Working with user input

PHP web development 2019/2020

Milena Tomova
Vratsa Software
https://vratsasoftware.com/

Table of Contents

ARRAY

ARRAY

Врача Софтуер Общество

- Масивът е променлива, която може да съхранява едновременно повече от една стойност.

```
$cars1 = "Volvo";
$cars2 = "BMW";
$cars3 = "Toyota";
```

този списък с променливи, може да бъде обединен в една - **масив $cars**

```
$cars = ["Volvo", "BMW", "Toyota"]
```

ARRAY

- Достъпваме елементите в масива чрез индекси, започвайки броенето от 0.

```
$cars = ["Volvo", "BMW", "Toyota"]
```

```
<?php echo $cars[0];?>//Volvo
<?php echo $cars[1];?>//BMW
<?php echo $cars[2];?>//Toyota
```

Не е задължително данните в масива да са от един и същи тип!


ASSOCIATIVE
ARRAY

ASSOCIATIVE ARRAY

Врача Софтуер Общество

- Асоциативните масиви са масиви, чиито индекси са стрингове. Наричаме ги ключове.

```
1//
$ages = ["Peter"=>"35", "Ben"=>"37", "Joe"=>"43"]
```

```
2//
$ages = [];
$ages['Peter'] = "35";
$ages['Ben'] = "37";
$ages['Joe'] = "43";
```

1// и 2// са начини да декларираме променлива масив и да зададем стойности на нейните елементи.

ARRAY

Врача Софтуер Общество

Опечатване на целия масив

```
print_r($ages)
```

Отпечатване на елемент от масива

```
echo $ages[‘Peter’]
```

Проверяваме броя на елементите на масива, стойността и типа им

```
var_dump($ages)
```

GLOBAL ARRAYS

GLOBAL ARRAYS

Врача Софтуер Общество

$_GET[]

```
<form action=“index.php” method=“get”>
```

$_POST[]

```
<form action=“index.php” method=“post”>
```

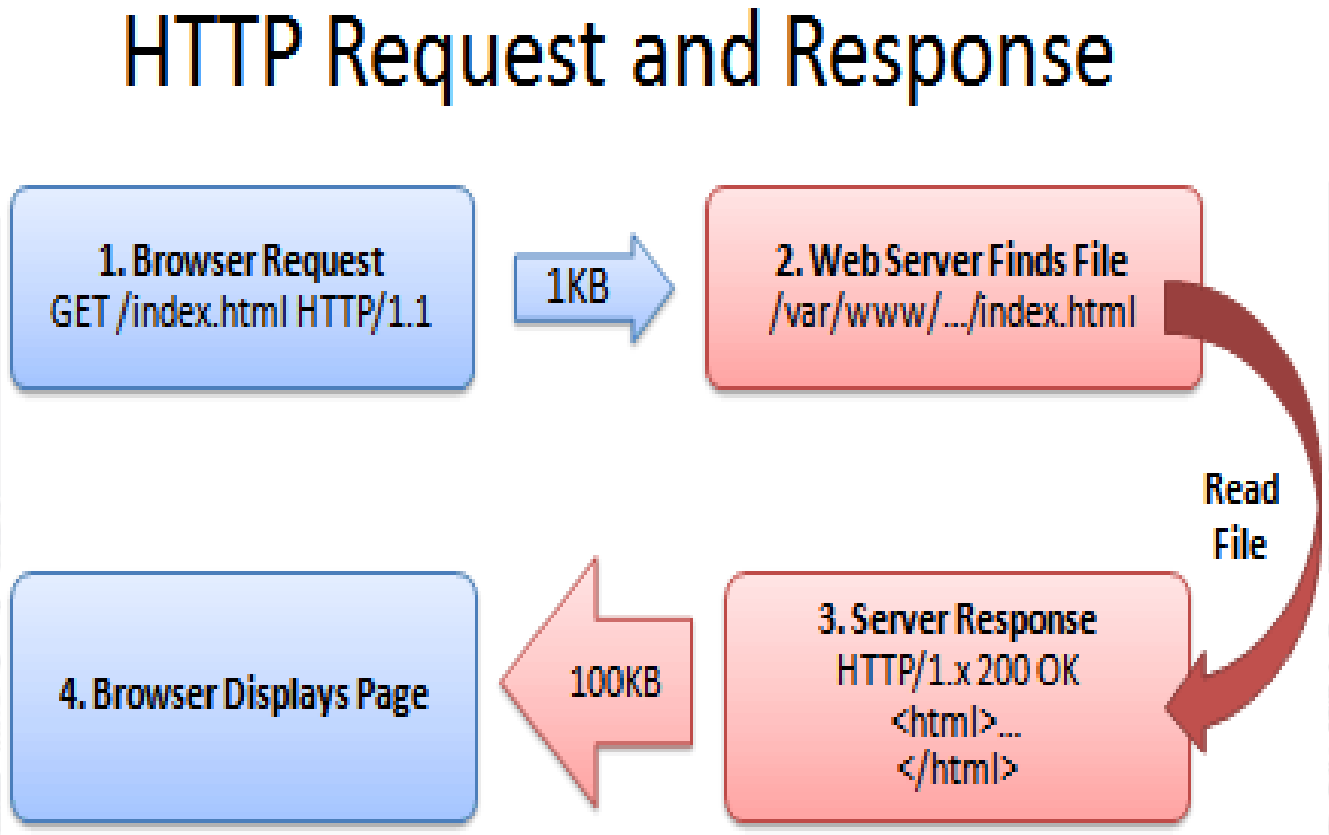
$_FILE[]

```
<form action="upload.php" method="post" enctype="multipart/form-data">
                                Select image to upload:
                                <input type="file" name="fileToUpload"
id="fileToUpload">
                               <input type="submit" value="Upload Image"
name="submit">
                    </form>
```


http

http

Работи с request & response
Stateless протокол, т.е. всяка заявка е независима от предходните




GET/POST

GET/POST

GET/POST

GET:

извличане от съществуващ ресурс чрез URL

POST:

създава нов ресурс

GET/POST

GET/POST

GET requests can be cached
POST requests are never cached

GET requests remain in the browser history
POST requests do not remain in the browser history

GET requests can be bookmarked
POST requests cannot be bookmarked

GET requests should never be used when dealing with sensitive data - passwords, credit-card numbers
Use POST requests instead

GET requests should be used only to retrieve data
GET requests have length restrictions - up to 250 signs
POST requests have no restrictions on data length

GET/POST

RESPONSE STATUS CODES

1xx: Informational Messages

2xx: Successful

3xx: Redirection

4xx: Client Error

5xx: Server Error

working with user input

working with user input

Врача Софтуер Общество

index.html

```
<form action="index.php" method="get">

    <input type="text" name="age" />
    ....
    <input type="submit" name="submit" value="Send" />

</form>
```

index.php

```
<?php $age = $_GET['age'];
```

action

form action="...."

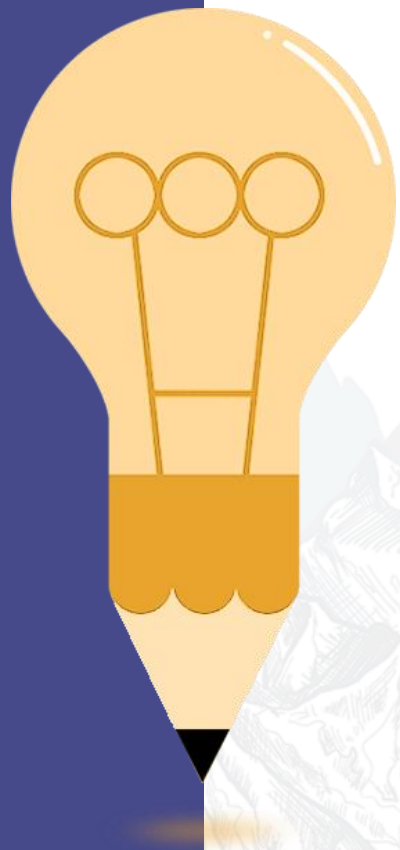
action = 'filename.php'

обработваме данните във filename.php

action = /няма зададена стойност/

обработваме данните в същия файл

задача



Добавете във форма полета от типа

input type = “text”

input type="hidden"

input type="radio"

input type = “checkbox”

Кои са задължителните атрибути на

- формата
- input полетата

Особености при атрибутите на

- input type="radio"
- input type = “checkbox”

Проверете какъв тип са данните, които получавате?

sessions and session handling

Sessions and Session Handling

$_SESSION

http заявките са stateless -

не запаметяват текущия статус в приложението

Ако преминете от един в друг скрипт - .php файл /изпълните зявка към браузъра/,

променливите и стйностите им

в първия файл /преди заявката/ не са видими във втория файл /след заявката/.

Sessions and Session Handling

$_SESSION

Сесията /session/

- позволява съхраняването и използването на информация между заявките в браузъра, записвайки я в

суперглобалния масив $_SESSION

- масив, който може да бъде достъпван във всеки файл от проекта ви.

Sessions and Session Handling

$_SESSION

Когато потребителят достъпи даден скрипт /отправи заявка към сървъра/,

PHP проверява дали

- е зададено автоматично стартиране на сесия

/в php.ini session.auto_start = 1/

- в скриптовете изрично е указано стартиране на сесията - в самото начало на скрипта с функцията session_start().

И в двата случая сесията получава уникално id, което се проверява от браузъра и служи за неин идентификатор

Sessions and Session Handling

приложение

- съобщения при успешно/неуспешно действие на потребителя
- запазване потребителското име на влезлия в приложението потребител
- други данни, за които да не отправяме заявка към базата данни, което изисква ресурс или не са записани /все още/ в базата данни или няма необходимост да ги записваме, но са необходими за да функционира приложението.

Sessions ....

Врача Софтуер Общество

Създаване на сесия

```
<?php

session_start();
```

Създаваме променливите /session variables/, които искаме да запазим и използваме при преминаване от страница в страница на приложението ни.

```
$_SESSION[“username"] = $_POST[‘username’];

$_SESSION[“email"] = $_POST[‘email’];
```

Sessions ....

Врача Софтуер Общество

Определяме колко време да се пазят данните в сесията.

```
session_set_cookie_params('600'); // 10 minutes
session_start();
```

/Set cookie parameters defined in the php.ini file. The effect of this function only lasts for the duration of the script. Thus, you need to call **session_set_cookie_params()** for every request and before session_start() is called. This function updates the runtime ini values of the corresponding PHP ini configuration keys which can be retrieved with the ini_get()./

Sessions and Session Handling

Продължителност на сесията

Ако няма друго указание, продължиелността на сесията е до затваряне на браузъра.

Sessions and Session Handling

Изтриване на записаните в сесията променливи

```
// remove all session variables
session_unset();

// remove all success message
session_unset($_SESSION['succes']);
```

Sessions and Session Handling

Унищожаване на сесията

```
session_destroy();
```

Натискаме бутона за излизане от приложението. Цялата сесия, заедно с декларираните в нея променливи се унищожават. При влизане в системата, дори и без да е затварян браузъра, се създава нова сесия, с нов id.

Sessions and Session Handling

session_unset or session_destroy

session_unset just clears out the session for usage. The session is still on the users computer. Note that by using session_unset, the variable still exists. session_unset just remove all session variables. it does not destroy the session....so the session would still be active.

Using session_unset in tandem with session_destroy however, is a much more effective means of actually clearing out data. As stated in the example above, this works very well, cross browser. session_destroy destroys the session. session_destroy() to kill all session information.....This is the more secure function to use when you want to destroy the session with all of its data.

Questions?


Враца Софтуер Общество
Гнездото Coworking
Цялостен курс по програми ране
Дизайн курс
Курс по дигит. маркети нг
MindHub
CodeWeek Враца
HACK VRATSA
#кодаправиш

Partners

Враца Софтуер Общество

AMERICA FOR BULGARIA FOUNDATION

ВРАЦА

Telerik Academy

НАЦИОНАЛНА МРЕЖА ЗА ДЕЦАТА

MindHub

ПРОМЯНАТА

Trainings @ Vratsa Software

- Vratsa Software – High-Quality Education, Profession and Jobs
  - www.vratsasoftware.com
- The Nest Coworking
  - www.nest.bg
- Vratsa Software @ Facebook
  - www.fb.com/VratsaSoftware
- Slack Channel
  - www.vso.slack.com