

# **Multimedia im Netz**

## Wintersemester 2012/13

### Übung 05

# Lösung zu Übungsblatt 03

# Movie-Player mit Untertitel (HTML5 + JavaScript)

**Video with subtitles**

Video Source: <http://www.bigbuckbunny.org/>



This is the story of the big buck bunny.

pause  00:27 / 09:56

**Add Subtitles**

clock  add

---

Timestamp	Subtitle
00:26	This is the story of the big buck bunny.
00:51	Waking up in the morning can be quite annoying.

# Übungsblatt 04

- **Thema: AJAX**
- **Abgabe: 05.12.2012; 11:00 Uhr**

Nickname: <input type="text"/>	Shout: <input type="text"/>	<input type="button" value="Shout"/>
<p>[Mon, 26 Nov 2012 14:42:30] Tanja shouts 'Das ist ein MMN-Uebungsblatt.'</p> <p>[Mon, 26 Nov 2012 14:42:51] Hans shouts 'Ja, es geht dabei um AJAX.'</p> <p>[Mon, 26 Nov 2012 14:43:10] Tanja shouts 'Deadline ist nächste Woche Mittwoch.'</p> <p>[Mon, 26 Nov 2012 14:43:18] John shouts 'Bla Bla Bla'</p>		

# AJAX

- Asynchronous JavaScript and XML
- **Vorteil:** Seite muss nicht neu geladen werden
- JavaScript wird benutzt um im Browser eine Anfrage zum Server zu stellen
- Wird Anfrage beantwortet wird eine Funktion beim Client ausgeführt
- Auf Serverseite können verschiedene Technologien verwendet werden (z.B. Ruby, PHP, CGI,...)

# XMLHttpRequest-Objekt

- zentrale Komponente von AJAX
- kein W3C Standard, sodass Verhalten von Browser zu Browser unterschiedlich sein kann
- um Anfragen zu senden oder Antworten zu bearbeiten, muss mit JavaScript zuerst ein XMLHttpRequest-Objekt erstellt werden

# XMLHttpRequest-Objekt erstellen

- ActiveX Objekt im Internet Explorer (Version 6)

```
var xhr = new ActiveXObject("Microsoft.XMLHTTP");
```

- natives Objekt in Mozilla, Opera, Safari & Internet Explorer (ab Version 7)

```
var xhr = new XMLHttpRequest();
```

# XMLHttpRequest-Objekt: Zustand

- `onreadystatechange`: Reaktion auf Zustandsänderung des Objekts
- `readyState`: Zustand des Objekts  
0 (nicht initialisiert), 1 (ladend), 2 (geladen), 3 (wartet), 4 (fertig)
- Beispiel:

```
xhr.onreadystatechange = myFunction;  
function myFunction() {  
    if (xhr.readyState == 4) {...}  
}
```

# XMLHttpRequest-Objekt: Methoden

- Einen Request an den Server senden  
`open(method, url, async);`  
`send(string);`
- `method`: GET oder POST
- `URL`: Location des Quellcodes auf dem Server
- `async`: true (asynchronous) oder false (synchronous)
- `string`: nur bei POST-Requests benötigt
- Die Antwort des Servers als String abfragen  
`xhr.responseText;`

# XMLHttpRequest: Beispiel

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body>
    <input type="button" value="AJAX Request"
          onclick="sendRequest();" />
    <div id="output"></div>
  </body>
</html>
```

# XMLHttpRequest: Beispiel

```
<head>
```

```
  <title>Example</title>
```

```
  <script type="text/javascript">
```

```
    var xhr;
```

```
    if(window.ActiveXObject) {
```

```
      xhr = new ActiveXObject(Microsoft.XMLHTTP);
```

```
    } else if(window.XMLHttpRequest) {
```

```
      xhr = new XMLHttpRequest();
```

```
    }
```

```
    ...
```

```
  </script>
```

```
</head>
```

# XMLHttpRequest: Beispiel

```
<head>
  <title>Example</title>
  <script type="text/javascript">
  ...
  function sendRequest() {
    xhr.onreadystatechange = output;
    xhr.open( "GET", "text.txt" );
    xhr.send(null);
  }
  </script>
</head>
```

# XMLHttpRequest: Beispiel

```
function output() {  
    if(xhr.readyState == 4) {  
        text = xhr.responseText;  
        document.getElementById("output").innerHTML=text;  
    }  
}
```