**Đề tài: xây dựng website quảng bá du lịch Việt Nam**

**Nhiệm vụ : tìm hiểu về ngôn ngữ lập trình PHP**

**và thiết kế bố cục trang web**

*sinh viên thực hiện: Nguyễn Văn Tạo*

Phụ lục

[I. Giới thiệu về ngôn ngữ lập trình PHP 2](#_Toc477985880)

[1. PHP hoạt động như thế nào? 2](#_Toc477985881)

[2. Cú pháp 3](#_Toc477985882)

[3. Kiểu dữ liệu 4](#_Toc477985883)

[II. Thiết kế bố cục trang web 4](#_Toc477985884)

[1. Viết code cho header.php 4](#_Toc477985885)

[2. Viết code cho footer.php 5](#_Toc477985886)

[3. Viết code cho functions.php 6](#_Toc477985887)

[- Thiết lập các hằng dữ liệu quan trọng 6](#_Toc477985888)

[- Thiết lập chiều rộng nội dung 6](#_Toc477985889)

[- Hàm thiết lập chức năng của theme 7](#_Toc477985890)

[- Tạo menu location 7](#_Toc477985891)

[- Tạo sidebar 7](#_Toc477985892)

[4. Viết code cho content.php 8](#_Toc477985893)

# Giới thiệu về ngôn ngữ lập trình PHP

**PHP** - viết tắt hồi quy của **"Hypertext Preprocessor"**, là một ngôn ngữ lập trình kịch bản được chạy ở phía server nhằm sinh ra mã html trên client. PHP đã trải qua rất nhiều phiên bản và được tối ưu hóa cho các ứng dụng web, với cách viết mã rõ rãng, tốc độ nhanh, dễ học nên PHP đã trở thành một ngôn ngữ lập trình web rất phổ biến và được ưa chuộng.

 PHP chạy trên môi trường Webserver và lưu trữ dữ liệu thông qua hệ quản trị cơ sở dữ liệu nên PHP thường đi kèm với Apache, MySQL và hệ điều hành Linux (LAMP).

* Apache là một phần mềm web server có nhiệm vụ tiếp nhận request từ trình duyệt người dùng sau đó chuyển giao cho PHP xử lý và gửi trả lại cho trình duyệt.
* MySQL cũng tương tự như các hệ quản trị cơ sở dữ liệu khác (Postgress, Oracle, SQL server...) đóng vai trò là nơi lưu trữ và truy vấn dữ liệu.
* Linux: Hệ điều hành mã nguồn mở được sử dụng rất rộng rãi cho các webserver. Thông thường các phiên bản được sử dụng nhiều nhất là RedHat Enterprise Linux, Ubuntu...

## PHP hoạt động như thế nào?

Khi người sử dụng gọi trang PHP, Web Server sẽ triệu gọi PHP Engine để thông dịch dịch trang PHP và trả kết quả cho người dùng như hình bên dưới.



## Cú pháp

PHP chỉ phân tích các đoạn mã nằm trong những dấu giới hạn của nó. Bất cứ mã nào nằm ngoài những dấu giới hạn đều được xuất ra trực tiếp không thông qua xử lý bởi PHP. Các dấu giới hạn thường dùng nhất là <?php và ?>, tương ứng với dấu giới hạn mở và đóng. Các dấu giới hạn <script language="php"> và </script> cũng đôi khi được sử dụng. Cách viết dấu giới hạn dạng thẻ ngắn cũng có thể được dùng để thông báo bắt đầu đoạn mã PHP, là <? hay <?= (dấu này được sử dụng để in ra (echo) các xâu kí tự hay biến) với thẻ thông báo kết thúc đoạn mã PHP là ?>. Những thẻ này thường xuyên được sử dụng, tuy nhiên giống với những thẻ kiểu ASP (<%hay <%= và %>), chúng không có tính di động cao bởi có thể bị vô hiệu khi cấu hình PHP. Bởi vậy, việc dùng các thẻ dạng ngăn hay các thẻ kiểu ASP không được khuyến khích. Mục đích của những dấu giới hạn này là ngăn cách mã PHP với những đoạn mã thuộc ngôn ngữ khác, gồm cả HTML. Mọi đoạn mã bên ngoài các dấu này đều bị hệ thống phân tích bỏ qua và được xuất ra một cách trực tiếp.

Các biến được xác định bằng cách thêm vào trước một dấu đô la (*$*) và không cần xác định trước kiểu dữ liệu. Không giống với tên hàm và lớp, tên biến là trường hợp nhạy cảm. Cả dấu ngoặc kép ("") và ký hiệu đánh dấu văn bản (<<<EOF EOF;) đều có thể dùng để truyền xâu và giá trị biến. PHP coi xuống dòng  như một khoảng trắng theo kiểu như một ngôn ngữ dạng tự do (*free-form language*) (trừ khi nó nằm trong trích dẫn xâu), và các phát biểu được kết thúc bởi một dấu chấm phẩy. PHP có ba kiểu [cú](https://vi.wikipedia.org/w/index.php?title=Ch%C3%BA_th%C3%ADch_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)&action=edit&redlink=1) pháp chú thích: /\* \*/ cho phép một đoạn chú thích tùy ý, trong khi đó // và # cho phép chú thích trong phạm vi một dòng. Phát biểu echo là một trong những lệnh của PHP cho phép xuất văn bản (vd. ra một trình duyệt web).

Về cú pháp các từ khóa và ngôn ngữ, PHP tương tự hầu hết các ngôn ngữ lập trình bậc cao có cú pháp kiểu C. Các phát biểu điều kiện *If* (*Nếu*), vòng lặp *for* và *while*, các hàm trả về đều tương tự cú pháp của các ngôn ngữ như C, C++ , Java và Perl.

## Kiểu dữ liệu

* + Kiểu số nguyên(int), chuỗi ký tự(string),kiểu số thực(float, double),...
  + Kiểu class, kiểu enum
  + Các hàm cơ bản:
* [echo()](http://php.net/manual/en/function.echo.php) : được dùng để in một dữ liệu chuỗi nào đó ra màn hình.
* [md5()](http://php.net/manual/en/function.md5.php):tự động chuyển đổi một chuỗi nào đó sang dạng mã hoá MD5.
* [str\_split()](http://php.net/manual/en/function.str-split.php):để cắt các ký tự trong một chuỗi và chuyển nó thành dạng mảng.
* [str\_word\_count()](http://php.net/manual/en/function.str-word-count.php) : đếm có bao nhiêu từ trong một chuỗi
* [strlen()](http://php.net/manual/en/function.strlen.php):đếm độ dài của một chuỗi, nghĩa là nó có bao nhiêu ký tự bao gồm các ký tự khoảng trắng và ký tự đặc biệt
* [substr()](http://php.net/manual/en/function.substr.php):để tách một phần trong một chuỗi
* [str\_replace()](http://php.net/manual/en/function.str-replace.php) :  tìm một từ khoá nào đó và thay thế thành từ khoá của bạn, nó có thể làm việc trong một chuỗi thông thường hoặc một chuỗi trong mảng
* [date()](http://php.net/manual/en/function.date.php):hiển thị ngày tháng hiện tại hay ở thời gian nào đó
* [phpinfo()](http://php.net/manual/en/function.phpinfo.php):xem chi tiết các thông số cấu hình PHP trên máy chủ của bạn

# Thiết kế bố cục trang web

## Viết code cho header.php

Ý nghĩa của header.php

Tại file này, chúng ta sẽ khai báo các thẻ HTML cần thiết mà trong một tài liệu HTML chuẩn đều có như <html>, <head>, <body>,….và nhất là phần thẻ <head> là phần quan trọng nhất. Ngoài ra trong dự án này, chúng ta sẽ viết thêm code hiển thị tên website và menu để vào file header.php này vì các thành phần đó chúng ta đều muốn nó hiển thị trong tất cả mọi trang.

<!DOCTYPE html>

<!--[if IE 8]> <html <?php language\_attributes(); ?> class="ie8"> <![endif]-->

<!--[if !IE]> <html <?php language\_attributes(); ?>> <![endif]-->

<head>

        <meta charset="<?php bloginfo( 'charset' ); ?>" />

        <link rel="profile" href="http://gmgp.org/xfn/11" />

        <link rel="pingback" href="<?php bloginfo( 'pingback\_url' ); ?>" />

        <?php wp\_head(); ?>

</head>

<body <?php body\_class(); ?> > <!--Thêm class tượng trưng cho mỗi trang lên <body> để tùy biến-->

        <div id="container">

              <header id="header">

                   <?php CNT\_logo(); ?>

                   <?php CNT\_menu( 'primary-menu' ); ?>

              </header>

Trong đó, bạn nên lưu ý nhất là cái hook wp\_head(), đây là cái hook để giúp WordPress hiểu được đây là khu vực thẻ <head> của theme bạn để nó có thể tự thêm các thành phần cần thiết lên, cũng như các plugin khác có muốn can thiệp vào khu vực này thì cũng sẽ dò qua hook wp\_head().

## Viết code cho footer.php

Cũng giống như header.php, footer.php là file mà sau này chúng ta sẽ làm cho nó hiển thị ở mọi trang.

<footer id="footer">

                <div class="copyright">

                        © <?php echo date('Y'); ?> <?php bloginfo( 'sitename' ); ?>. <?php \_e('All rights reserved', 'CNT'); ?>. <?php \_e('This website is proundly to use WordPress', 'CNT'); ?>

                </div>

        </footer>

        </div> <!--end #container -->

        <?php wp\_footer(); ?>

</body> <!--end body-->

</html> <!--end html -->

Cũng giống như wp\_head(), hook wp\_footer() là giúp cho WordPress nhận ra đâu là phần footer của trang web để một số plugin có thể hook vào và làm được những việc nó muốn.

## Viết code cho functions.php

Một trong những file quan trọng nhất của một theme WordPress đó là file functions.php. Đây là một file bắt buộc trong theme và nó sẽ chứa các đoạn code nguồn mà bạn muốn nó luôn được load mỗi khi tải website.

### Thiết lập các hằng dữ liệu quan trọng

\*\*

  @ Thiết lập các hằng dữ liệu quan trọng

  @ THEME\_URL = get\_stylesheet\_directory() - đường dẫn tới thư mục theme

  @ CORE = thư mục /core của theme, chứa các file nguồn quan trọng.

  \*\*/

  define( 'THEME\_URL', get\_stylesheet\_directory() );

  define( 'CORE', THEME\_URL . '/core' );

### Thiết lập chiều rộng nội dung

/\*\*

 @ Thiết lập $content\_width để khai báo kích thước chiều rộng của nội dung

 \*\*/

 if ( ! isset( $content\_width ) ) {

       /\*

        \* Nếu biến $content\_width chưa có dữ liệu thì gán giá trị cho nó

        \*/

       $content\_width = 620;

  }

### Hàm thiết lập chức năng của theme

/\*\*

  @ Thiết lập các chức năng sẽ được theme hỗ trợ

  \*\*/

  if ( ! function\_exists( 'CNT\_theme\_setup' ) ) {

        /\*

         \* Nếu chưa có hàm CNT\_theme\_setup() thì sẽ tạo mới hàm đó

         \*/

        function CNT\_theme\_setup() {

        }

        add\_action ( 'init', 'CNT\_theme\_setup' );

  }

### Tạo menu location

/\*

\* Tạo menu cho theme

\*/

register\_nav\_menu ( 'primary-menu', \_\_('Primary Menu', 'CNT') );

### Tạo sidebar

/\*

\*tao sidebar cho theme

\*/

$sidebar = array(

   'name' => \_\_('Main Sidebar', 'CNT'),

   'id' => 'main-sidebar',

   'description' => 'Main sidebar for CNT theme',

   'class' => 'main-sidebar',

   'before\_title' => '<h3 class="widgettitle">',

   'after\_title' => '</h3>'

);

register\_sidebar( $sidebar );

## Viết code cho content.php

<article id="post-<?php the\_ID(); ?>" <?php post\_class(); ?>>

        <div class="entry-thumbnail">

        </div>

        <header class="entry-header">

        </header>

        <div class="entry-content">

        </div>

</article>

Nghĩa là trong phần đó, chúng ta cho mỗi post hiển thị ra đều nằm trong thẻ <article> với ID là post-$id-của-post cùng với các class tượng trưng cho nó.

Tiếp đó, khu vực entry-thumbnail mình sẽ dùng để hiển thị ảnh đại diện của post. Phần .entry-header. sẽ hiển thị tên post và thông tin của post, phần.entry-content là hiển thị nội dung của post.