

# Proseminar „Auszeichnungssprachen“

---

Thema: **AJAX**

Ayşe Gül Kılıç

Ahmed Anis Kerkeni

Dortmund, 29.05.2007

# [ Gliederung ]

---

- Definition

# [ Gliederung ]

---

- Definition
- Geschichte

# [ Gliederung ]

---

- **Definition**
- **Geschichte**
- **Konzept**

# [ Gliederung ]

---

- **Definition**
- **Geschichte**
- **Konzept**
- **Anwendungen und Techniken**

# [ Gliederung ]

---

- **Definition**
- **Geschichte**
- **Konzept**
- **Anwendungen und Techniken**
- **Beispiele**

# [ Gliederung ]

---

- **Definition**
- **Geschichte**
- **Konzept**
- **Anwendungen und Techniken**
- **Beispiele**
- **XMLHttpRequest**

# [ Gliederung ]

---

- **Definition**
- **Geschichte**
- **Konzept**
- **Anwendungen und Techniken**
- **Beispiele**
- **XMLHttpRequest**
- **Schemata**



# [ Gliederung ]

---

- **Definition**
- **Geschichte**
- **Konzept**
- **Anwendungen und Techniken**
- **Beispiele**
- **XMLHttpRequest**
- **Schemata**
- **Kritik: Vor- und Nachteile**

# [ Gliederung ]

---

- **Definition**
- **Geschichte**
- **Konzept**
- **Anwendungen und Techniken**
- **Beispiele**
- **XMLHttpRequest**
- **Schemata**
- **Kritik: Vor- und Nachteile**
- **Fazit**

# [ Gliederung ]

---

- **Definition**
- **Geschichte**
- **Konzept**
- **Anwendungen und Techniken**
- **Beispiele**
- **XMLHttpRequest**
- **Schemata**
- **Kritik: Vor- und Nachteile**
- **Fazit**
- **Literaturliste**

# Definition

- Ajax (**A**synchronous **J**avaScript **A**nd **X**ML)
- AJAX ist ein Begriff der mehrere Technologien zusammen fasst
- AJAX ersetzt Seitenelemente ohne diese neu zu laden

# [ Geschichte ]

---

- Der Begriff AJAX stammt von Jesse James Garrett, Berater bei der amerikanischen Firma AdaptivePath.
- Er veröffentlichte im Februar 2005 einen Artikel, in dem er den Begriff zum ersten Mal eingeführt hat.

# Konzept (Klassische Webanwendung)

- Client sendet aus dem Browser eine Anfrage zum Webserver
- Angefragter Webserver fragt Datenbank oder andere Komponente an
- Neue HTML-Seite entsteht
- HTML-Seite wird komplett zum angefragten Browser versendet

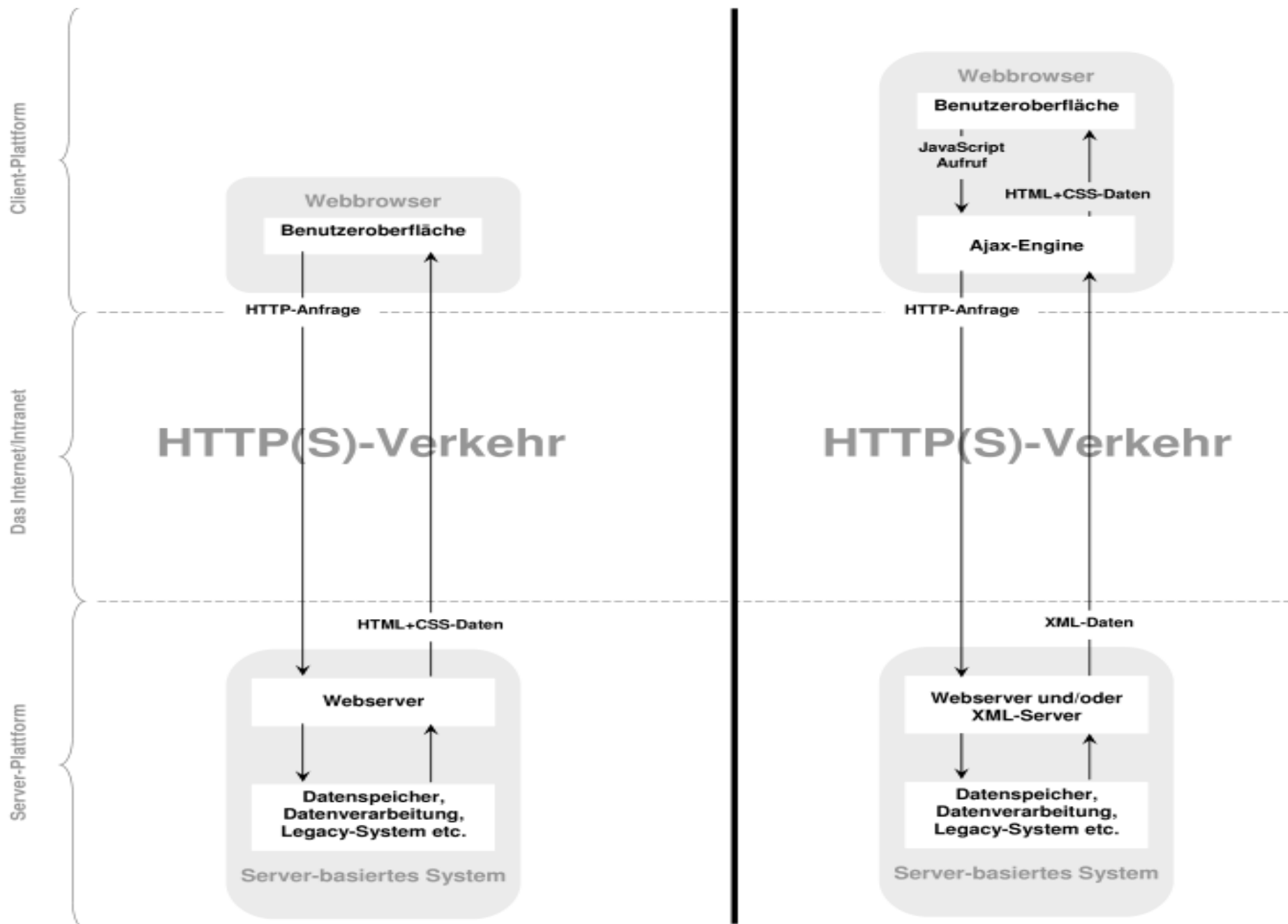
# [ Konzept (AJAX) ]

---

- Ajax wird durch ein JavaScript-Code präsentiert
- XMLHttpRequest-Objekt versendet eine asynchrone Anfrage zum Server
- Antwort folgt im XML-Format
- Anzeige im Browser aktualisiert sich automatisch

## Klassisches Modell einer Web-Anwendung

## Ajax Modell einer Web-Anwendung





# Anwendungen und Techniken

Eine Ajax-Anwendung basiert auf folgenden Web-Techniken:

- \* **HTML** (oder **XHTML**) und **CSS** werden verwendet um die Webseite beeinflussen zu können

- \* **Document Object Model** ist für die Dynamische Anzeige und Dateninteraktion da.

# Anwendungen und Techniken

- \* **JavaScript** verwendet man zur Manipulation des “Document Object Models” und zur dynamischen Darstellung der Inhalte.
- \* **XML** verwendet man zum Datenaustauschformat
- \* **XSLT** dient zur Datentransformation
- \* Das **XMLHttpRequest-Objekt** ist Bestandteil vieler Browser um Daten auf asynchroner Basis mit dem Webserver austauschen zu können.

# [ Beispiele ]

---

- [http://www.goweb.de/was\\_ist\\_ajax.htm](http://www.goweb.de/was_ist_ajax.htm)

Diese Beispielanwendung holt sich alle 60 Sekunden die aktuellen Systeminfos vom Server und stellt diese dar.

Die Seite wird nicht neu geladen. Nur der Inhalt des weissen Feldes wird ausgetauscht mit den jeweiligen Daten vom Server.



Suche ▾



PageRank ABC



Rechtschreibprüfung ▾



Abonnieren ▾



ipt

**AJAX** wird sicher primär für Applikationen als für normale Webseiten verwendet, weil **AJAX** den Nachteil hat, dass Suchmaschinen - die ja nunmal kein Javascript ausführen - den Inhalt der von einer **AJAX** Applikation erzeugt wird, nicht sehen.

tellen

Diese Beispielanwendung holt sich alle 60 Sekunden die aktuellen Systeminfos vom Server und stellt diese hier dar:

```
Prozesse: 285 sleeping, 8 running, 0 stopped, 0 zombie
Systembelastung: 59.2% User, 8.7% System, 0.0% Nice, 32.0% Idle
Speicher: 506596K verfügbar, 498248K benutzt, 8348K frei, 71556K gebuffert
```

q

Die Seite wird nicht neu geladen. Nur der Inhalt des weissen Feldes wird ausgetauscht mit den jeweiligen Daten vom Server.

Das ist natürlich nur eine Beispielanwendung. Man kann jede Art von **XML Datei** verarbeiten und innerhalb einer Seite anzeigen.

# Beispiel Bild 1

Google - Mozilla Firefox

File Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe

http://www.google.com/webhp?complete=1&hl=en Go

Personalized Home | Sign in

# Google™

Suggest LABS

Web Images Video<sup>New!</sup> News Maps more »

javascript	
javascript	50,200,000 results
javascript tutorial	6,100,000 results
javascript reference	7,880,000 results
javascripts	1,530,000 results
javascript array	1,500,000 results
javascript alert	2,230,000 results
javascript window.open	526,000 results
javascript redirect	557,000 results
javascript substring	248,000 results
javascript tutorials	4,660,000 results

Advanced Search  
Preferences  
Language Tools

As you type, Google results. [Learn more](#)

Fertig


# Beispiel Bild 2

1



# [XMLHttpRequest ()]

- Der Xml Http Request wird verwendet um mittels Javascript Http Requests an Webserver zu verschicken.
- Er ist ein wichtiger Bestandteil für das Web-Remoting in Ajax-Anwendungen.
- Nach dem man eine Variable dem XML Objekt zugewiesen hat, können alle weiteren Anfragen über gleiche Routinen erfolgen.

- 
- Alle modernen Browser unterstützen die **XMLHttpRequest()** Da diese Klasse nicht standardisiert ist, ist die Initialisierung bei den Browsern unterschiedlich.
  - Im Microsoft Internet Explorer nennt sie sich **ActiveXObject("Msxml2.XMLHTTP.4.0")**  
**ActiveXObject("Microsoft.XMLHTTP")** und zwar je nach IE Version.
  - **Initialisierung:**  
(nicht MSIE)  
var xobj = new XMLHttpRequest();  
(Microsoft Internet Explorer)  
var xobj = new  
ActiveXObject("Msxml2.XMLHTTP.4.0");



- Dank der bedingten Kompilierung in JScript geht folgender Code in beiden Browsern:

```
var xmlhttp=false;
/* @cc_on @*/
/* @if (@_jscript_version >= 5)
try {
  xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
  try {
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
  } catch (E) {
    xmlhttp = false;
  }
}
@end @*/
if (!xmlhttp && typeof XMLHttpRequest!='undefined') {
xmlhttp = new XMLHttpRequest();
}
```

# Schemata

## Ein AJAX Beispiel:

- **Dieses Beispiel zeigt, wie man den Inhalt einer Seite anfordert und auf der eigenen Seite das Ergebnis sehen kann. Die Anfrage wird durch ein Maus Klick ausgeführt.**

[ ]

---

```
<script language="javascript"
  type="text/javascript">
  // Unser Request Objekt global verwendbar.
  var http_request = false;
  // doRequest(url) Erzeugt ein Requestobjekt abhängig
  vom Browser
  // und stellt eine anfrage an die übergebene URL
  function doRequest(url) { http_request = false;
  // Testen ob der Browser ein XMLHttpRequest Objekt
  // zur Verfügung stellt. (Mozilla und andere)
```

[

]

```
if (window.XMLHttpRequest)
    { http_request = new XMLHttpRequest();
      if (http_request.overrideMimeType)
          { //Das Ergebnis soll als XML-
            Dokument betrachtet werden.

            http_request.overrideMimeType('text/xml');
          }
    }
```

```
[
// Wenn der Browser ein IE ist,
// dann müssen andere Objekte benutzt werden. }
else if (window.ActiveXObject) {
    try
        { http_request = new
          ActiveXObject("Msxml2.XMLHTTP");}
catch (e)
    { try
        { http_request = new ActiveXObject
          ("Microsoft.XMLHTTP"); }
      catch (e) {}
    }
}
```

[

---

// Falls kein Request Objekt zur Verfügung steht

// funktioniert das ganze nicht

```
if (!http_request)
    { alert('Es kann kein Request Objekt
        erzeugt werden. ');
    return false;
}
```

[

]

// Eine Callbackfunktion wird angegeben,  
// die auf events des Request Objekts  
reagieren soll.

```
http_request.onreadystatechange = myHandler;  
http_request.open('GET', url, true);  
http_request.send(null);  
}
```

```
[
function myHandler() {
// Testen ob das Request objekt mit dem laden fertig ist
    if (http_request.readyState == 4) {
// War die Anfrage erfolgreich / die URL erreichbar ?
        if (http_request.status == 200)
            {
                ergebnisverarbeiten();
            }
        else
            { alert('Die URL konnte nicht geöffnet
                werden.');
```



[ ]

---

```
function ergebnisverarbeiten()  
{ ergebnisdiv=  
  document.getElementById('ergebnisfeld')  
  ergebnisdiv.innerHTML=http_request.responseText; }  
</script>  
<div onclick="doRequest('ajax.html')">  
<font color="blue">  
Klick mich. zum Absetzen des AJAX Requests </font>  
</div>  
<div id="ergebnisfeld"> </div>
```

# Klick mich. zum Absetzen des AJAX Requests

- **AJAX HTML**

Ich bin das Ergebnis.

# Vorteile von AJAX

## **Benutzerfreundlichkeit:**

- Die Wartezeit auf Reaktionen wird verringert (Latenzzeit).
- Das Interface reagiert viel schneller auf Benutzeraktionen als bei herkömmlichen Web-Applikationen.
- Der Rest der Seite kann weiterhin benutzt werden während eine Anfrage an den Server gestellt wird.

## **Benutzung bestehender Technologien:**

- AJAX verwendet keine neue Technologien, sondern benutzt schon bekannten Standard-Technologien.
- Dadurch müssen keine neuen Browser Plugins installiert werden.



## **Geringe Auslastung der Bandbreite:**

- **Der Client-Server Verkehr wird stark reduziert und dadurch die Bandbreite weniger ausgelastet.**

## **Unterstützung mehrerer Programmiersprachen:**

- **AJAX-Applikationen können mit den unterschiedlichsten Sprachen auf der Serverseite erstellt werden.**

# Nachteile von Ajax

## **Erreichbarkeit:**

- Um eine AJAX-Applikation ausführen zu können, muss folgende Punkte erfüllt sein:
  - JavaScript muss im Browser aktiviert sein
  - die Unterstützung des XMLHttpRequest Objekts muss vorhanden sein
  - beim Internet Explorer muss die Ausführung von ActiveX Objekten aktiviert sein
  - Suchmaschinen durchlaufen eventuell nicht den dynamischen Content

## **Browserabhängigkeit:**

- Die Erzeugung und der Zugriff auf das XMLHttpRequest Objekt sind in unterschiedlichen Browsern verschieden. Entwickler müssen also in ihren Scripts eine Unterscheidung der verschiedenen Browser vornehmen.

A large black left bracket and a large yellow right bracket are positioned at the top of the slide, with a horizontal olive-green line passing through them.

### **Wartbarkeit:**

- Durch die nötige Unterscheidung von Browsern, müssen die Scripts aktualisiert werden, sobald es einen neuen Browser auf dem Markt gibt.

### **Zurück-Button und History:**

- Da neue Seiteninhalte nicht mehr in Zusammenhang mit einer neuen Seite stehen, führen der Zurück-Button und die History nicht zur gewünschten Ergebnissen.

### **Link kopieren:**

- Es ist unmöglich, den Link zu einem gewissen Zustand der Seite zu kopieren und weiterzugeben, da sich an der URL während der Benutzung der Seite nichts ändert.


### **Probleme bei der Entwicklung:**

- Der Aufwand zur Erstellung einer AJAXApplikation ist erheblich höher als bei normalen Web-Applikationen, da der Entwickler berücksichtigen muss, dass nicht jeder User JavaScript aktiviert hat.


# [ Fazit ]


---

- AJAX stellt gute Lösungen dar  
z.B. Website hat die Möglichkeit im persönlichen Arbeitsplatz zu erscheinen.
- Auf diesen Arbeitsplatz, im Gegensatz zu traditionellen Desktop-Programmen kann man jederzeit über Internet zugreifen.  
Dabei lassen sich E-Mails verwalten,  
Notizen schreiben, Bilder online Verarbeiten,  
eine Favoritenliste anlegen oder auch neueste Artikel oder Nachrichten lesen.

- 
- Neuartige Benutzungsschnittstellen und Interaktionsmuster sind im Entstehen und regen die Phantasie von Nutzern und Entwicklern gleichermaßen zur Erschaffung neuartiger Anwendungen an.



- 
- Die Möglichkeiten, Nachteile von AJAX durch die Verwendung von Frameworks zu beseitigen, nehmen fast täglich zu.
  - Die ständige Entwicklung von AJAX-Frameworks, die das Erstellen von Ajax-Anwendungen erleichtert, lässt vermuten, dass die Verwendung von AJAX für immer mehr Web-Entwickler zur täglichen Aufgabe wird und die Anzahl der AJAXApplikationen weiter zunimmt.

- 
- Mittlerweile kann man sogar AJAX-Applikationen mit einem funktionierenden Zurück-Button ausstatten, und damit einen der bis jetzt bekanntesten und schwerwiegendsten Nachteile von AJAX beheben.
  - Die Stärken von AJAX liegen vor allem im Zusammenspiel unterschiedlicher bekannten Technologien, deren Kombination neue Anwendungsmöglichkeiten erschließt.



Wir danken

Ihnen für

ihre Aufmerksamkeit

# [ Literaturliste ]

---

- JavaScript und Ajax  
(7te Auflage von Christian Wenz)
- [www.bitroom.de](http://www.bitroom.de) (Schemata)
- <http://www.ajax-community.de>