

Understanding the Output Buffer in PHP

The **output buffer** in PHP is a mechanism that allows capturing the HTML or text content generated by a script **before it is sent to the browser**.

Why use buffering?

- **Separate content from display** (useful for templates or rendering engines)
- **Modify the HTML content** before displaying it
- **Dynamically assemble multiple content blocks**
- **Prevent premature output** (especially for HTTP redirections)

Main Functions

PHP Function	Description
<code>ob_start()</code>	Starts the buffer
<code>ob_get_clean()</code>	Gets the buffer content and clears it
<code>ob_get_contents()</code>	Gets the buffer content without clearing
<code>ob_end_clean()</code>	Clears the buffer without displaying
<code>ob_flush()</code>	Manually sends buffer content to browser

Basic Example

```
ob_start(); // Start buffering

echo "Hello";
echo " world!";

$content = ob_get_clean(); // Capture generated output

echo strtoupper($content); // Displays: HELLO WORLD!
```

Example in a Template System

File: `template.php`

```
$title = "Home";
ob_start();
require 'page.php';
$content = ob_get_clean();
require 'layout.php';
```

File: `layout.php`

```
<!DOCTYPE html>
<html>
<head><title><?= $title ?></title></head>
<body>
    <?= $content ?>
</body>
</html>
```

✓ Educational Advantages

- Helps **understand how template engines work**
- Approach **close to modern PHP frameworks**
- Makes code **modular and readable**

⚠ Note

- Buffering is **active by default** in some contexts (e.g., CMS or frameworks).
- It may interfere with HTTP headers (`header()`) if not cleared properly.