exceptions

* Một ngoại lệ là một đối tượng mô tả một lỗi hoặc hành vi bất ngờ của một tập lệnh PHP
* Try…catch:

try {  
  code that can throw exceptions  
} catch(Exception $e) {  
  code that runs when an exception is caught  
} finally {  
  code that always runs regardless of whether an exception was caught  
}

* Đối tượng ngoại lệ: chứa thông tin về lỗi hoặc hành vi bất ngờ mà chức năng gặp phải

new Exception(message, code, previous)

## PHP OOP

### Class/Objects

* Lập trình hướng đối tượng có một số lợi thế so với lập trình thủ tục:
  + OOP nhanh hơn và dễ thực hiện hơn
  + OOP cung cấp một cấu trúc rõ ràng cho các chương trình
  + OOP giúp giữ cho mã PHP KHÔ "Đừng lặp lại chính mình", và làm cho mã dễ dàng hơn để duy trì, sửa đổi và gỡ lỗi
  + OOP cho phép tạo ra các ứng dụng có thể tái sử dụng đầy đủ với ít mã hơn và thời gian phát triển ngắn hơn
* Class
* Đối tượng: khởi tạo bằng từ khóa new
* $this: đề cập đến đối tượng hiện tại
* Instanceof: kiểm tra xem 1 đối tượng có thuộc 1 lớp cụ thể hay không

### Constructor

* \_\_**constructor**: cho phép bạn khởi tạo các thuộc tính của một đối tượng khi tạo ra đối tượng

### Destructor

* \_\_**destruct**: được gọi khi đối tượng bị hủy hoặc tập lệnh bị dừng hoặc thoát ra.

### Access Modifiers

* **public** - thuộc tính hoặc phương thức có thể được truy cập từ khắp mọi nơi. Đây là mặc định
* **protected** - thuộc tính hoặc phương thức có thể được truy cập trong lớp và theo các lớp có nguồn gốc từ lớp đó
* **private** - thuộc tính hoặc phương thức chỉ có thể được truy cập trong lớp

### Inheritance

* Kế thừa trong OOP = Khi một lớp xuất phát từ một lớp khác.
* Từ khóa **extends**
* **Overriding**: Các phương pháp kế thừa có thể được ghi đè bằng cách xác định lại các phương pháp (sử dụng cùng tên) trong lớp con
* **Final**: được sử dụng để ngăn chặn kế thừa lớp hoặc để ngăn chặn ghi đè phương pháp

### Constants

* Hằng số không thể thay đổi một khi nó được khai báo
* Từ khóa **const**
* Chúng ta có thể truy cập một hằng số từ bên trong lớp bằng cách sử dụng từ khóa **self**, theo sau là toán tử phân giải phạm vi **(: :)**

### Abstract Classes

* Các lớp và phương pháp trừu tượng là khi lớp cha mẹ có một phương pháp được đặt tên, nhưng cần lớp con của nó để điền vào các nhiệm vụ
* Từ khóa **abstract**()
* Phương pháp lớp con phải được xác định với cùng tên và nó khai báo lại phương pháp trừu tượng của cha mẹ
* Phương pháp lớp con phải được xác định với cùng một hoặc sửa đổi truy cập ít bị hạn chế hơn
* Số lượng các đối số cần thiết phải giống nhau. Tuy nhiên, lớp con có thể có các đối số tùy chọn ngoài

### Interfaces

* Giao diện cho phép bạn chỉ định những phương pháp mà một lớp nên thực hiện.
* Từ khóa: **interface**
* Giao diện tương tự như các lớp trừu tượng. Sự khác biệt giữa giao diện và các lớp trừu tượng là:
* Giao diện không thể có thuộc tính, trong khi các lớp trừu tượng có thể
* Tất cả các phương pháp giao diện phải được công khai, trong khi các phương pháp lớp trừu tượng là công khai hoặc được bảo vệ
* Tất cả các phương pháp trong một giao diện là trừu tượng, vì vậy chúng không thể được thực hiện trong mã và từ khóa trừu tượng là không cần thiết.
* Các lớp có thể thực hiện một giao diện trong khi kế thừa từ một lớp khác cùng một lúc
* Sử dụng giao diện: **implements**

### Traits

* PHP chỉ hỗ trợ thừa kế duy nhất: một lớp con chỉ có thể thừa kế từ một phụ huynh duy nhất.
* Nếu một lớp cần thừa hưởng nhiều hành vi 🡪 **trait**
* Sử dụng 1 đặc điểm trong 1 lớp 🡪 **use**
* Sử dụng nhiều đặc điểm

### Static Methods

* Các phương pháp tĩnh có thể được gọi trực tiếp - mà không tạo ra một phiên bản của lớp trước
* Từ khóa **static**
* Để truy nhập phương pháp tĩnh, hãy dùng tên lớp, dấu hai (::) và tên phương thức : ClassName::staticMethod();
* Một lớp có thể có cả phương pháp tĩnh và không tĩnh. Một phương pháp tĩnh có thể được truy cập từ một phương pháp trong cùng một lớp bằng cách sử dụng từ khóa **self** và **(::)**
* Các phương pháp tĩnh cũng có thể được gọi từ các phương pháp trong các lớp khác 🡪 phải là public

### Static Properties

* Thuộc tính tĩnh có thể được gọi trực tiếp - mà không tạo ra một phiên bản của một lớp.
* Từ khóa **static**
* Để truy nhập thuộc tính tĩnh, hãy dùng tên lớp, dấu hai (::) và tên thuộc tính: ClassName::$staticProp;
* Để gọi thuộc tính tĩnh từ lớp con, hãy dùng từ khóa **parent** bên trong lớp con

### Namespaces

* Khai báo: namespace; Cú pháp: namespace Code\Html;
* Sử dụng:
  + Dùng các lớp từ không gian tên Html: $table = new Html\Table()  
     $row = new Html\Row();
  + Sử dụng các lớp từ không gian tên html mà không cần Html\qualifier

namespace Html;  
$table = new Table();  
$row = new Row();

* Namespace alias:

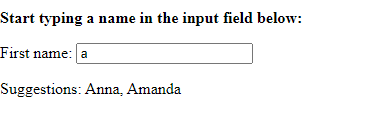
use Html as H;  
$table = new H\Table();

## PHP AJAX

* Là một kỹ thuật để tạo ra các trang web nhanh và năng động. AJAX cho phép các trang web được cập nhật không đồng bộ bằng cách trao đổi một lượng nhỏ dữ liệu với máy chủ đằng sau hậu trường. Điều này có nghĩa là có thể cập nhật các phần của trang web mà không cần tải lại toàn bộ trang. Các trang web cổ điển, (không sử dụng AJAX) phải tải lại toàn bộ trang nếu nội dung thay đổi.
* AJAX dựa trên các tiêu chuẩn internet và sử dụng kết hợp:
  + Đối tượng XMLHttpRequest (để trao đổi dữ liệu không đồng bộ với máy chủ)
  + JavaScript/DOM (để hiển thị/tương tác với thông tin)
  + CSS (để tạo kiểu cho dữ liệu)
  + XML (thường được sử dụng làm định dạng để truyền dữ liệu)

### Ajax php

* Một trang web có thể giao tiếp với máy chủ web trong khi người dùng nhập ký tự trong trường nhập liệu



### Ajax live Search

* Bạn nhận được kết quả tìm kiếm trong khi nhập. Tìm kiếm trực tiếp có nhiều lợi ích so với tìm kiếm truyền thống: Kết quả được hiển thị khi bạn nhập