**PHP – HYPERTEXT PREPROCESSOR**

**ABOUT PHP:**

* Used to build scalable and easily maintainable websites.
* It integrates super well with HTML.
* It is a server-side language (sit on the web server and interact with the client to make the website more powerful).

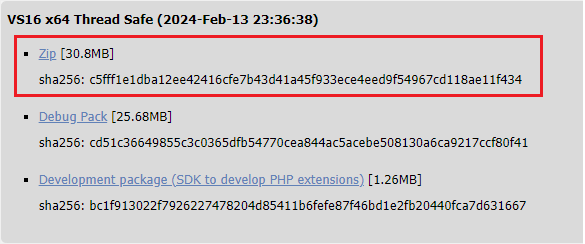
**SETUP PHP ON WINDOWS:**

Step 01: Visit <https://www.php.net/downloads>

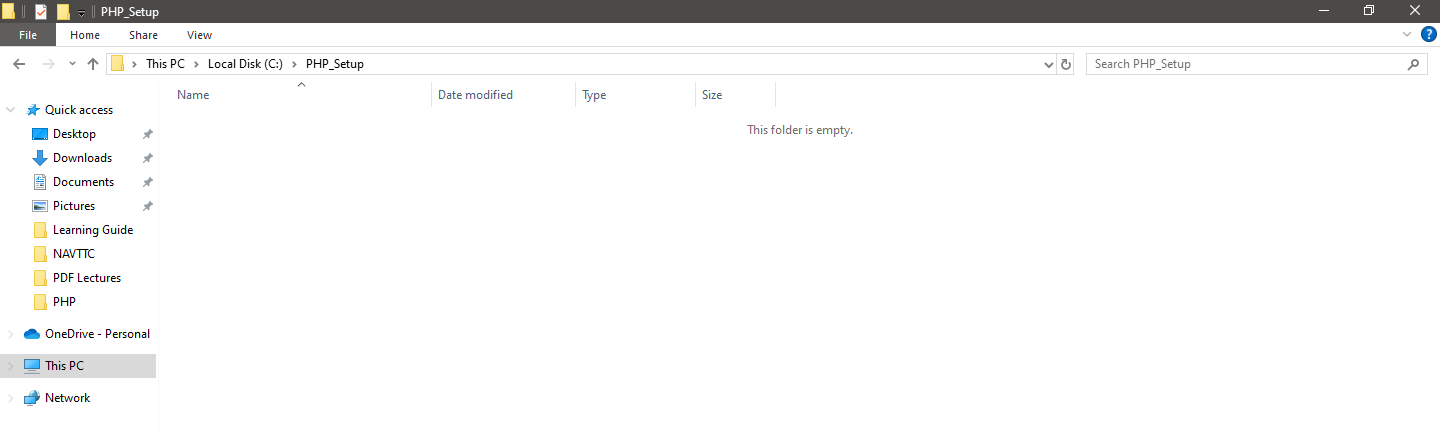
Step 02: Click on Windows downloads



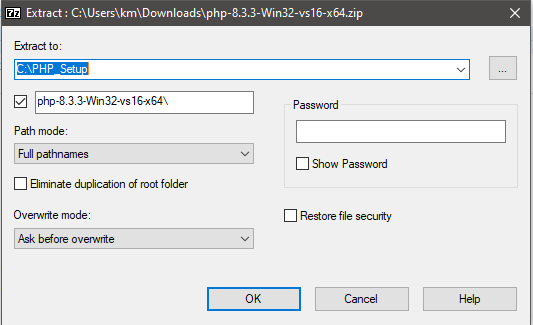
Step 03: Download the zip file based on your OS.

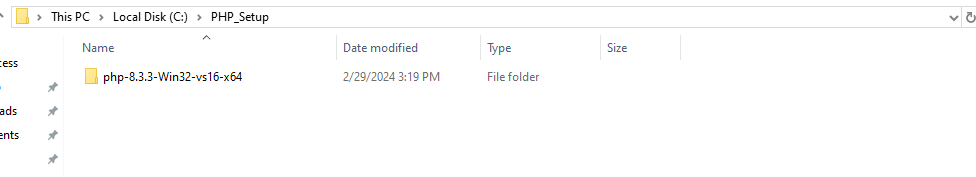


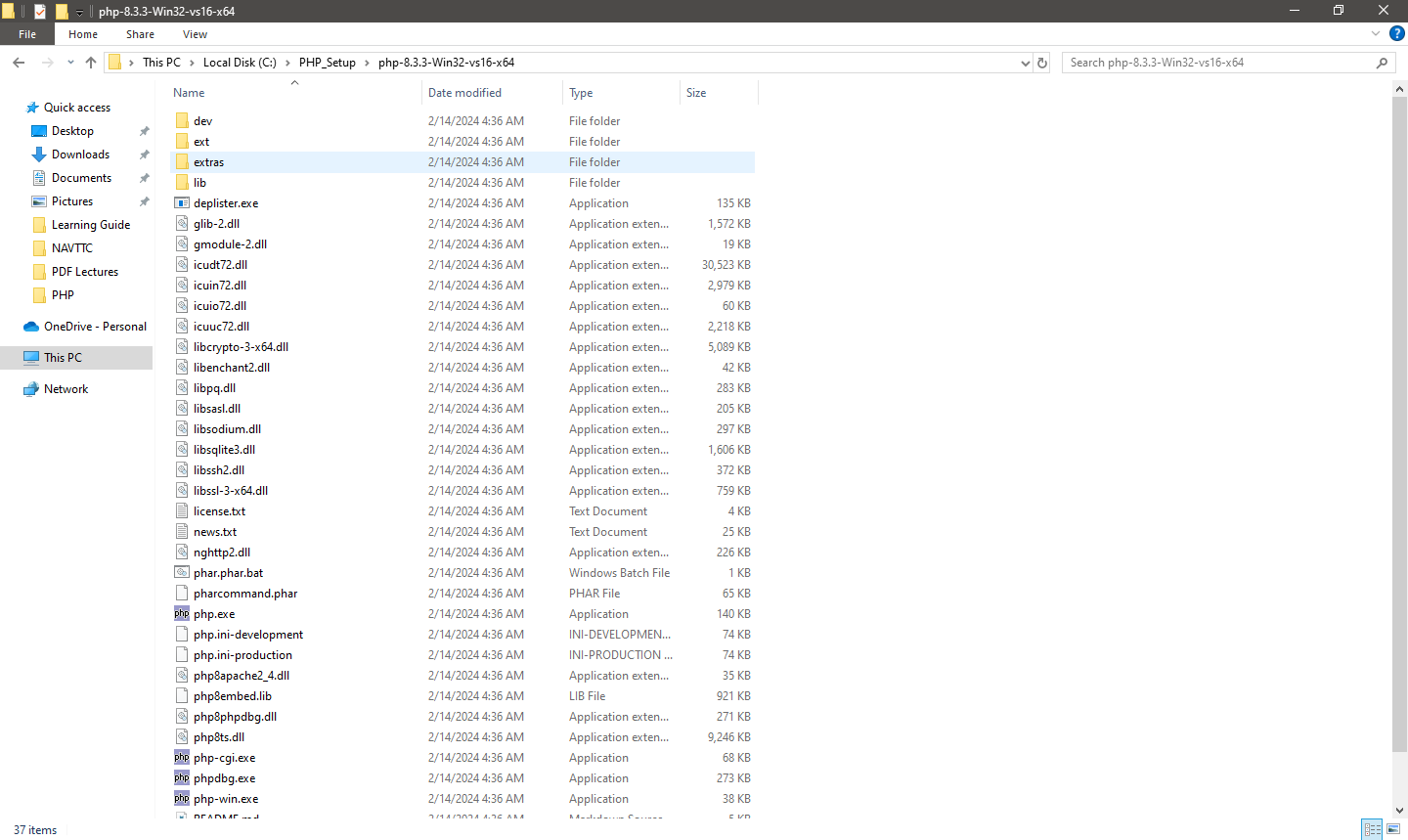
Step 04: Create a new folder on C drive with name “PHP\_Setup”



Step 05: Extract all the files from the zip file that you have downloaded earlier.

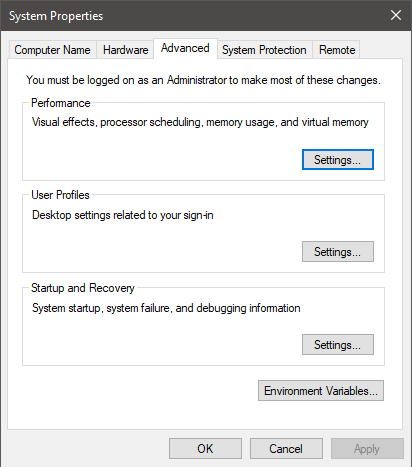




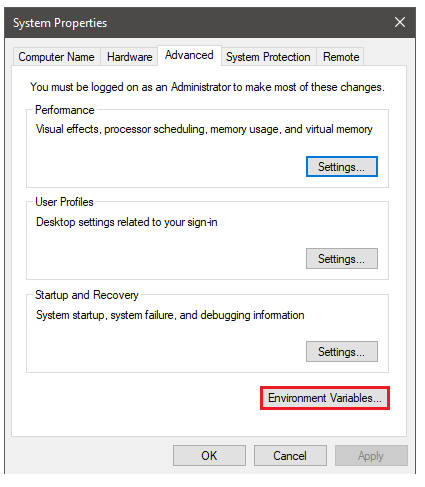


Step 06: Configuring Windows path variable

Type environment on the search bar and open the window.

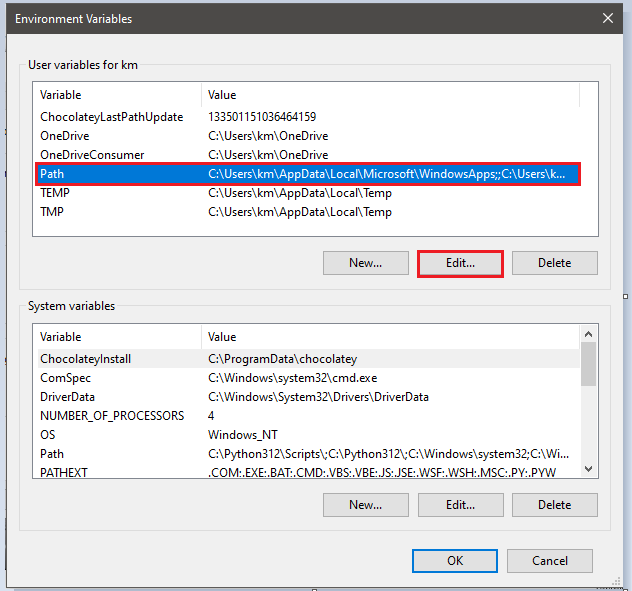


Click on “Environment Variables”;

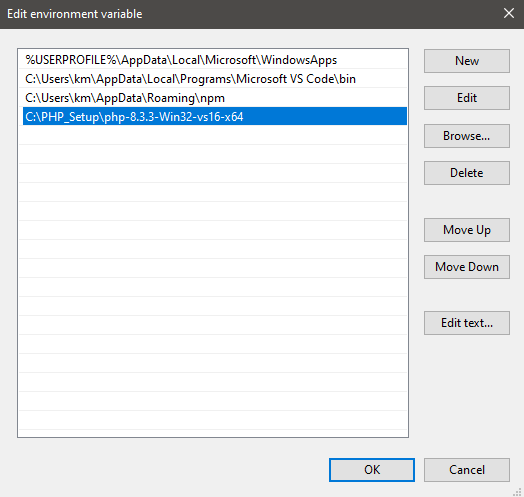


Step 07:

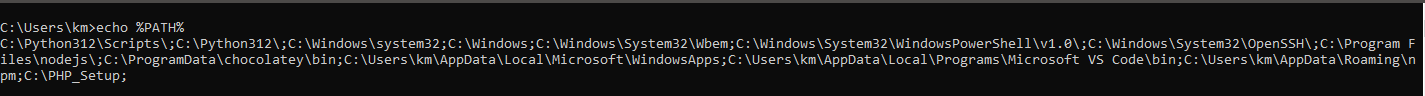
Select the path and click on Edit



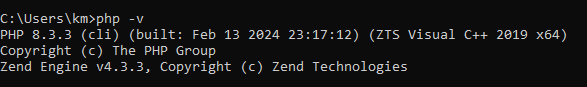
Click on New button and write the location of the directory where the PHP files are extracted.



Step 08: Open the command prompt and print out the path variable, using the command **“echo %path%”**



Step 09: Check the PHP version from the command prompt.

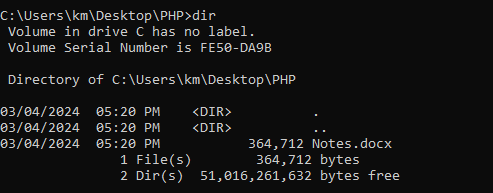


**CREATING OUR FIRST FILE AND RUN THE CODE:**

PHP is run on a web server.

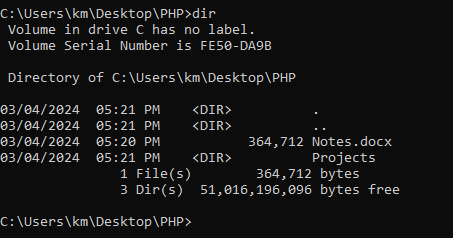
**Step 01: Create a new project in the current directory**

Checking the files of the current directory

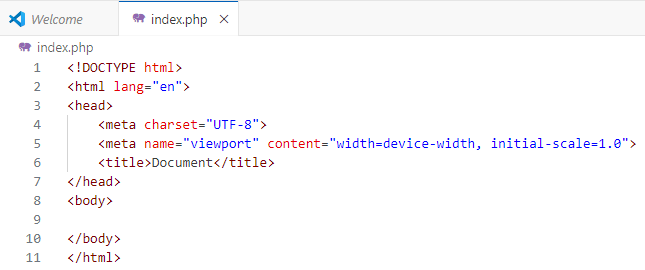


Create a new folder;





Step 02: Create a new file inside **Projects** folder.

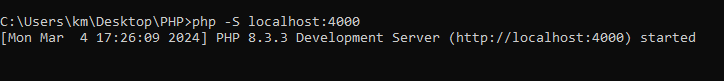


Step 03: Inside body tag, add php code.

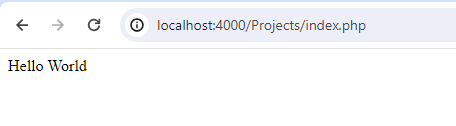


Step 04: Run a web server on a current directory.

Run the command “php -S localhost:4000”



Step 05: View the output on the web browser



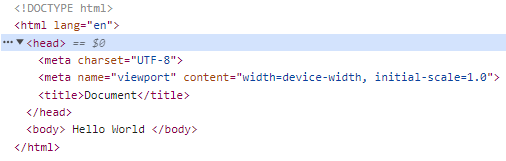
**WRITING HTML:**

ECHO:

A PHP command allows one to write HTML information onto the HTML document.

Example:

Inspecting the previous source code on the browser;

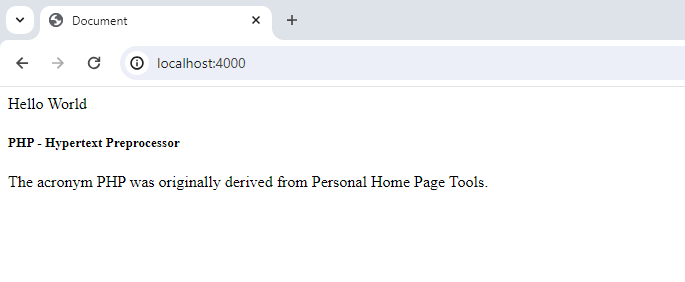


Example 02:

**Echo “<start tag> content </end tag>”** is used to write HTML.



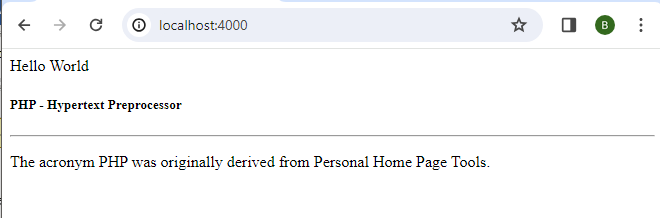
Output:



Example 03:



Output:



Note:

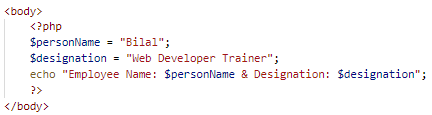
Never forget to place a semicolon at the end of the PHP line.

**VARIABLES:**

Variable are containers where we can store different pieces of information that we want a keep track of in our program. Syntax of defining variable:

**$variable\_name = variable\_value;**

Example 01:

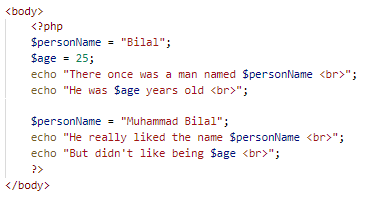


Output:

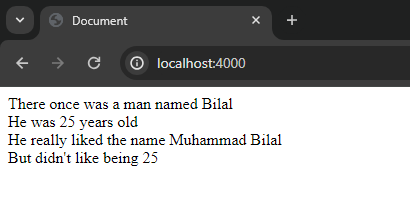


Example 02:

The value of a variable can be reassigned.



Output:



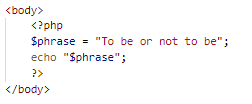
**DATA TYPES:**

1. String
2. Number
3. Boolean
4. Null

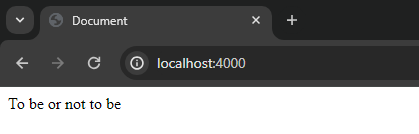
**STRING:**

They are used to write plain text.

Example:



Output:



**NUMBER:**

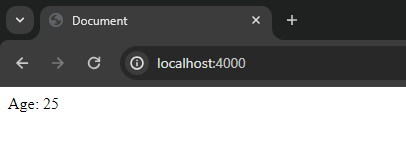
There are two categories of numbers in PHP:

1. Whole number
2. Decimal number

Example of Whole Number:

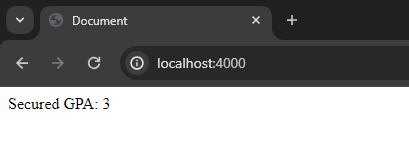


Output:



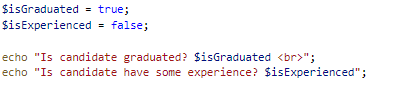
Example of Decimal Number:



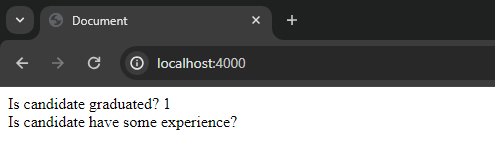


**BOOLEAN:**

Example:



Output:



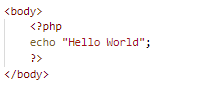
**NULL:**

“null” means no value.

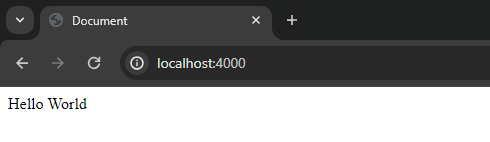
**WORKING WITH STRINGS:**

String: Plain text that are placed inside single quotation mark or double quotation mark.

Example:



Output:

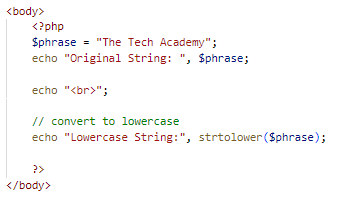


STRING FUNCTION:

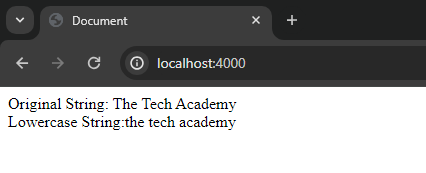
They are used to find information on the string and modify the string.

1. strtolower(“string”): To convert all the letters of the string to lowercase.

Example:

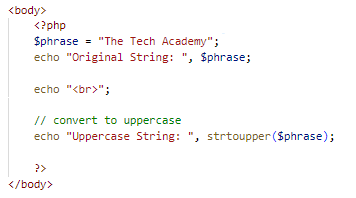


Output:

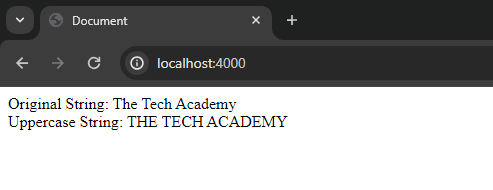


2. strtoupper(“string”): Convert all the letter of the string to uppercase.

Example:

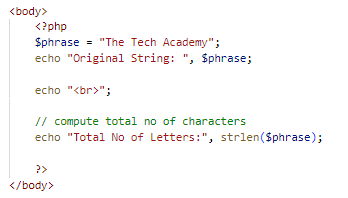


Output:

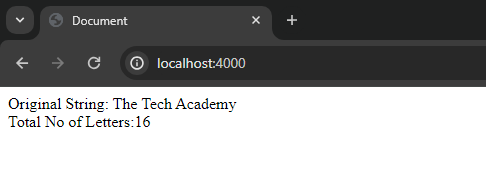


3. strlen(“string”): Return total count of letters in a string.

Example:



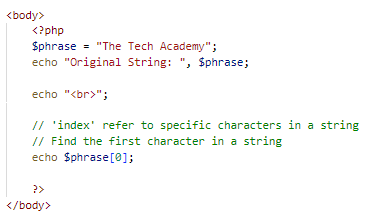
Output:



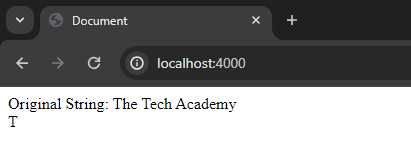
4. FIND THE LETTER IN A STRING USING INDEX:

“index” is a specific character in a string.

Example:

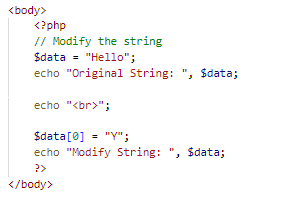


Output:

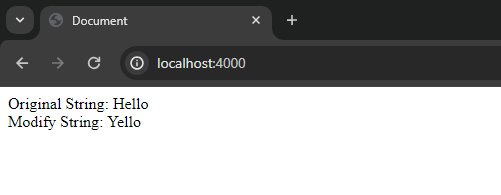


5. MODIFY THE STRING:

Example:

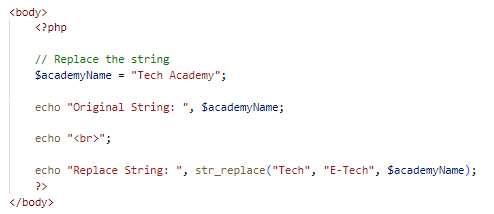


Output:

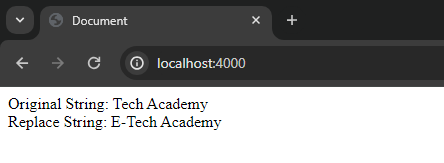


6. REPLACE SUB-STRING:

Example:

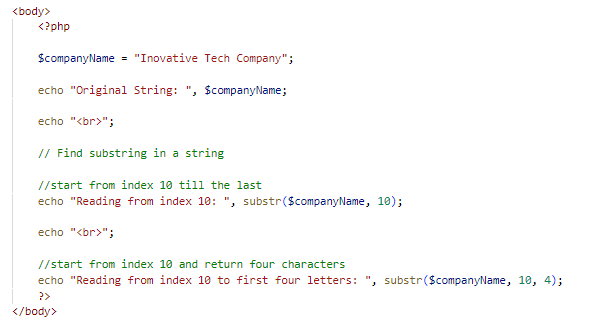


Output:

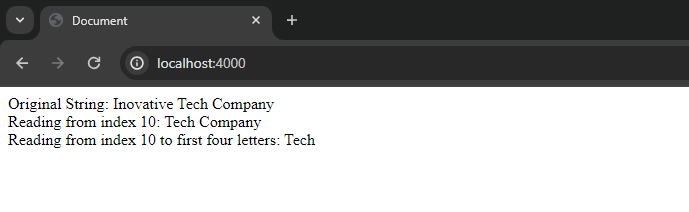


7. FIND SUBSTRING IN A STRING:

Example:



Output:



**WORKING WITH NUMBERS:**

|  |  |  |
| --- | --- | --- |
| **#** | **EXAMPLE** | **OUTPUT** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 |  |  |
| 6 |  |  |
| 7 |  |  |
| 8 |  |  |
|  | Order Of Operation   1. Division 2. Multiplication 3. Addition 4. Subtraction | |
| 9 | Here, first 5\*10 which is 50, and then 4 is added to 50, resulting in 54 |  |
| 10 | The bracket is placed, to change the order of operation; |  |
| 11 |  |  |
| 12 |  |  |
| 13 |  |  |
| 14 | Using shorthand, increment to the number; |  |

**MATH OPERATION USING BUILT-IN FUNCTION:**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **FUNCTION** | **EXAMPLE** | **OUTPUT** |
| 1 | Absolute | Return the absolute of a number; |  |
| 2 | Power | “pow(a, b): a\*\*b” |  |
| 3 | Square root |  |  |
| 4 | Maximum |  |  |
| 5 | Minimum |  |  |
| 6 | Round | Round to the integer number; |  |
| 7 | Ceil | Round up to the integer number; |  |
| 8 | Floor | Round down to the integer number; |  |

**GETTING USER INPUT:**

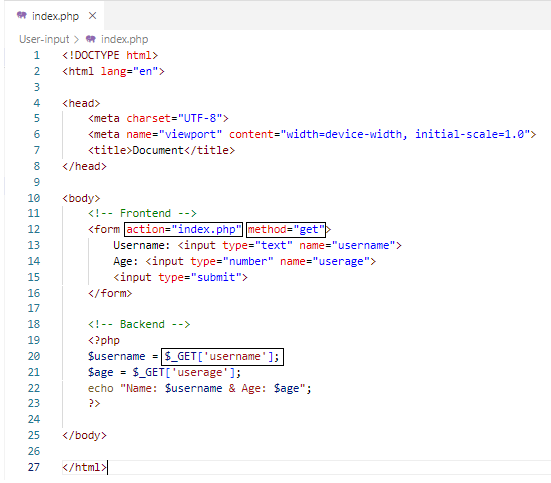
FORM:

A form is a basic HTML element, that allows the user to input information and send that information to a PHP program (server). The form is like a middleman between HTML elements and PHP.

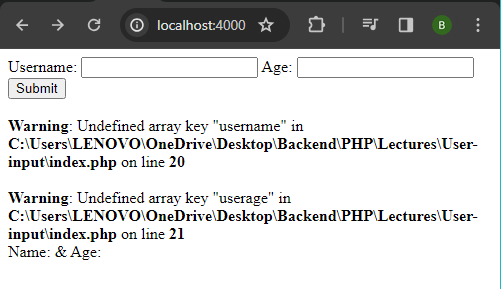
Example:

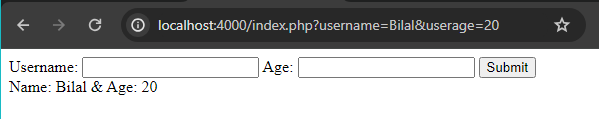
In below code:

* **“action = ‘index.php’”** is the path of the PHP file.
* Method=get refers to reading data in the below file.
* $\_GET[‘username’] – It is the value of the **name** attribute of the input field.

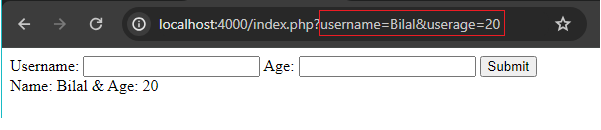


Output:





The highlighted box in the below figure tell us the value of variables:



Anytime that we’re entering information with the form, when the form get submitted, the information that got submitted is appear in the URL.