



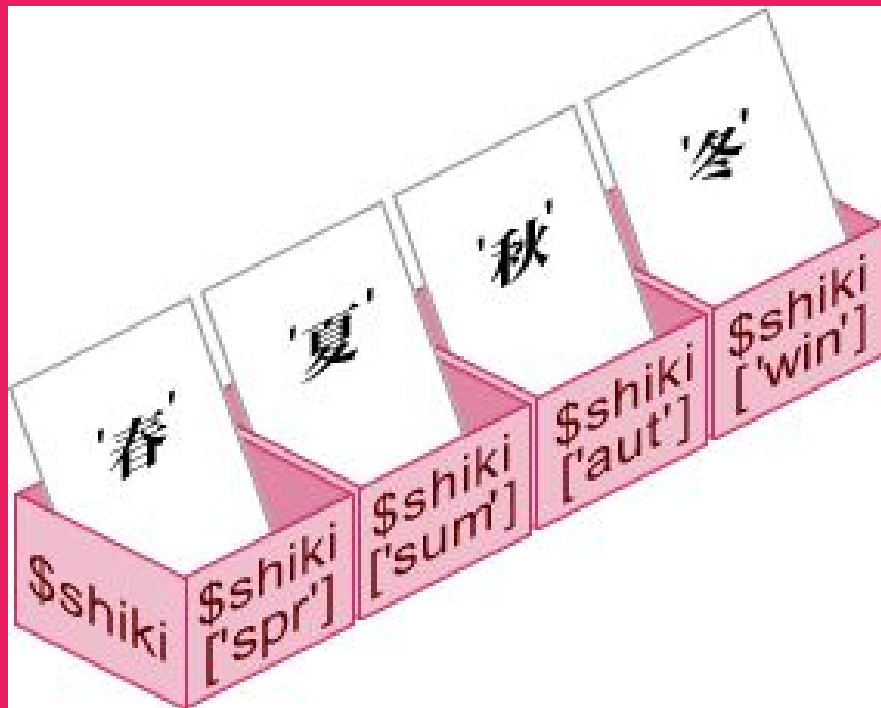
**WEB** разработка

**PHP** работа с форми

# Съдържание

- Масиви
- HTTP протокол
- GET
- POST
- GET vs POST
- user input
- PHP сесии

# Асоциативни масиви



# Масиви

**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 = array("Volvo", "BMW", "Toyota")
```

# елементи на масива

`$cars = array("Volvo", "BMW", "Toyota")` - масив с три елемента.

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

елемент с индекс **0** - "Volvo" - `$cars[0]`

елемент с индекс **1** - "BMW" - `$cars[1]`

елемент с индекс **2** - "Toyota" - `$cars[2]`

```
<?php echo $cars[0];?> //"Volvo"
```

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

# Асоциативни масиви

Асоциативните масиви са масиви, чиито индекси са имена/стрингове (ключ).

В декларацията за масива изреждаме **ключ => стойност** за всеки елемент от масива.

```
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
```

или

```
$age['Peter'] = "35";
```

```
$age['Ben'] = "37";
```

```
$age['Joe'] = "43";
```

# Асоциативни масиви - 2

```
$user_info = array("username"=>"Peter22", "password"=>"33237", "first_name"  
=>"Peter", "second_name" => "Petrov", "age"=>35);
```

```
$user_info = array("username"=>"Peter22",  
                  "password"=>"33237",  
                  "first_name"=>"Peter",  
                  "second_name" => "Petrov",  
                  "age"=>35);
```

# Асоциативни масиви - 3

```
$user_info ["username"] = "Peter22";  
$user_info ["password"] = "33237";  
$user_info ["first_name"] = "Peter";  
$user_info ["second_name"] = "Petrov";  
$user_info ["age"] = 35;
```

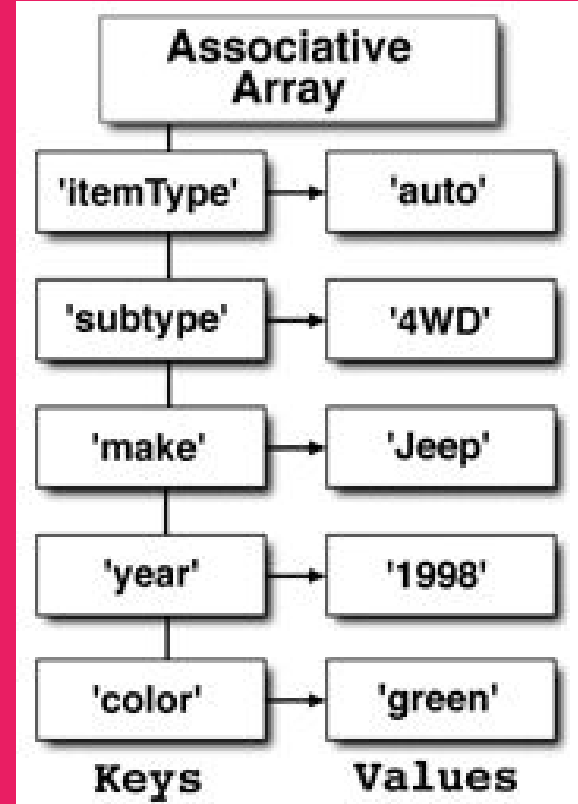
Опечатване на целия масив - `print_r($user_info)`

Отпечатване на елемент от масива `echo $user_info ["age"]`

В процеса на работа достъпваме масива с `var_dump($user_info)`



# Асоціативні масив демо задачи

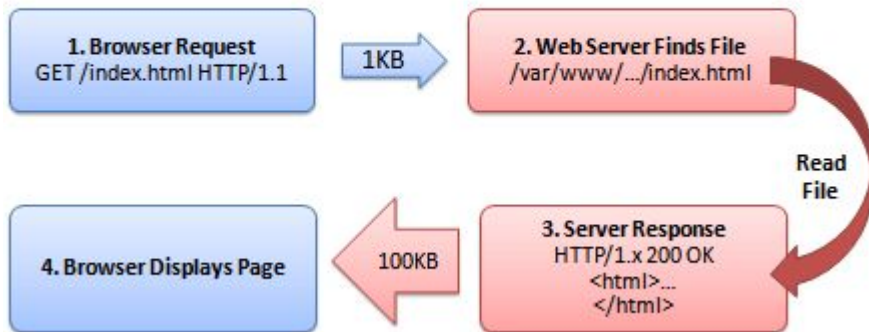


# http протокол

# http протокол

Работи с request & response

## HTTP Request and Response



Stateless протокол, т.е. всяка заявка е независима от предходните

# http протокол - 2

Действия: GET, POST

GET: извличане от съществуващ ресурс чрез URL

POST: създава нов ресурс

Статус кодове: 1xx: Informational Messages

2xx: Successful

3xx: Redirection

4xx: Client Error

5xx: Server Error

# работа с формы

```
<form method="post"  
      action="action.php">  
<input type="text"  
      name="user_name">  
...  
</form>
```



PHP \$\_POST & \$\_GET  
Superglobals

The Basics of HTTP Interaction

By - Intechgrity.com

# работа с форми

html

```
<form action="index.php" method="get">  
  <input type="text" name="age" />  
  ....  
  <input type="submit" name="submit" value="Send" />  
</form>
```

php

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

```
    /*след което използваме $age в логиката на  
    някаква задача*/  
?>
```

# работа с форми - 2

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

с `action="new_file.php"` - отвеждаме към нов рnr файл или

`action="same_file.php"` - оставаме в същия файл за изпълнение на  
някаква логика/код/

# работа с формами - get vs. post

<form action="file.php" method="get">

vs.

<form action="file.php" method="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



работа с форми  
демо + задачи



# Данните, постъпващи с формата ... /user input/

What are you?

- a.  Rockstar
- b.  Ninja
- c.  Guru
- d.  None of the above



**2016**  
DEVELOPER  
SURVEY

# user input

Видове полета за въвеждане на данни от потребителя и как “събираме” и работим с тези данни?

input type = “text”

input type="hidden"

input type="radio"

input type = “checkbox” - A checkbox is only submitted if it's actually checked

/пробвай кога какво се получава като данни/

# user input

```
<form action="index.php" method="get">  
<input type="text" name="age" />  
<input type="submit" name="submit" value="Send" />  
</form>
```

```
<?php $age = $_GET['age'];  
    if ($age < 18) {  
        echo "You are not allowed here!";  
    } else {  
        echo "Welcome, user!";  
    }  
?>
```

User input  
демо + задачи



# PHP сесии

## Sessions

- php.ini – session.save\_path
- Maximum size of a PHP session  
– 128 M
- session\_start()
- isset()
- unset()
- session\_destroy()

# PHP сесии

PHP сесии (\$\_SESSION)

\$\_SESSION е 'суперглобална' променлива или автоматично глобална променлива. Казано по-просто - може да я достъпвате навсякъде в кода си.

Съхранява информация през последователност от много страници на нашето уеб приложение.

*/\*Sessions preserve data between different HTTP requests*

*Implemented through cookies*

*\$\_SESSION is an global array holding the session variables*

*After session\_start() it is stored on the HDD\*/*

# PHP сесии - 2

Създаване на сесия

```
<?php
```

```
// Start the session
```

```
session_start();
```

```
// Set session variables
```

```
$_SESSION["username"] = $_POST['username'];
```

```
$_SESSION["egn"] = $_POST['egn'];
```



# PHP сесии - продължителност

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\(\)](#)./

# PHP сесии

- Унищожаване на сесията.

-

```
<?php
```

```
// remove all session variables
```

```
session_unset();
```

```
// destroy the session
```

```
session_destroy();
```

```
?>
```

# Сесии демо



`$_SESSION`