



**Prepared By: Eng. Noha Salah**  
March 2021

# PHP

## Day 3



# Day 3 Contents

- Form
- Form Validation
- File Uploading



# PHP- GET Method

- The GET method produces a long string that appears in your server logs, in the browser's Location: box.
- The GET method is restricted to send upto 1024 characters only.
- Never use GET method if you have password or other sensitive information to be sent to the server.
- GET can't be used to send binary data, like images or word documents, to the server.
- The data sent by GET method can be accessed using QUERY\_STRING environment variable.
- The PHP provides \$\_GET associative array to access all the sent information using GET method.



# PHP- POST Method

- The POST method does not have any restriction on data size to be sent.
- The POST method can be used to send ASCII as well as binary data.
- The data sent by POST method goes through HTTP header so security depends on HTTP protocol. By using Secure HTTP you can make sure that your information is secure.
- The PHP provides `$_POST` associative array to access all the sent information using POST method.



# Form Validation

- Validation is a must, You should never trust user inputs.
- If you try to use a variable that does not exist PHP processor will through a notice. This is why if you are unsure about the existence of specific variable you should check first.
- `isset($variable)` :

To determine if a variable exists and is not NULL. Return TRUE if the variable exist is not null Return False if not.

```
if (isset($_POST["name"]))      { //code ..... }
```



# Form Validation - Cont'd

- Handling Empty Form Fields

However, users often forget /don't want to fill in certain fields in a form, some data is not sent to the server. This empty field is sent as an empty string; sometimes no field name is sent at all.

- `empty($variable)` :

Determine whether a variable is empty.

Return TRUE if the variable is an empty and an unset variable. Return False if not.



# Generate an Interactive Valid Form

```
<?php  
include('template/header.html');  
  
if (/* form has been submitted */) {
```

*validation*

```
} else {
```



*form*

```
}  
include('template/footer.html');  
?>
```



# PHP- File Uploading

- A PHP script can be used with a HTML form to allow users to upload files to the server.
- Initially files are uploaded into a temporary directory and then relocated to a target destination by a PHP script.
- It's necessary for both temporary and final locations to have permissions set that enable file writing. If either is set to be read-only then process will fail.
- An uploaded file could be a text file or image file or any document.



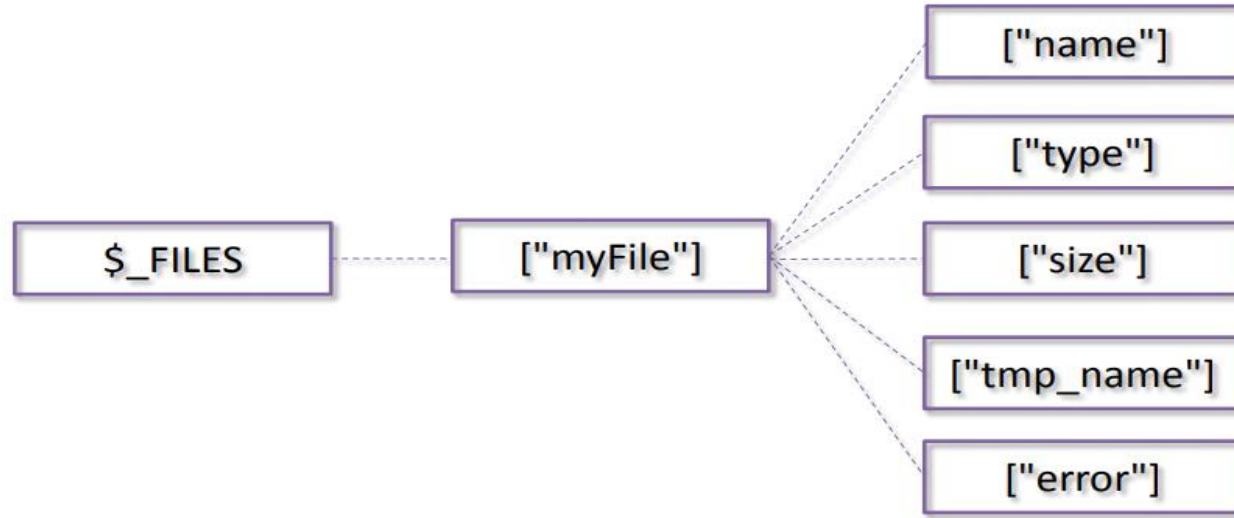
There is one global PHP variable called `$_FILES`. This variable is an associate double dimension array keeps all the information related to uploaded file.

In uploading form file, PHP would create following five variables :

- `$_FILES['file']['tmp_name']` – the uploaded file in the temporary directory on the web server.
- `$_FILES['file']['name']` – the actual name of the uploaded file.
- `$_FILES['file']['size']` – the size in bytes of the uploaded file.
- `$_FILES['file']['type']` – the MIME type of the uploaded file.
- `$_FILES['file']['error']` – the error code associated with this file upload.



- By using `$_FILES` array you can upload files from a client computer to the remote server.



- You MUST add the `enctype= "multipart/form-data"` attribute to your `<form>` so it can handle files.