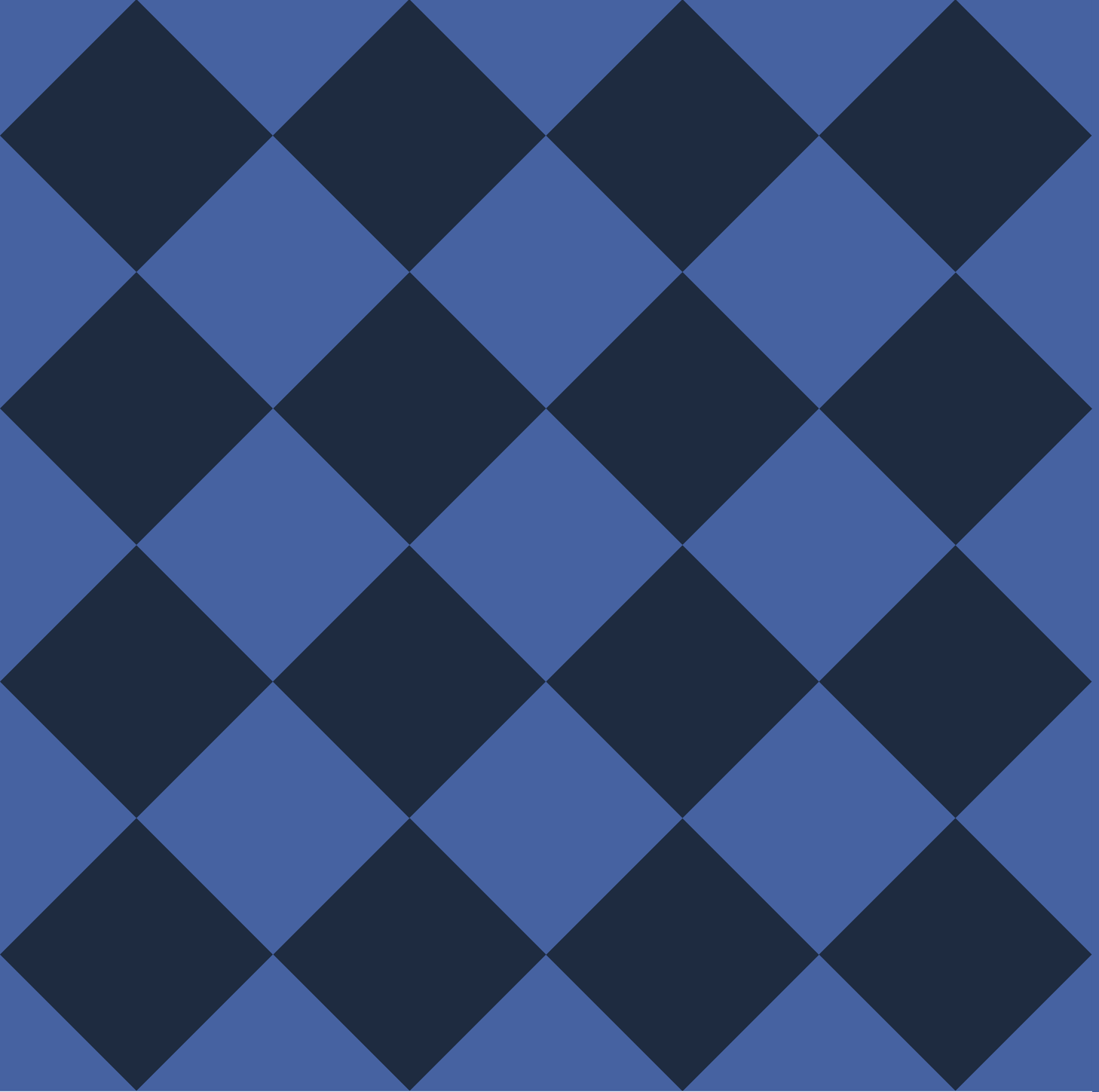
PHP Advanced

المقدم | Jassim Alabdulwahab

التاريخ | 2025.11.13

TOPICS



SUPERGLOBALS

SUPERGLOBALS

Superglobals are predefined variables in PHP which are always accessible regardless of scope (i.e $GLOBALS[‘x’] is always accessible regardless of where it was called)

Superglobals are important for retrieving user-provided data

```php
<?php

    $_COOKIE['1'];
    $_GET['2'];
    $_POST['3'];
    $_REQUEST['4'];
    $_COOKIE['5'];
    $_SESSION['6'];
    $_SERVER['7'];
    $_FILES['8'];
    $_ENV['9'];
?>
```

SUPERGLOBALS CONT.

01

$_GET

Used for retrieving data provided in URL along with GET request (look up HTTP GET for more info)

I.E: GET /catalog.php?id=54 HTTP/1.1

$_GET[‘id’] => 54

02

$_POST

Used for retrieving POST data in request body

I.E: POST /login.php HTTP/1.1

…

user=john&pass=doe

$_POST[‘user’] => “john”
$_POST[‘pass’] => “doe”

03

$_COOKIE

Used to retrieve persistent data (discussed later)

FORM HANDLING

FORM HANDLING

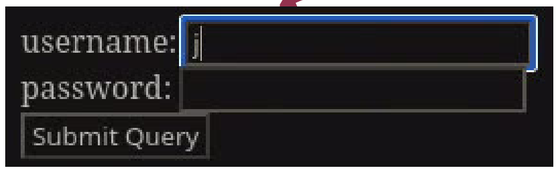
We can use Superglobals to retrieve form data from something like login pages.

```html
<form action="backend.php" method="POST">
username: <input type="text" name="name"><br>
password: <input type="text" name="pass"><br>
<input type="submit">
```

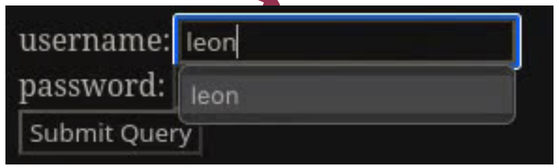
POST backend.php HTTP/1.1
...
name=john&pass=doe

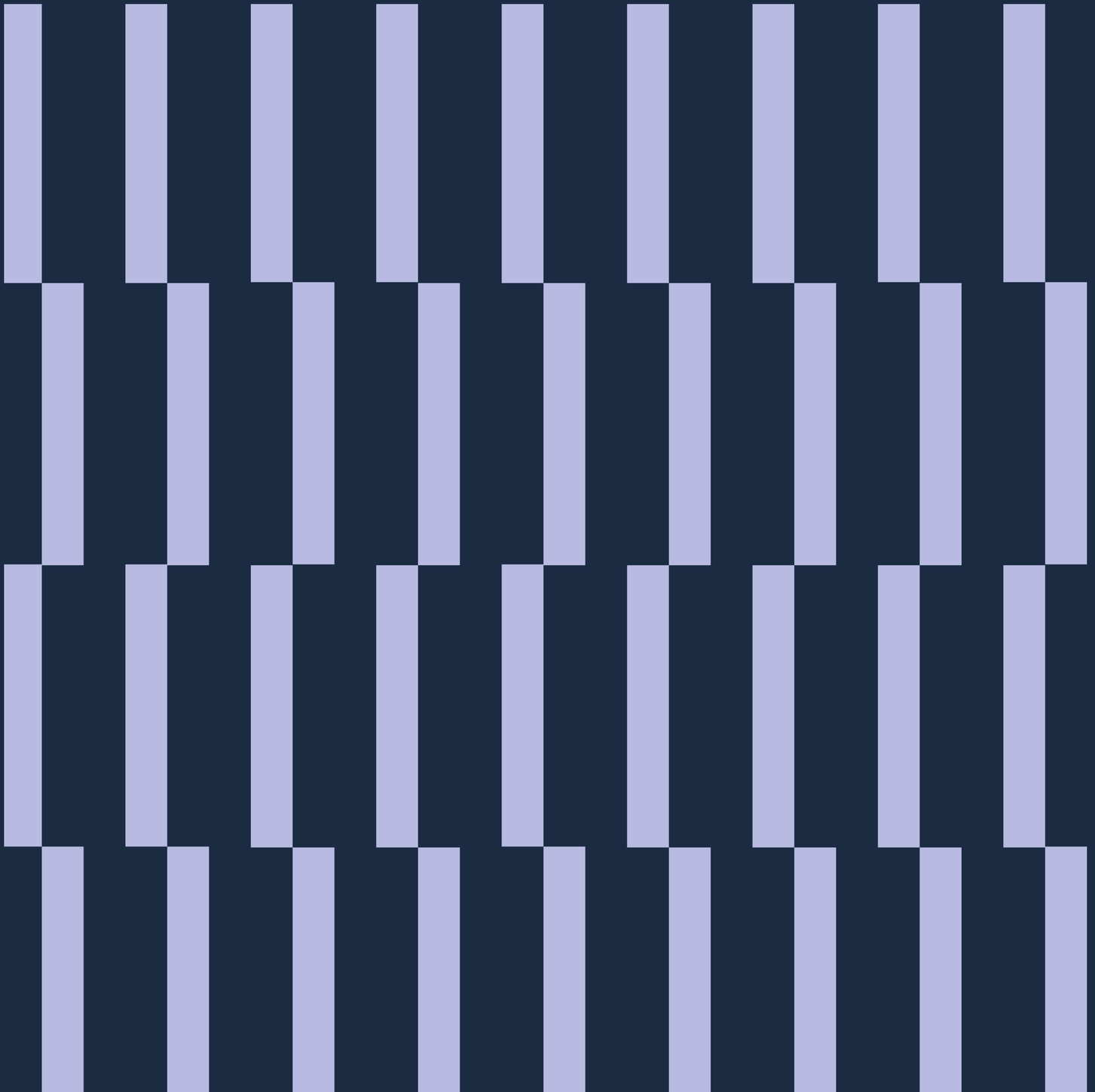
```php
<?php
    function verify($user, $pass) {···
    }

    $verified = verify($_POST['name'], $_POST['pass'])
    // ... verify user <discussed in next lecture>
    if ($verified) {
        echo "Welcome " . $_POST['name'];
    } else {
        echo "incorrect username or password"
    }
?>
```

user is verified



user isn't verified





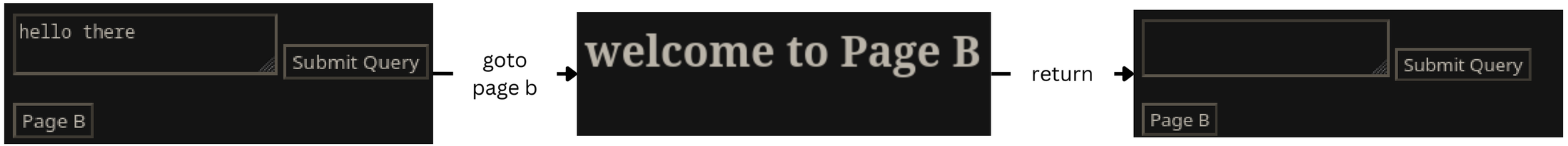
COOKIES & SESSIONS

Cookies

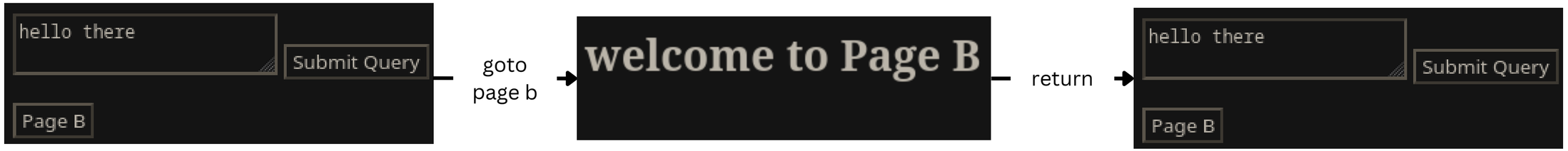
Websites are stateless! Meaning, by default, they don't remember your previous actions.

this is solved using cookies. **I.e:**

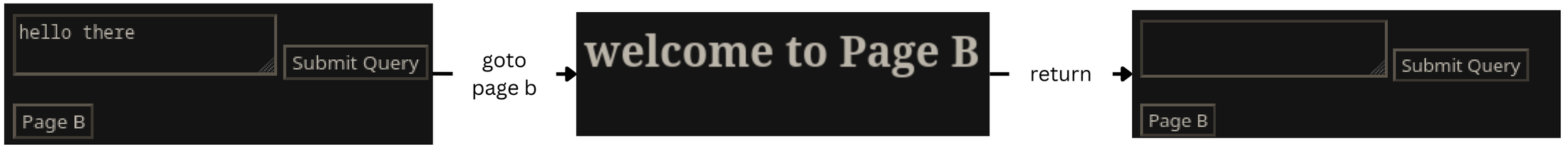
without cookies:



with cookies



without Cookies



Submit query

GET b.php HTTP/1.1

...

Cookies: (nil)

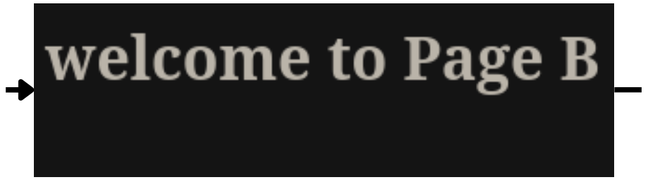
GET / HTTP/1.1

...

Cookies: (nil)

with Cookies

goto page b



return

Submit query

```php
<?php
    setcookie('textcontent', $_POST['textbox']);
?>
```

GET b.php HTTP/1.1

…

Cookies:

textcontent=”hello there”

GET / HTTP/1.1

…

Cookies:

textcontent=”hello there”

```php
<textarea name="textbox">
    <?php echo $_COOKIE['textcontent'];?>
</textarea>
```

SESSIONS

Sessions are another way of storing information across webpages. Unlike cookies however, they store data on the server-side. Not the client-side

An example of a session would be an E-mail client. You can have 10 drafts ready but storing them as browser cookies is a security risk. Sessions fix this by storing the drafts server-side and providing them to you after logging in.

It is preferable to store data in sessions rather than cookies

Sessions in action

SELF-STUDY

The code presented in this lecture was very minimal and the topics themselves are very heavy. This week’s self-study will just be a deeper look into today’s topics

MILESTONE 1

Thanks!