Web Programming - PHP

## XAMPP Installation

## Follow the document on Canvas to install XAMPP. If you are running MacOS, watch the video for installing XAMPP on your system.

Troubleshoot if you experience any issues. There are documents and links on Canvas.

## Instructions for Setting Up the Week 7 Exercises

## 1. Create a New Folder:

## - In your local wp directory (that is in the htdocs directory of XAMPP), create a folder named week7.

## 2. File Placement:

## - Place all new files for the exercises inside the week7 directory.

## 3. Naming Conventions:

## - Ensure there are no spaces in any file or folder names.

## - Use only lowercase letters for naming folders and files.

## 4. Reasons:

## - Avoiding spaces and using lowercase helps ensure compatibility with web environments, including case-sensitive servers like those at RMIT.

## Exercise 1

Create a file called vars.php. Make sure that you place this file in a subdirectory of *htdocs* directory of your XAMPP installation. You can call it week7 or give it any other name as long as it does not have spaces.

In vars.php create php variables that contain:

* Your name
* Your age
* Number 99.99
* A boolean with value *true.* Hint – booleans (true and false) do not have quotes around them

Print all these variables in a sentence using either *print* or *echo* functions.

## Exercise 2

Create a php file called hello.php. Inside this file add all html tags for a page (use a VSCode shortcut). Make sure that the file is saved as .php!

Create a variable called $name inside the page. You can add the variable on the top of the page or inside the html. Make sure you use php tags and the variable is declared before using it.

Inside the <body> element, in a paragraph, print a greeting with $name that you set before.

## Exercise 3

Create a php file called home.php. Create entire html content using echo statements.

## Exercise 4

Create a php file called includes.php. Inside this file add all html tags for a page (use a VSCode shortcut).

Inside the body element add a paragraph

<p>Here is our VVV page</p>

Add a subdirectory *includes* to the directory where includes.php file resides. Create two new files in *includes* subdirectory. Call them header.inc and footer.inc. In header.inc place all html tags from includes.php starting with

<!DOCTYPE html>

and finishing with

<body>

Just cut these lines from includes.php and paste into header.inc.

From includes.php cut lines

</body>

</html>

and paste them into footer.inc.

Your includes.php should have only 1 line at this stage

<p>Here is our VVV page</p>

Why we did all this cutting and pasting? Just to ensure that our starting point is a complete valid html page.

Now you need to modify includes.php to include header.inc on the top and footer.inc on the bottom of the file.

Set a variable with “testing php includes” string. Use this variable in a print statement to replace VVV in the paragraph.

View the includes.php page in the browser. Validate the **resulting html** (not php code) in W3C HTML validator.

## Exercise 5

Modify header.inc file you created in Exercise 4. Instead of printing a literal string for the title element, print a variable called $title. Set $title in includes.php. Make sure that you set it before you add header.inc as an include.

## Exercise 6

Create a php file called arrays.php. Create two arrays. One array will contain numbers from 1 to 9. Print numbers in positions 1, 3, 5, and 8. Try to print a number in position 9. What has happened and why?

The second array should be an associative array and should contain 3 names and ages. Like "Alison"=>20, "Roxi"=>37, "Pippin"=>43

Print the age of Pippin. Use foreach loop to print all names and ages

## Committing Exercise Files to Git via Visual Studio Code

1. **Open Your Project in VS Code**:

- Launch VS Code.

- Open your project folder (**week7**) by going to File > Open Folder and selecting your project directory.

2. **Stage Your Changes**:

- In VS Code, go to the Source Control panel (the icon looks like a forked branch on the sidebar).

- You'll see a list of untracked/changed files. Hover over the "Changes" section and click the '+' icon to stage all changes, or stage individual files by clicking the '+' icon next to each file.

3. **Commit the Changes**:

- After staging, type a commit message in the input box at the top of the Source Control panel. Follow best practices for commit messages (clear, concise, and descriptive).

- Click the Commit button and select “Commit & Sync” (you can also use “Commit & Push”)

A screenshot of a computer

Description automatically generated

## Deploy Code to the RMIT webserver

1. Open Terminal or Command Prompt:

- On your local machine, open your terminal or command prompt.

2. SSH into the Server:

- Use the command **ssh s1234567@titan.csit.rmit.edu.au** where s1234567 is your student ID

- Enter your RMIT password.

3. Navigate to the **wp** Directory:

- Once connected, navigate to the directory where your Git repository is located on the server using **cd ~/public\_html /wp**.

4. Pull the Latest Changes:

- Run the command **git pull origin main** to pull the latest changes from the remote repository.

5. Exit SSH Session (optional):

- After pulling the changes, you can end your SSH session by typing **exit**