

PHIẾU HỌC TẬP CHỦ ĐỘNG (PHT)

Môn học: CSE485: Công nghệ Web

Họ tên: Nguyễn Văn Trường

Mã sinh viên: 2351170625

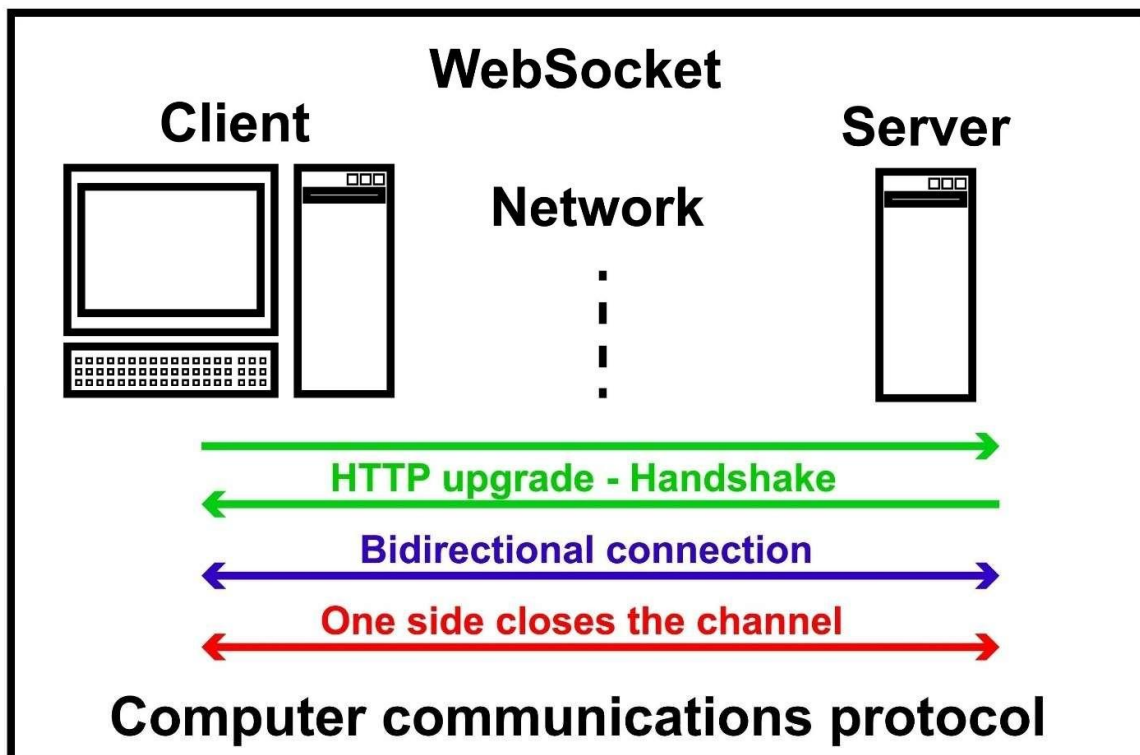
Lớp: 65KTPM

CHƯƠNG 1: TÌM HIỂU VỀ CÔNG NGHỆ WEB ⁴

1. Lý thuyết Cốt lõi (Khái niệm)

Chương 1 giới thiệu về cấu trúc và nguyên tắc làm việc của WWW⁵. Khái niệm cốt lõi bạn phải nắm là **Mô hình Client-Server** và **luồng hoạt động của một yêu cầu HTTP**:

1. **Web Client (Trình duyệt)**: Máy của bạn (Chrome, Firefox).
2. **Web Server**: Một máy tính "ở xa" chứa mã nguồn (PHP, HTML...) của trang web.
3. **URL (Địa chỉ)**: Định danh duy nhất của tài nguyên (ví dụ: <https://www.tlu.edu.vn/>).
4. **DNS (Hệ thống tên miền)**: Giống như "danh bạ điện thoại" của Internet, phân giải tên miền (như [tlu.edu.vn](https://www.tlu.edu.vn/)) thành một địa chỉ IP (như 103.1.238.194) mà máy tính có thể hiểu⁶.
5. **Giao thức HTTP**: Ngôn ngữ (quy tắc) giao tiếp giữa Client và Server⁷.



Shutterstock

Luồng hoạt động chuẩn:

Bạn gõ URL \rightarrow Trình duyệt (Client) hỏi DNS \rightarrow DNS trả về IP \rightarrow Client gửi Yêu cầu HTTP (HTTP Request) đến IP đó \rightarrow Web

Server nhận yêu cầu, xử lý, và gửi lại Phản hồi HTTP (HTTP Response) \rightarrow Client nhận Response (thường là HTML) và hiển thị cho bạn.

2. Nhiệm vụ Thực hành (BẮT BUỘC)

Chương này không code PHP, nhưng chúng ta sẽ "thực hành" với công cụ quan trọng nhất của lập trình viên web: **Developer Tools (F12)**.

Kịch bản: Bạn sẽ "khám nghiệm" (autopsy) chính trang web của Trường Đại học Thủy Lợi để xem Client và Server "nói chuyện" với nhau như thế nào, dựa trên lý thuyết HTTP bạn vừa học⁹.

"Code" Khởi đầu (Starter "Code"):

Không có code. "Starter" của bạn là URL: <https://www.tlu.edu.vn/>

Các bước (Tương đương // TODO):

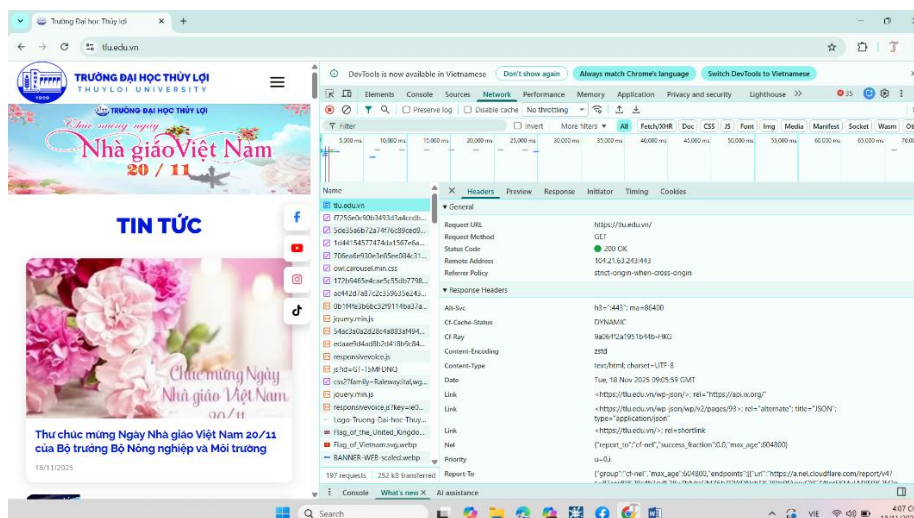
1. // **TODO 1:** Mở trình duyệt Google Chrome (hoặc Firefox) và truy cập trang <https://www.tlu.edu.vn/>.
2. // **TODO 2:** Mở Developer Tools (Nhấn phím **F12**).
3. // **TODO 3:** Chuyển sang tab "**Network**" (Mạng).
4. // **TODO 4:** Tải lại trang (Nhấn **F5** hoặc nút Refresh) để xem các yêu cầu mạng (network requests) xuất hiện.
5. // **TODO 5:** Tìm và nhấp vào yêu cầu (request) **đầu tiên** trong danh sách (thường là www.tlu.edu.vn).
6. // **TODO 6:** Quan sát cửa sổ "Headers" vừa xuất hiện.

3.Yêu cầu Bằng chứng (Proof of Work)

Ban phải nộp lại 2 bằng chứng sau:

A. Ảnh chụp màn hình Kết quả (Trình duyệt Web):

Chụp ảnh màn hình tab "Network" của bạn, sau khi đã nhấp vào yêu cầu www.tlu.edu.vn (Phải thấy rõ cửa sổ "Headers" như hình minh họa).



B. Thông tin trích xuất:

Nhìn vào cửa sổ "Headers" (Phần 2, TODO 6), tìm và chép lại chính xác 3 thông tin sau:

1. **Request URL:** `https://tlu.edu.vn/`
2. **Request Method:** `GET`
3. **Status Code:** `200 OK`

4. Câu hỏi Phản biện (Bắt buộc)

Sau khi hoàn thành Phần 2 & 3, hãy đặt 01 câu hỏi tư duy về những gì bạn vừa quan sát.

(Gợi ý: "Tại sao khi tải 1 trang web, tab Network lại hiển thị hàng chục yêu cầu (requests) khác nhau (như file .css, .js, .jpg) thay vì chỉ 1 yêu cầu duy nhất đến `www.tlu.edu.vn`?").

Câu hỏi của tôi là: *Khi kiểm tra các yêu cầu mạng, tại sao phần lớn các tài nguyên phụ (như tập tin CSS, JS, hoặc hình ảnh) lại có Status Code là 200 OK, trong khi một số tài nguyên khác có thể xuất hiện với Status Code là 304 Not Modified? Điều này nói lên điều gì về vai trò của bộ nhớ đệm (caching) trong việc tối ưu hóa hiệu suất tải trang web?*

5. Kết nối Đánh giá (Rất quan trọng)

Kỹ năng "Đọc Tab Network" bạn vừa thực hành là **kỹ năng gỡ lỗi (debug) số 1** của lập trình viên web.

Khi làm **Bài tập PHP (20%)** ¹⁰ hoặc **Bài tập lớn (50%)** ¹¹ sau này, nếu ứng dụng của bạn báo lỗi (ví dụ: Lỗi 500, 404, hoặc dữ liệu không hiển thị), tab "Network" là nơi đầu tiên bạn phải kiểm tra để xem Web Server (PHP/Laravel) đã thực sự trả về lỗi gì.

CHƯƠNG 2: LẬP TRÌNH PHP CĂN BẢN

1. Lý thuyết Cốt lõi (Khái niệm)

Chương này là nền móng. PHP là ngôn ngữ chạy phía **Server**. Bạn viết code, server thi hành, và chỉ trả về kết quả (thường là HTML) cho trình duyệt. Trình duyệt **không bao giờ** thấy code PHP của bạn.

Các cú pháp cốt lõi bạn phải nhớ:

- **Khối lệnh PHP:** Mọi code PHP phải nằm trong `<?php ... ?>`.
- **Biến (2.1):** Luôn bắt đầu bằng dấu đô-la (\$). Ví dụ: `$ten = "Tuấn";`, `$tuoi = 20;`. Biến PHP không cần khai báo `let` hay `var` như JavaScript.
- **In ra màn hình:** Dùng `echo` hoặc `print`. Ví dụ: `echo "Chào bạn $ten";`.
- **Nối chuỗi:** Dùng dấu chấm (`.`), **không phải** dấu cộng (`+`) như JavaScript. Ví dụ: `echo "Tuổi của bạn là: " . $tuoi;`.
- **Cấu trúc điều khiển (2.2):** Cú pháp `if...else`, `for`, `while`, `switch` giống hệt C, Java, và JavaScript.
- **Hàm (2.3):** Dùng từ khóa `function`. Ví dụ: `function tinhTong($a, $b) { return $a + $b; }`.
- **Kết thúc dòng:** Mọi câu lệnh PHP phải kết thúc bằng dấu chấm phẩy (`;`).

2. Nhiệm vụ Thực hành (BẮT BUỘC)

Kịch bản: Bạn sẽ tạo một tệp `sinhvien.php`. Tệp này sẽ khai báo thông tin cơ bản của một sinh viên, sau đó dùng logic `if/else` để xếp loại và dùng vòng lặp `for` để in ra một thông điệp.

Code Khởi đầu (Starter Code):

Tạo 1 tệp `chapter2.php` (hoặc tên bất kỳ) trong thư mục `htdocs` của XAMPP và dán code sau:

PHP

```
<!DOCTYPE html>

<html lang="vi">

<head>

    <meta charset="UTF-8">

    <title>PHT Chương 2 - PHP Căn Bản</title>

</head>

<body>

    <h1>Kết quả PHP Căn Bản</h1>

    <?php

// BẮT ĐẦU CODE PHP CỦA BẠN TẠI ĐÂY
```

// TODO 1: Khai báo 3 biến

// \$ho_ten = "Nguyễn Văn A"; (Thay bằng tên của bạn)

// \$diem_tb = 7.5; (Thay bằng điểm bạn muốn)

// \$co_di_hoc_chuyen_can = true; (hoặc false)

// TODO 2: In ra thông tin sinh viên

// Dùng lệnh echo để in ra: "Họ tên: \$ho_ten", "Điểm: \$diem_tb"

// (Lưu ý: Phải in ra cả thẻ
 để xuống dòng trong HTML)

// TODO 3: Viết cấu trúc IF/ELSE IF/ELSE (2.2)

// Dựa vào \$diem_tb, in ra xếp loại:

// - Nếu \$diem_tb >= 8.5 VÀ \$co_di_hoc_chuyen_can == true => "Xếp loại: Giỏi"

// - Ngược lại, nếu \$diem_tb >= 6.5 VÀ \$co_di_hoc_chuyen_can == true => "Xếp loại: Khá"

// - Ngược lại, nếu \$diem_tb >= 5.0 VÀ \$co_di_hoc_chuyen_can == true => "Xếp loại: Trung bình"

// - Các trường hợp còn lại (bao gồm cả \$co_di_hoc_chuyen_can == false) => "Xếp loại: Yếu (Cần cố gắng thêm!)" //

Gợi ý: Dùng toán tử && (AND)

// TODO 4: Viết 1 hàm đơn giản (2.3)

// Tên hàm: chaoMung()

// Hàm này không có tham số, chỉ cần `echo "Chúc mừng bạn đã hoàn thành PHT Chương 2!"`

// TODO 5: Gọi hàm bạn vừa tạo

// Gợi ý: Gõ tên hàm và dấu ();

// KẾT THÚC CODE PHP CỦA BẠN TẠI ĐÂY

?>

```
</body>
```

```
</html>
```

3. Yêu cầu Bằng chứng (Proof of Work)

Bạn phải nộp lại 2 bằng chứng sau:

A. Code đã hoàn thiện: Dán (paste) toàn bộ code của tệp chapter2.php mà bạn đã hoàn thiện.

```
<!DOCTYPE html>
<html lang="vi">

<head>
    <meta charset="UTF-8">
    <title>PHT Chương 2 - PHP Căn Bản</title>
</head>

<body>
    <h1>Kết quả PHP Căn Bản</h1>
    <?php
        // BẮT ĐẦU CODE PHP CỦA BẠN TẠI ĐÂY
        // TODO 1: Khai báo 3 biến
        $ho_ten = "Nguyễn Văn Trường";
        $diem_tb = 9;
        $co_di_hoc_chuyen_can = true;

        // TODO 2: In ra thông tin sinh viên
        echo "Họ tên: $ho_ten <br>";
        echo "Điểm trung bình: $diem_tb <br>";

        // TODO 3: Viết cấu trúc IF/ELSE IF/ELSE (2.2)
        if ($diem_tb >= 8.5 && $co_di_hoc_chuyen_can == true) {
            echo "Xếp loại: Giỏi <br>";
        } elseif ($diem_tb >= 6.5 && $co_di_hoc_chuyen_can == true) {
            echo "Xếp loại: Khá <br>";
        } elseif ($diem_tb >= 5.0 && $co_di_hoc_chuyen_can == true) {
            echo "Xếp loại: Trung bình <br>";
        } else {
            echo "Xếp loại: Yếu (Cần cố gắng thêm!) <br>";
        }

        // TODO 4: Viết 1 hàm đơn giản (2.3)
```

```

function chaoMung()
{
    echo "Chúc mừng bạn đã hoàn thành PHT Chương 2!";
}

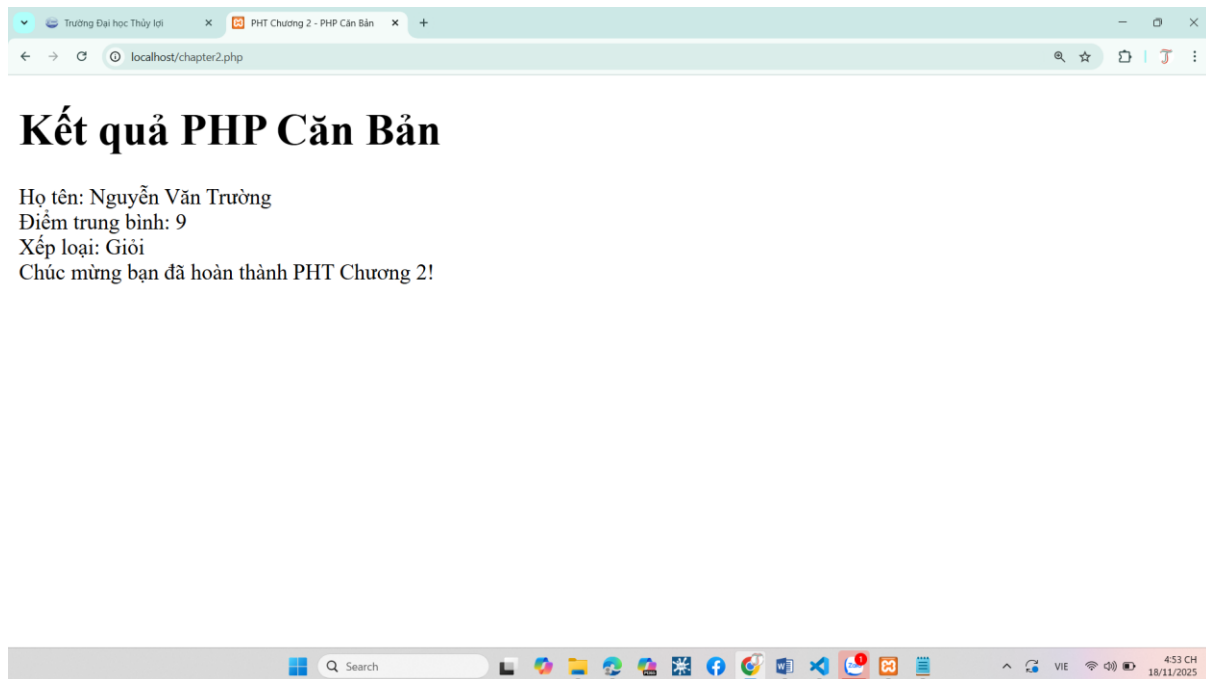
// TODO 5: Gọi hàm bạn vừa tạo
chaoMung();
?>

</body>

</html>

```

B. Ảnh chụp màn hình Kết quả (Trình duyệt Web): Chạy tệp PHP của bạn trên XAMPP (ví dụ: <http://localhost/chapter2.php>) và chụp ảnh màn hình trình duyệt hiển thị kết quả.



4. Câu hỏi Phản biện (Bắt buộc)

Sau khi hoàn thành Phần 2 & 3, hãy đặt 01 câu hỏi tư duy.

(Gợi ý: "PHP là ngôn ngữ kịch bản. Vậy sự khác biệt chính giữa 'Lớp và Đối tượng' (2.4) trong PHP so với 'Lớp và Đối tượng' trong một ngôn ngữ biên dịch như Java là gì? Tại sao PHP lại cần đến OOP?").

Câu hỏi của tôi là: Trong cấu trúc *if/else if/else* để xếp loại sinh viên, tại sao chúng ta phải kiểm tra điều kiện từ điểm cao nhất xuống (từ ≥ 8.5 rồi đến ≥ 6.5) thay vì ngược lại? Nếu đảo ngược thứ tự kiểm tra, logic xếp loại sẽ bị sai như thế nào?

5. Kết nối Đánh giá (Rất quan trọng)

Việc làm chủ các cú pháp if/else, function, và echo trong PHT này là bắt buộc.

Đây là những kỹ năng nền tảng nhất để bạn có thể làm Bài tập trên lớp (Phần PHP) , chiếm 20% tổng điểm, dự kiến vào Tuần 5. Nếu bạn không thể làm PHT này, bạn sẽ không thể lấy dữ liệu (\$_POST - Chương 3) hay truy vấn CSDL (Chương 4), vì tất cả đều cần PHP căn bản.