

# Sessions Management

# Introduction

- ❖ The HTTP protocol is stateless. The server doesn't associate requests with specific clients.
- ❖ A session is the state in between requests from the same client (even if the time intervals are big).
- ❖ The PHP engine maintains the sessions by passing a unique session identifier between the requests. The session identifier is usually a cookie.

# Introduction

- ❖ The PHP engine transparently handles the sessions through cookies and URL rewriting.
- ❖ The alternative URL rewriting technique for managing sessions is available.

# The `session_start()` Function

- ❖ Calling the `session_start()` function in the beginning of our script will start a session.

```
bool session_start()
```

Calling this function creates a session or resumes the current one based on the current session id that was passed via the request (GET or POST) or based on the required cookie that already exists. This function returns 'true' in both cases.

# The `$_SESSION` Super Global Array

- ❖ During the session we can set/access its data via the `$_SESSION` super global array.

```
$_SESSION['id'] = 10001002;
```

This code adds the 'id' key together with the '10001002' value in the `$_SESSION` super global array.

```
if($_SESSION['id']==10001002) {...}
```

Throughout the session we will always be able to retrieve the value associated with the 'id' key and use it to identify our client in order to handle their session properly.

# The `session_status()` Function

- ❖ Calling the `session_status()` function we shall get a numeric value that indicates about the status of the current session. That value can be one of the following constants:

`PHP_SESSION_DISABLED`, `PHP_SESSION_NONE` or  
`PHP_SESSION_ACTIVE`.

```
int session_status()
```

Calling this function we shall get one of the following constants:

`PHP_SESSION_DISABLED`, `PHP_SESSION_NONE` or `PHP_SESSION_ACTIVE`. The returned value indicates about the session status.

# The `session_id()` Function

- ❖ Calling the `session_id()` function we shall get the unique ID of the current session. Calling this function we can also set that ID by passing it over to the function.

```
string session_id([string])
```

Passing over the ID it will replace the current session ID with our new one.

# Session Upload Progress

- ❖ When the `session.upload_progress.enabled` INI option is enabled the upload progress of each individual file being uploaded will be tracked.
- ❖ The detailed information about the current state of a file being uploaded will be available via the `$_SESSION` super global array using a key as explained in the following code sample.



# Session Upload Progress

```
<?php
$key = ini_get("session.upload_progress.prefix").
        $_POST[ini_get("session.upload_progress.name")];
print_r($_SESSION[$key]);
?>
```

# Session Upload Progress

- ❖ Getting the status of an uploaded file is highly useful when the HTTP request arrives via an XHR object from code written in Java Script.

# The `session_start()` Function Options

- ❖ As of PHP 7, we can call this function and pass over an array of options (php.ini options) in order to configure the way this function works.

```
session_start(['use_only_cookies' => true]);
```