

🔗 PHP session architecture overview

There are **three “places”** where session information lives:

Layer	What it contains	Location	Who controls it
1 In-memory (\$_SESSION)	Your PHP variables during script execution	In RAM (only while the script runs)	PHP (runtime)
2 On disk (session file)	Serialized session data between requests	e.g. xampp\tmp\sess_abcd1234	PHP (server)
3 Cookie (PHPSESSID)	Just the session ID string	Browser (client)	Client (browser)

⚙️ What happens step by step

1. When you start a session

`session_start();`

- PHP checks if the browser sent a cookie named PHPSESSID.
- If yes → it loads that session's file (e.g. sess_abc123) into memory as `$_SESSION`.
- If no → PHP creates a new session ID and sends a cookie back to the browser.

✅ At this point, three things exist:

- A **file** on disk
 - A **copy of the data** in memory (`$_SESSION`)
 - A **cookie** in the browser containing the session ID only
-

2. When you call `session_unset()`

`session_unset();`

- PHP clears the `$_SESSION` array (the in-memory copy).

- The **file** on disk is still there, but PHP will overwrite it with an *empty* array when the script ends.
- The **cookie** is untouched.

💡 So now the session file may remain, but with **no data** inside (0 KB).

3. When you call `session_destroy()`

`session_destroy();`

- PHP **deletes** the session file (e.g. `/xampp/tmp/sess_abc123`).
- The **`$_SESSION` array in memory** is *not* touched — it still has all your data for the rest of the script.
- The **cookie in the browser** still exists, and still holds the session ID string.

So even though the file is deleted, your PHP script still has a copy of the data sitting in memory (`$_SESSION`) until the script ends or you manually clear it.

⚠ Important distinction

The cookie **never contains the actual session data** — it only stores the **session ID** (a random string).

So when you say:

“It still remains — is it in the same cookie?”

No 😊 — the cookie doesn’t store your session variables. It just stores something like:

`PHPSESSID=abc123xyz456`

That’s just an ID.

🔍 So where does it "remain"?

When you don’t call `session_unset()`:

- The session file **is deleted** (by `session_destroy()`).

- But the data **still exists in the current PHP process memory** — inside `$_SESSION`.

So it's temporarily living in PHP's memory (RAM) for that request only, **not** in the cookie and **not** on disk anymore.

Once your script ends, the memory is cleared automatically, and the next page load won't find anything — because the file's gone.

Summary Table

Layer	Contains	Cleared by <code>session_unset()</code>	Cleared by <code>session_destroy()</code>	Cleared by deleting cookie
<code>\$_SESSION</code> (memory)	Session variables	✓ Yes	✗ No	✗ No
Session file (disk)	Saved data	✗ No	✓ Yes	✗ No
Cookie (browser)	Session ID only	✗ No	✗ No	✓ Yes

✓ So the full clean-up sequence (logout) is:

```
session_start();
```

```
session_unset(); // clear memory
```

```
session_destroy(); // delete file
```

```
setcookie(session_name(), "", time() - 3600, '/'); // delete cookie
```

After this:

- Memory → cleared
- File → deleted
- Cookie → gone

Nothing remains anywhere **100**