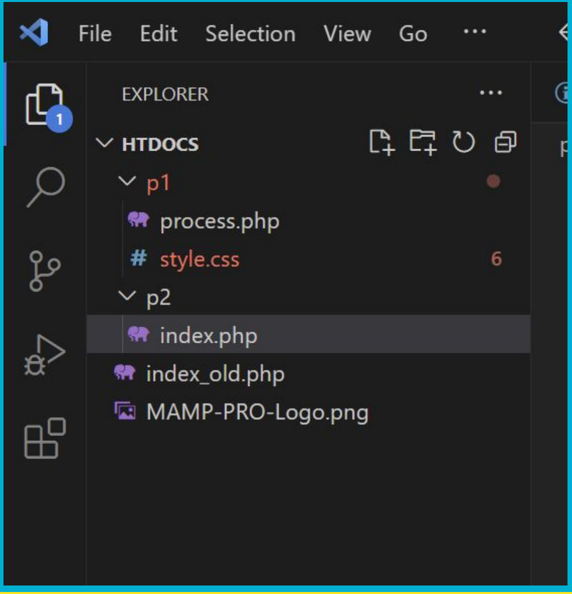
# PHP-I

Manas Jyoti Das, PhD
Computer science
SIUE

# PHP

- Open MAMP
- Use vscode and go to the folder htdocs found under the installation directory of MAMP
- Create a folder inside of it
- Name the file index.php
- It is easy to integrate with HTML (client side)
- PHP is a server side language
- Your server is in localhost:port_num



# Hello World

```
<!DOCTYPE html>
<html>
<head>
       <title>Hello World PHP</title>
</head>
<body>

  <?php
    echo "Hello, World!";
  ?>

</body>
</html>
```

# Embedding html into php

```
<!DOCTYPE html>
<html>
<head>
        <title>PHP with HTML</title>
</head>
<body>

<?php
        echo '<h1>Hello, World!</h1>';
        echo '<p>This is a PHP program using HTML</p>';
        echo '<p>You can embed HTML tags and content directly in PHP code.</p>';
?>

</body>
</html>
```

# Variables $

- It start with a $
- $variable_name

# Variable

```
<!DOCTYPE html>
<html>
<head>
  <title>PHP Variable Demo</title>
</head>
<body>
<?php
  $message = "Hello, World!";
  echo "<p>$message</p>";

  $num1 = 10;
  $num2 = 5;
  $sum = $num1 + $num2;
  $product = $num1 * $num2;

  echo "<p>The sum of $num1 and $num2 is: $sum</p>";
  echo "<p>The product of $num1 and $num2 is: $product</p>";
  $num1=15;
  echo "<p> num1 changed to $num1</p>"
?>

</body>
</html>
```

You can declare the variable anywhere before its use

# Data Types

- Integer
- Floats
- Strings
- Boolean (true or false)
- Arrays
- null
- Many more

# Array type

```
<!DOCTYPE html>
<html>
<head>
        <title>PHP Array Example</title>
</head>
<body>
<?php
        $fruits = array("apple", "banana", "cherry", "date");

        echo "<p>My favorite fruits are:</p>";
        echo "<ul>";
        echo "<li>" . $fruits[0] . "</li>";
        echo "<li>" . $fruits[1] . "</li>";
        echo "<li>" . $fruits[2] . "</li>";
        echo "<li>" . $fruits[3] . "</li>";
        echo "</ul>";

        $fruits[] = "grape";

        echo "<p>After adding 'grape', the array becomes:</p>";
        echo "<ul>";
        foreach ($fruits as $fruit) {
        echo "<li>$fruit</li>";
        }
        echo "</ul>";
?>

</body>
</html>
```

# String manipulation

```
<!DOCTYPE html>
<html>
<head>
	<title>String Length Demo</title>
</head>
<body>

<?php

	$text = "Hello, World!";

	$length = strlen($text);

	echo "<p>The length of the string \"$text\" is $length characters.</p>";
?>

</body>
</html>
```

# Strings

```
<!DOCTYPE html>
<html>
<head>
        <title>String Length Demo</title>
</head>
<body>

<?php

        $text = "Hello, World!";

    echo "The first letter is: $text[0]"
?>

</body>
</html>
```

# String

```
<!DOCTYPE html>
<html>
<head>
        <title>String Length Demo</title>
</head>
<body>

<?php

     $text = "Hello, World!";
    $text[0] ="M";
    echo "The word is: $text";
?>

</body>
</html>
```

# Substring

```
<!DOCTYPE html>
<html>
<head>
        <title>String Length Demo</title>
</head>
<body>

<?php

        $text = "Hello, World!";
  $text = substr($text,1,7);
  echo "The first letter is: $text";
?>

</body>
</html>
```

There are many more string manipulation functions in PHP

# Numbers

```
<!DOCTYPE html>
<html>
<head>
    <title>Increment Operator Demo</title>
</head>
<body>

<?php
    $x = 5;
    $y = ++$x; // $x is incremented first, then assigned to $y
    echo "<p>Pre-increment: x = $x, y = $y</p>";

    $a = 10;
    $b = $a++; // $a is assigned to $b, then incremented
    echo "<p>Post-increment: a = $a, b = $b</p>";
?>

</body>
</html>
```

- --
- +=
- *=
- abs(-99)
- pow(2,10)
- round(2.7)
- ceil(2.2)
- floor(3.8)

GUESS WHO'S BACK...
BACK AGAIN..

# HTML forms are back!!

# HTML form and PHP

```
<!DOCTYPE html>
<html>
<head>
        <title>User Input Demo</title>
</head>
<body>

<form method="GET">
        <label>Enter something:</label>
        <input type="text" name="userInput">
        <input type="submit" value="Submit">
</form>

<?php
        if ($_SERVER["REQUEST_METHOD"] == "GET") {
        $userInput = $_GET["userInput"];

        echo "<p>You entered: $userInput</p>";
        }
?>

</body>
</html>
```

# Another form

```
<!DOCTYPE html>
<html>
<head>
        <title>User Input Demo</title>
</head>
<body>
<form method="GET">
        <label>Enter name:</label>
        <input type="text" name="userInput">
  <br>
  <label>Enter GPA:</label>
  <input type="number" name="userGPA">
        <input type="submit" value="Submit">
</form>
<?php
        if ($_SERVER["REQUEST_METHOD"] == "GET") {
        $userInput = $_GET["userInput"];
         $userInput_gpa= $_GET["userGPA"];

        echo "<p>You entered: $userInput</p>";
         echo "<p>You entered: $userInput_gpa</p>";
        }
?>
</body>
</html>
```

# Get vs Post

- Main difference is how they send data to the server
- GET requests send data in the URL, while POST requests send data in the body of the request
- GET:
  - To retrieve a list of products from an e-commerce website
  - To load a web page
  - To search for a term on a website
- POST:
  - To create a new account on a website
  - To place an order on an e-commerce website
  - To submit a form on a website

# PHP in a separate file

index.html

```
<!DOCTYPE html>
<html>
<head>
        <title>Hello World Example</title>
</head>
<body>
        <h1>Hello, World!</h1>
        <p>This is a basic example of a PHP-HTML
combination.</p>

        <p>Display a message from a PHP script:</p>
        <?php include 'hello.php'; ?>
</body>
</html>
```

hello.php

```
<?php
$greeting = "Hello from PHP!";
?>

<p><?php echo $greeting; ?></p>
```