

ÜBUNG ZUR VORLESUNG
MULTIMEDIA IM NETZ

Ludwig-Maximilians-Universität
Wintersemester 2010/2011

ÜBUNGSBLATT - 3

- Thema: JavaScript / AJAX
- Anzeigen von Filminformationen, wenn der Benutzer über ein Filmcover fährt.
- Die Information soll über einen XMLHttpRequest aus einer XML-Datei ausgelesen werden
- Die Abgabe muss nicht in allen Browsern laufen (Firefox reicht!)

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
- Server schickt meist – aber nicht zwingend – XML Dokumente zurück
- Auf Serverseite können verschiedene Technologien verwendet werden (z.B. Ruby, PHP, CGI,...)

XMLHttpRequest (1)

- zentrale Komponente von AJAX
- es ist kein W3C Standard, sodass das 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:
 - Internet-Explorer: ActiveX Object
 - Firefox, Safari, Opera: JavaScript Object

XMLHttpRequest (2)

- ActiveX Objekt im Internet Explorer (Version 5)
`var xhr = new ActiveXObject("Msxml2.XMLHTTP");`
- 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 (3)

```
var xhr;

if (window.ActiveXObject) {
    xhr = new ActiveXObject("Microsoft.XMLHTTP");
} else if (window.XMLHttpRequest) {
    xhr = new XMLHttpRequest();
}
```

METHODEN

- ein paar wichtige Methoden für das XMLHttpRequest Object:

- `open("method", "url", flag)`

method: z.B.: GET oder POST

url: Pfad zur Datei

flag: true = asynchron

false = synchron

- `send(content)`

EIGENSCHAFTEN

- **onreadystatechange**: Reaktion auf jede Zustandsänderung des HTTP-XML-Objekts
- **readyState**: Zustand der Anfrage
 - 0 = nicht initialisiert
 - 1 = ladend
 - 2 = geladen
 - 3 = wartet
 - 4 = fertig
- **responseText**: Die Antwort des Servers als String
- **responseXML**: Die Antwort des Servers als XML/ als DOM kompatibles Dokument

DOCUMENT OBJECT MODEL (DOM)

- Mit DOM kann jedes Element und dessen Inhalt in einem HTML (und XML) Dokument referenziert werden
- Die Elemente, ihr Inhalt und ihre Struktur kann modifiziert werden

document: Inhalt der im Browserfenster angezeigt wird

getElementById(): Greift auf das HTML-Element mit entsprechender ID zu

getElementsByTagName(): Greift auf Tags anhand ihres Namen zu

Knoten.firstChild : Liefert den ersten Kindknoten des angegebenen Knoten

Knoten.nodeValue: Setzt/Liefert den Wert eines Knotens

- Siehe auch: <http://de.selfhtml.org/javascript/index.htm>

Beispiel

Auslesen von Text aus einer txt-Datei wenn auf einen Button geklickt wird.

BEISPIEL (1)

```
<html>

  <head>

    <title>Example</title>

  </head>

  <body>

    <input type="button" value="AJAX Request" onclick="sendRequest();" />

    <div id="output"></div>

  </body>

</html>
```

BEISPIEL (2)

```
<html>

  <head>

    <title>Example</title>

    <script type="text/javascript">
      function sendRequest() {

        createXHR();

        xhr.onreadystatechange = output;

        xhr.open("GET", "text.txt");

        xhr.send(null);

      }
    </script>
```

...

BEISPIEL (3)

```
<html>
```

```
<head>
```

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

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

```
function sendRequest() {
```

```
    createXHR();
```

```
    xhr.onreadystatechange = ou
```

```
    xhr.open("GET", "text.txt")
```

```
    xhr.send(null);
```

```
}
```

```
</script>
```

```
function createXHR() {
    if(window.ActiveXObject) {
        xhr = new ActiveXObject(Microsoft.XMLHTTP);
    } else if(window.XMLHttpRequest) {
        xhr = new XMLHttpRequest();
    }
}
```

...

BEISPIEL (4)

```
<html>

  <head>

    <title>Example</title>

    <script type="text/javascript">
      function sendRequest() {

        createXHR();

        xhr.onreadystatechange = output;

        xhr.open("GET", "text.txt");

        xhr.send(null);

      }
    </script>
```

...

BEISPIEL (5)

```
<html>
```

```
<head>
```

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

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

```
    createXHR();
```

```
    xhr.onreadystatechange = output;
```

```
    xhr.open("GET", "text.txt");
```

```
    xhr.send(null);
```

```
}
```

```
</script>
```

...

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

AUFGABE

- Laden Sie sich die XML-Datei news.xml von der Vorlesungswebseite herunter
- Lesen Sie die Schlagzeilen in der XML-Datei mittels eines XMLHttpRequest aus und zeigen diese im Browser an.
- Hinweis:
Text zwischen Tags wird als DOM-Element betrachtet (Textknoten) und ist Kindknoten des umgebenden Tags.