W12 – PHP – Web Programming III

Cookie / Session

Cookies

- Text file stored on the **Client** computer.

- Cookies are associated with one domain at a time. Necessarily associated to a domain name.

- Relies on the HTTP protocol in the headers of the request.

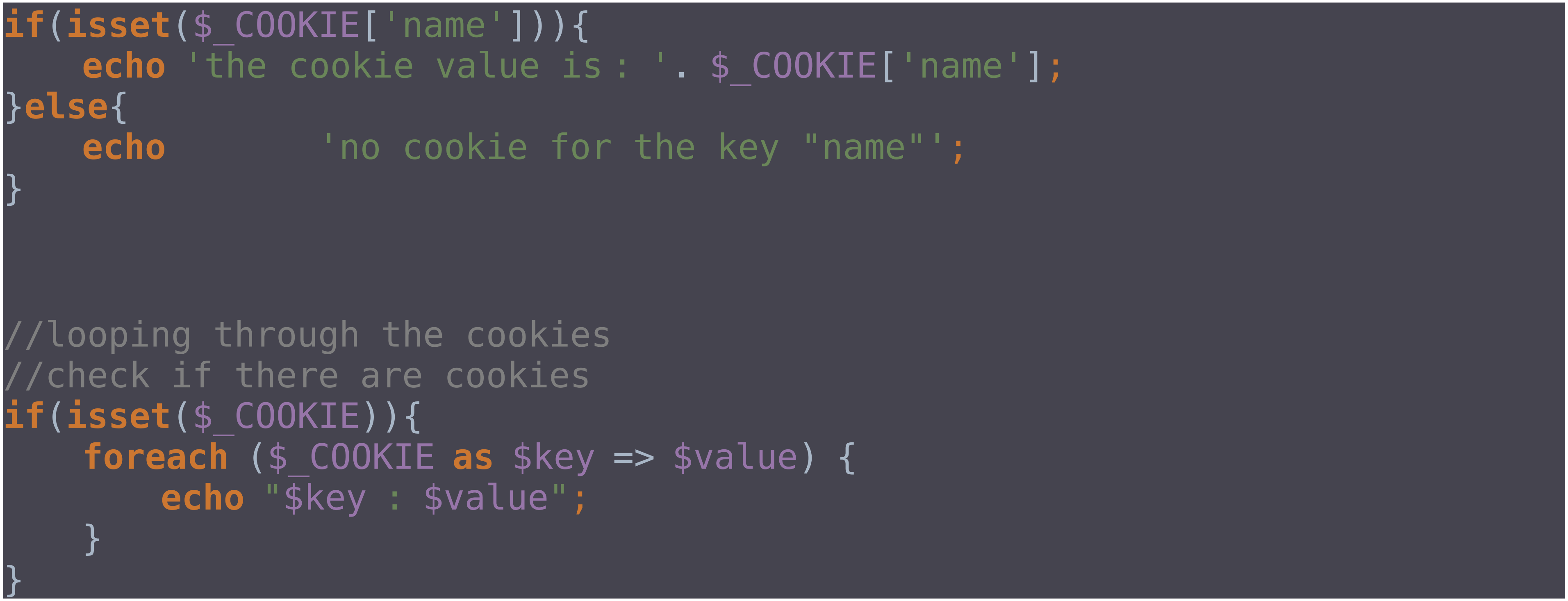
- Set of key-value pairs in a text file managed by the browser. Both the Key and value are of type string.

- A client can store a maximum of 300 cookies, 20 of which can come from the same domain

- Each cookie can be up to 4000 bytes (about 4 KB). So size is limited.

Accessing via $_COOKIE

- Recover cookie from $ _COOKIE
- $_COOKIE is an associative array
- Always check if cookie exists with isset()



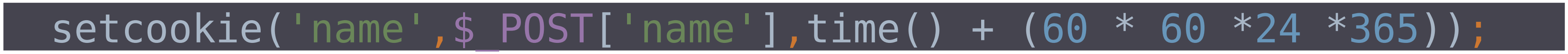
```
if(isset($_COOKIE['name'])){
    echo 'the cookie value is : '. $_COOKIE['name'];
}else{
    echo       'no cookie for the key "name"';
}



//looping through the cookies
//check if there are cookies
if(isset($_COOKIE)){
    foreach ($_COOKIE as $key => $value) {
        echo "$key : $value";
    }
}
```

Creating Cookies

- Use the function : setCookie('key' , 'value', lifespan)



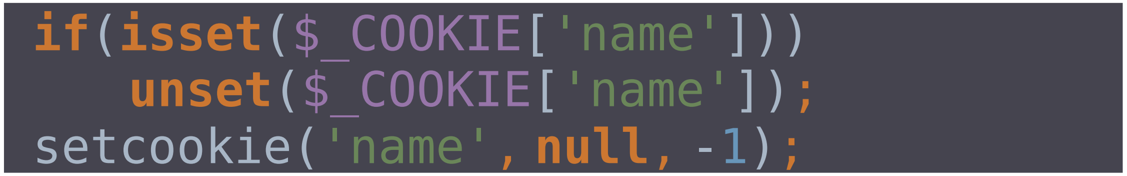
```
setcookie('name',$_POST['name'],time() + (60 * 60 *24 *365));
```

Deleting Cookies

Two ways to delete cookies:

- "re-set" the cookie with a negative life value
- delete the value in the $ _COOKIE super global with unset.

This makes sure that the value of the cookie will be used in the rest of the PHP code.



```
if(isset($_COOKIE['name']))
    unset($_COOKIE['name']);
setcookie('name', null, -1);
```

Sessions

- See session documentation here
- way to preserve data between multiple accesses by the same client.
- Each visitor accessing the web page is assigned a unique identifier, called "session ID". The session is stored either in a cookie (phpidsession) or passed in the URL.
- PHP will automatically check if there is a session of the same identifier (with session_start ())

$_SESSION

- Like $_COOKIE, $_SESSION is an associative array.But we are no longer limited to string.
- session_start() creates a session or restores the one found on the server, via the session ID passed in a GET, POST or cookie request.
- Must be first instruction before any other in all PHP files where we manipulate the session

$_SESSION

```php
<?php
// first instruction on the page
session_start();

//add information to the session
$_SESSION['userId'] = 45;

$arr = [
145 => 2,
"test" => 5,
"id" => "hello world", 45 => 4
];
//add an aray to a session
$_SESSION['array'] = $arr;
```

```php
<?php
//in another PHP page
//always first instruction
session_start();

//accessing values saved
$idUser = $_SESSION['userId'];
$arrSave = $_SESSION['array'];

//output
var_dump($idUser);
echo "<br />";
var_dump($arrSave);
```

Sending files

- attribute on the form

  enctype="multipart/form-data"

- post method for more size

- access the file via $ _FILE

- If everything is validated use move_uploades_file(str : fileToMove, str : destination)

file upload

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Title</title>
</head>
<body>
<form
action="file_management.php"
method="post"
enctype="multipart/formdata"
>
   <input type="file"
name="fichierToUpload" />
<br />
   <input type="submit"
value="Send" />
</form>
</body>
```

```
<?php
if (isset($_FILES['fileToUpload'])) {
   echo "<ul>";
   foreach ($_FILES['fileToUpload'] as $key => $value)
{
     echo "<li>$key : $value </li>";
   }
   echo "</ul>";
   $isGood = true;

   if($_FILES['fileToUpload']['size'] > 200000)
      $isGood = false;

   if (!strcmp($_FILES['fileToUpload']['type'], "image/*")) {
      $isGood = false;
   }

   if($isGood){
      echo "file good";
      move_uploaded_file($_FILES['fileToUpload']
['tmp_name'], '/img/test.jpg');

      echo '<img src="img/test.jpg" />';

   }else{
      echo "file not good no upload";
   }
}
```