In This file, I Will Explain What superglobal examples do, in brief :

PHP $GLOBALS

$GLOBALS is a PHP superglobal variable that is used to access global variables from anywhere in the PHP script (also from within functions or methods).

PHP stores all global variables in an array called $GLOBALS[index]. The index holds the name of the variable.

PHP $_SERVER

$_SERVER is a PHP superglobal variable that holds information about headers, paths, and script locations.

PHP $_REQUEST

PHP $_REQUEST is a PHP superglobal variable that is used to collect data after submitting an HTML form.

The example in the index file shows a form with an input field and a submit button. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the <form> tag. In this example, I point to this file itself for processing form data. Then, I can use the super global variable $_REQUEST to collect the value of the input field.

PHP $_GET

PHP $_GET is a PHP superglobal variable that is used to collect form data after submitting an HTML form with method="get".

$_GET can also collect data sent in the URL.

If I Assume we have an HTML page that contains a hyperlink with parameters:

```
<html>
<body>
<a href="test_get.php?subject=PHP&web=W3schools.com">Test $GET</a>
</body>
</html>
```

Then When I click on the link "Test $GET", the parameters "subject" and "web" are sent to "test_get.php", and I can then access their values in "test_get.php" with $_GET.

The example in the index file shows the code in the "test_get.php file".

PHP $_POST

PHP $_POST is a PHP superglobal variable that is used to collect form data after submitting an HTML form with method="post". $_POST is also widely used to pass variables.

The example in the index file shows a form with an input field and a submit button. When I submit the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the <form> tag. In this example, I point to the file itself for processing form data. Then, I can use the super global variable $_POST to collect the value of the input field.

PHP Create/Retrieve a Cookie

The index file also creates a cookie named "user" with the value "hadeel". The cookie will expire after 30 days (86400 * 30). The "/" means that the cookie is available on the entire website (otherwise, select the directory you prefer).

I then retrieve the value of the cookie "user" (using the global variable $_COOKIE). We also use the asset() function to find out if the cookie is set.

A session is a way to store information (in variables) to be used across multiple pages. Unlike a cookie, the information is not stored on the user's computer.

What is a PHP Session?
When you work with an application, you open it, do some changes, and when you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet, there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain a state.
Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, session variables last until the user closes the browser.
So; Session variables hold information about one single user, and are available to all pages in one application.
Tip: If you need permanent storage, you may want to store the data in a database.
Start a PHP Session
A session is started with the session_start() function.
Session variables are set with the PHP global variable: $_SESSION.
Now, On the index page, I start a new PHP session and set some session variables.

$_FILES
$_FILES — HTTP File Upload variables
Description
An associative array of items is uploaded to the current script via the HTTP POST method. The structure of this array is outlined in the POST method uploads section.
Note: This is a 'superglobal', or automatic global, variable. This simply means that it is available in all scopes throughout a script.

$_ENV
$_ENV — Environment variables
Description
An associative array of variables is passed to the current script via the environment method. These variables are imported into PHP's global namespace from the environment under which the PHP parser is running. Many are provided by the shell under which PHP is running and different systems are likely running different kinds of shells, a definitive list is impossible. Please see your shell's documentation for a list of defined environment variables.
Other environment variables include the CGI variables, placed there regardless of whether PHP is running as a server module or CGI processor.