Multimedia im Netz

Online Multimedia

Winter semester 2015/16

Tutorial 14 – Minor Subject

Today's Agenda

- Repetition
  - AJAX
  - PHP + MySQL
  - Theory
- Q&A

AJAX

AJAX

- Asynchronous Javascript and XML.
- Data-exchange without page refresh
- Usage:
  - Search engines
  - Web-apps (e.g. MS Office online)
  - Search-functionality in websites (e.g. Netflix)
- XML is now more and more replaced by JSON
  - Smaller file size
  - Easy usage in JavaScript

AJAX with jQuery

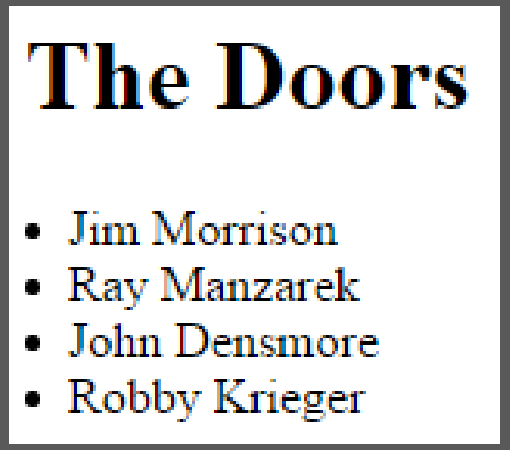
- Generic AJAX method: `$.ajax()`
- GET: `$.get()`
  - Parameters are attached to URL
  - Example:

```
$.get('01-thedoors.json', function (data) {
    // do something with data.
});
```

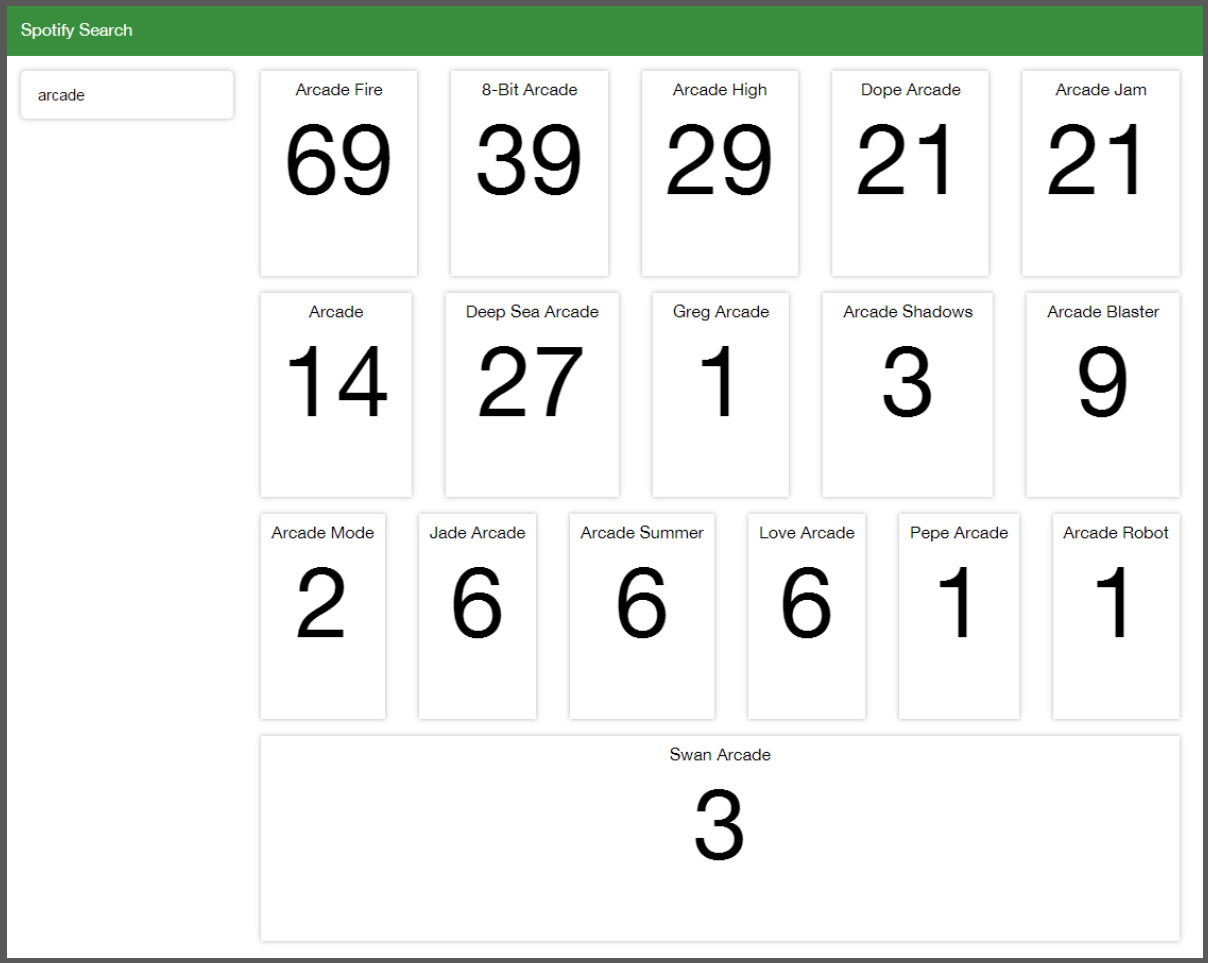
  - Use this to **retrieve** information from the server
- POST: `$.post()`
  - Parameters are transmitted in HTTP message body (less visible)
  - Use this to **modify** things on the server.

Breakout: Handling Data

- Extend breakout/01-ajax.html
- Load the file 01-thedoors.json with an AJAX request.
- Create a heading containing the band name
- Create a list containing the band members
  - $(…).each() is helpful here.
- Test if your solution still works if you use another JSON file.
  We provided 01-arcadefire.json



Assignment 07: Spotify Search Code-Along



PHP + MySQL

PHP

- Server side scripting language
- Usually interpreted by an Apache server with a PHP module
- Famous services that use PHP:
  - Wordpress
  - Facebook
  - MailChimp
  - … and many many others that you use each day.

- Scripts can be embedded into HTML (but are evaluated by the server, before the files are sent back to the server)

Server Script ➔ Generated HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
<?php
$name = "Martina";
echo
"<h2>".$name."</h2>";
?>
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
<h2>Martina</h2>
</body>
</html>
```

Sessions

- Sessions maintain “states” on the **server side**
- Sessions store current state of variables as long as connected to the client
- On the client side, sessions are identified with a **session ID cookie**:
  - default cookie name in PHP: PHPSESSID
  - renaming possible with session_name()

Sessions with PHP

- Sessions need to be started **before any output occurs**
- Create session ID cookie:
  `session_start()`
- Delete the session ID cookie:
  `session_destroy()`
- Read / write session values:
  - superglobal `$_SESSION` array
  - immediately reset session like this `$_SESSION = array();`

PHP Example: “Add my number” What’s the problem?

```
<!DOCTYPE html>
<html><head lang="en"><meta charset="UTF-8">
    <title></title></head><body>
<?php
if(isset($_POST['add'])) {
    $currentCounter = 0;
    echo "$currentCounter + " . $_POST['add']
         . " = " . ($currentCounter + $_POST['add']);

    $currentCounter += $_POST['add'];
}
?>
<form method="POST" action="add-my-number.php">
    <input type="number" placeholder=""
           name="add">
    <input type="submit" name="submit" value="Add">
</form>
</body></html>
```

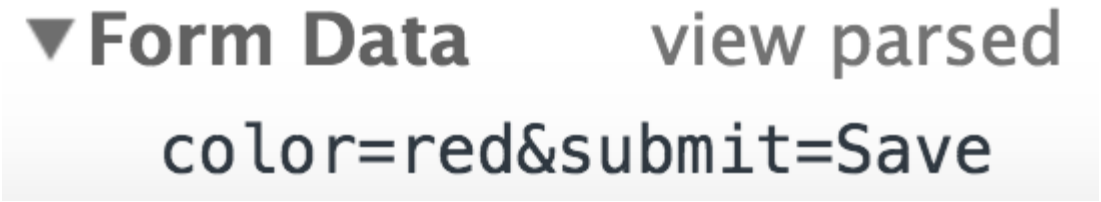
Breakout: Code-Along PHP

- Fix the code from the previous slide.
- When the form is submitted, the number from the input field should be added to the previous result.
- Timeframe: 10 minutes

GET & POST in PHP

- GET
  - Query string is sent in the URL of the request: http://localhost/test.php?lecture=onlineMultimedia
  - Parameters are visible to the user!
  - Superglobal variable in PHP: `$_GET` (Associative Array!)

- POST
  - Query string is sent in the HTTP message body of the request
  - Superglobal variable in PHP: `$_POST` (Associative Array!)



Example: Add an entry

```
INSERT INTO Contacts
   (FirstName, LastName, PhoneNumber)
VALUES
   ("Max", "Mustermann", 089455544431);
```

Table: Contacts

PersonID	FirstName	LastName	PhoneNumber
1	Max	Mustermann	089455544431

Example: Retrieve data

- Retrieve all data from a table
  `SELECT * FROM Contacts`

- Retrieve entries that fulfill a certain condition:
  `SELECT * FROM Contacts WHERE FirstName="Laura";`

Table: Contacts

PersonID	FirstName	LastName	PhoneNumber
1	Max	Mustermann	089455544431
2	Laura	Stern	070815643593
3	Tanja	Baumann	0895673138
4	Felix	Maurer	0894562897

Mysqli (procedural)

- Establish connection (replace user, password, and mydb with your own credentials)
```
$c = mysqli_connect("localhost", "user", "password", "mydb");
```
- Select database
```
mysqli_select_db($c,"mydb");
```
- Close connection
```
mysqli_close($c);
```
- PHP statement for MySQL query
```
$results = mysqli_query($c, $query);
```
- Process the results:
```
mysqli_fetch_array($results);
mysqli_fetch_array($results, MYSQLI_NUM);

mysqli_fetch_array($results, MYSQLI_ASSOC);
mysqli_fetch_assoc($results);
```

mysqli Example

id (INT PRIMARY KEY AUTO_INCREMENT)	amount (FLOAT NOT NULL)	reason (VARCHAR NOT NULL)	person (VARCHAR NOT NULL)	spending_date (DATE NOT NULL)

```php
<?php
$host = 'localhost';
$user = 'mmn1516';
$password = 'mmnpassword';
$database = 'mmn1516';
$c = mysqli_connect($host,$user,$password,$database);

$now = date('Y-m-d H:i:s');
$queryString = "INSERT INTO expenses
                        (amount,reason,person,spending_date)
                 VALUES (25,'coffee beans','Max','$now')";

mysqli_query($c,$queryString);

$queryString = "SELECT * FROM expenses";
$results = mysqli_query($c,$queryString);

while($row = mysqli_fetch_assoc($results)){
  echo
      $row['id'].' '.
      $row['amount'].' '.
      $row['reason'].' '.
      $row['person'].' '.
      $row['spending_date'].'<br />';
}
?>
```

Examples/02-php-mysql.php

Theory from Assignments

Lazy Loading

- Design Pattern (not only on the web)
- Common use-case on the web: placeholders that are replaced with the actual images
- Advantages
  - Web-site content becomes visible/accessible faster
  - Traffic can be reduced
- Disadvantages
  - Number of requests can rapidly increase
  - Difficult to cache / bookmark
- Where do you see lazy loading every day?

https://en.wikipedia.org/wiki/Lazy_loading

Watermarking: Characterization Task

EXIF information in JPEG file

- Visibility: **not in image directly**. EXIF tool necessary.
- Universality: **Depends**. Images from the same camera will all have the same EXIF information regarding the camera model / vendor.
- Detectability: **High**. File explorer shows data with a mouse click.
- Robustness: **Low**. Printing and scanning destroys watermark.
- Capacity: **Medium**. There are many different fields. But difficult to store “rich” information (e.g. logos) in EXIF info
- Security: **Low**. EXIF information can easily be changed.
- Efficiency: **High**. There is little overhead with inserting the data.

Thanks for joining!

Good luck for the exam!