



Grundlagen Internet-Technologien

INF3171

AJAX und Shibboleth

Version 1.0

17.07.2025





Jetzt aktualisieren! Chrome-Sicherheitslücke wird angegriffen

Google hat in der Nacht zum Mittwoch den Chrome-Webbrowser aktualisiert. Das Update schließt auch eine bereits attackierte Lücke.



(Bild: heise online / dmk)

08:31 Uhr Lesezeit: 3 Min. | Security

Von Dirk Knop

Erneut muss Google Sicherheitslücken im Webbrowser Chrome abdichten, von denen eine in freier Wildbahn bereits von Angreifern missbraucht wird. Wer Chrome nutzt, sollte sicherstellen, dass der Browser in aktueller Version läuft.



Phishing aktuell

Neue Datei über bwSync&Share

Sie haben eine Datei erhalten

Hallo [REDACTED],

eine Datei wurde Ihnen über den bwSync&Share-Dienst der Universität Hohenheim zur Verfügung gestellt:

Dateiname: SEPA_Echtzeitbestaetigung_20250704_0935.pdf
Größe: 3,2 KB

Datei im Browser ansehen

Sie können die Datei direkt im Browser öffnen, ohne sie herunterladen zu müssen.

© 2025 Universität Hohenheim – bwSync&Share
Dies ist eine automatisch generierte Nachricht. Bitte nicht antworten.

- Mailversand (intern) am 04.07.25 ca. 13:39h an 175 Personen.
- Phishing URL:
→ <https://uni-hohenheim.bwsyncandshare.de/idp/profile/SAML2/Redirect/SSO/?execution=e1s2>
- <https://uni-hohenheim.bwsyncandshare.de/idp/profile/SAML2/Redirect/SSO/?execution=e1s2>
- Korrekte URL:
→ https://bwsyncandshare.kit.edu/apps/user_saml/saml/selectUserBackEnd?redirectUrl=





- **Ajax:**

- zunächst Abkürzung für **Asynchronous JavaScript And XML**
- Jesse J. Garrett (AdaptivePath)
 - <http://www.adaptivepath.com/publications/essays/archives/000385.php>
(nicht mehr aktiv)
- nicht an sich neu („Technologiemix“), aber *marketingmäßig bedeutend*
- heute eigenständiges Synonym, kein Akronym



A screenshot of a web browser displaying the 'adaptive path' website. The page title is 'Ajax: A New Approach to Web Applications' by Jesse James Garrett, published on February 18, 2005. The article discusses the rise of web applications and the gap between desktop and web experiences. The sidebar features a photo of Jesse James Garrett and a list of his other ideas. The top of the page has a navigation bar with links to 'Work', 'Ideas', 'Events', 'About', and 'Contact', along with a search bar. The main content area includes a photograph of a group of people and some handwritten notes.

Ajax: A New Approach to Web Applications

BY JESSE JAMES GARRETT 0 Comments

If anything about current interaction design can be called "glamorous," it's creating Web applications. After all, when was the last time you heard someone rave about the interaction design of a product that wasn't on the Web? (Okay, besides the iPod.) All the cool, innovative new projects are online.

Despite this, Web interaction designers can't help but feel a little envious of our colleagues who create desktop software. Desktop applications have a richness and responsiveness that has seemed out of reach on the Web. The same simplicity that enabled the Web's rapid proliferation also creates a gap between the experiences we can provide and the experiences users can get from a desktop application.

That gap is closing. Take a look at [Google Suggest](#). Watch the way the suggested terms update as you type, almost instantly. Now look at [Google Maps](#). Zoom in. Use your cursor to grab the map and scroll around a bit. Again, everything happens almost instantly, with no waiting for pages to reload.

ABOUT JESSE
President



MORE IDEAS FROM JESSE

[Ten Years Later: Way Beyond Digital](#)
March 2, 2011

[Congratulations About.me!](#)
December 20, 2010

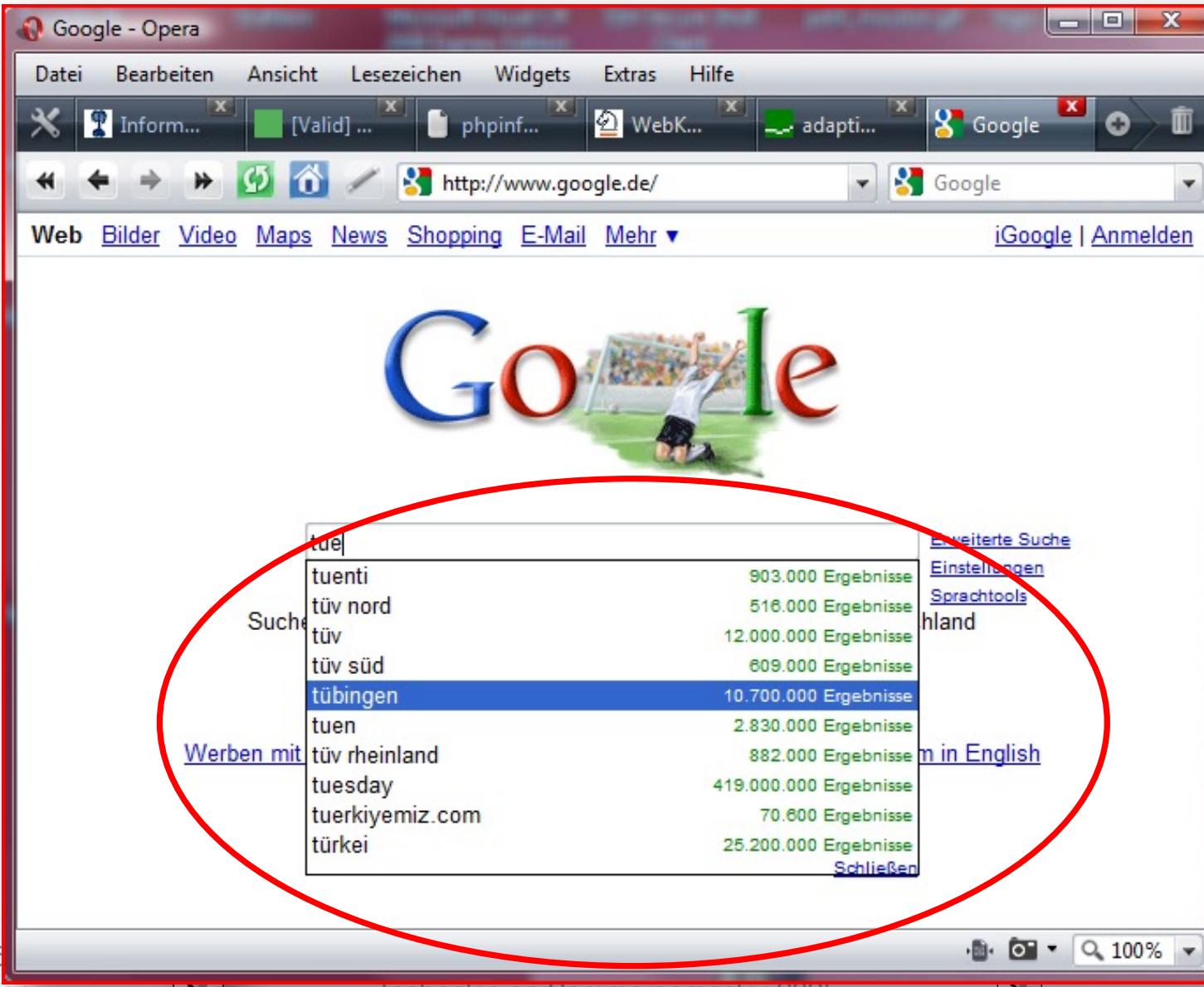
[Weighing the Pros and Cons—Professionals and Consumers, That Is](#)
October 27, 2010

[iPad Frees Designers From the Tyranny of QWERTY](#)
June 11, 2010

[Facebook and the User Experience of Privacy](#)
June 3, 2010



Beispiel für Ajax



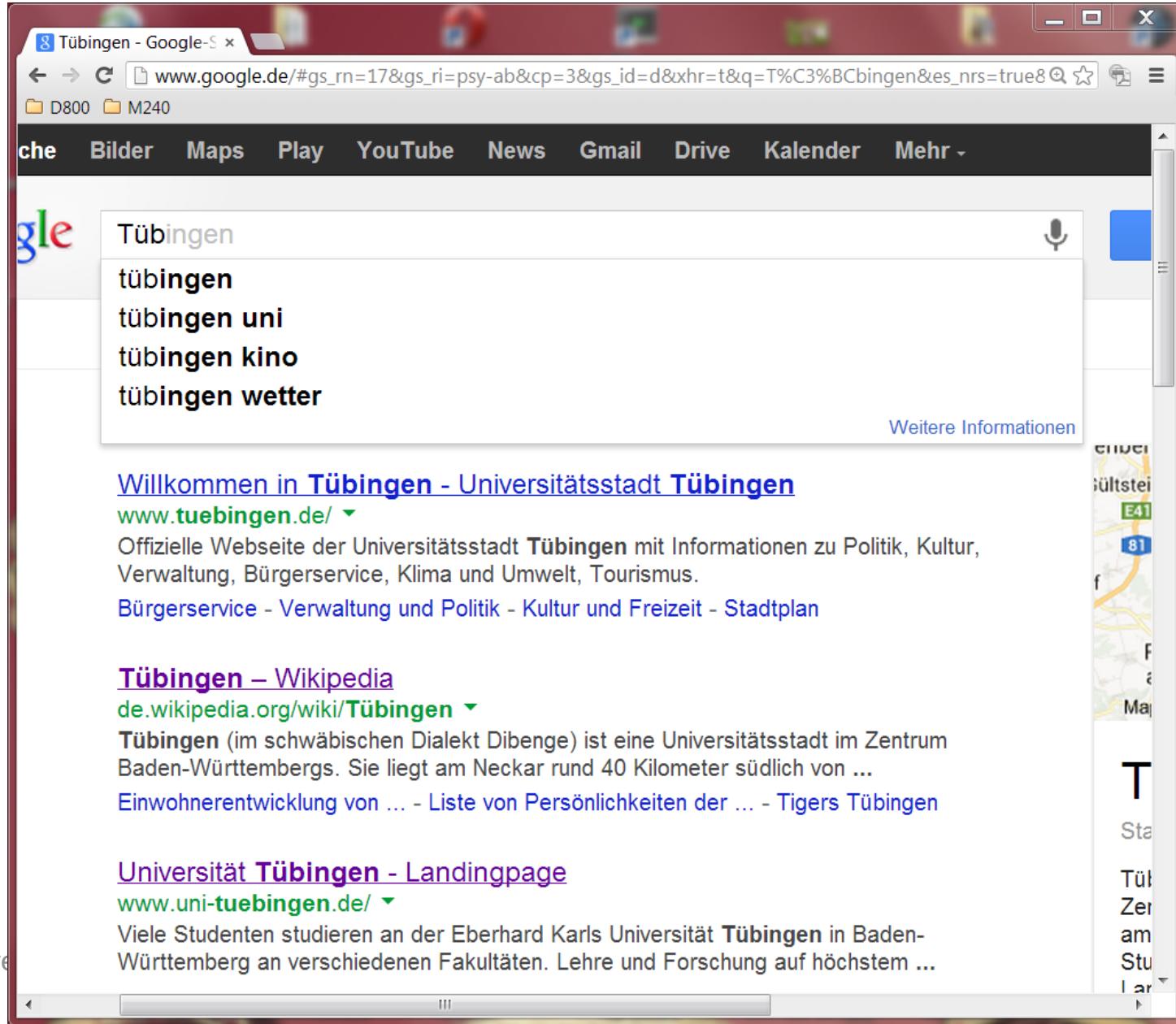
The screenshot shows a Google search page in an Opera browser. A red circle highlights a search suggestions dropdown for the query 'tue'. The dropdown lists various search terms and their result counts:

Suche	Ergebnisse
tuenti	903.000 Ergebnisse
tüv nord	516.000 Ergebnisse
tüv	12.000.000 Ergebnisse
tüv süd	609.000 Ergebnisse
tübingen	10.700.000 Ergebnisse
tuen	2.830.000 Ergebnisse
tüv rheinland	882.000 Ergebnisse
tuesday	419.000.000 Ergebnisse
tuerkiyemiz.com	70.600 Ergebnisse
türkei	25.200.000 Ergebnisse

Below the suggestions, there are links for 'Werben mit' and 'in English'. The browser interface includes a menu bar, a toolbar with various icons, and a status bar at the bottom.



Beispiel für Ajax



The screenshot shows a web browser window with the address bar containing `www.google.de/#gs_rn=17&gs_ri=psy-ab&cp=3&gs_id=d&xhr=t&q=T%C3%BCbingen&es_nrs=true8`. The search bar has the text "Tübingen". A dropdown menu lists suggestions: "tübingen", "tübingen uni", "tübingen kino", and "tübingen wetter". Below the suggestions, a link to the official website of Tübingen is shown: [Willkommen in Tübingen - Universitätsstadt Tübingen](http://www.tuebingen.de/). The page content includes a map of the area and links to various city services. Other search results listed are "Tübingen – Wikipedia" and "Universität Tübingen - Landingpage".



Automatische Vervollständigung bei der Suche nutzen

Beim Eingeben von Suchbegriffen bei Google finden Sie das Gesuchte schneller, wenn Sie auf die automatischen Vervollständigungen achten. Diese Vervollständigungen werden Ihnen als mögliche Suchbegriffe angezeigt, die Sie verwenden können. Sie hängen mit den Suchbegriffen zusammen, die Sie bereits eingegeben haben, und wurden auch schon von anderen Nutzern verwendet.

So funktioniert die automatische Vervollständigung

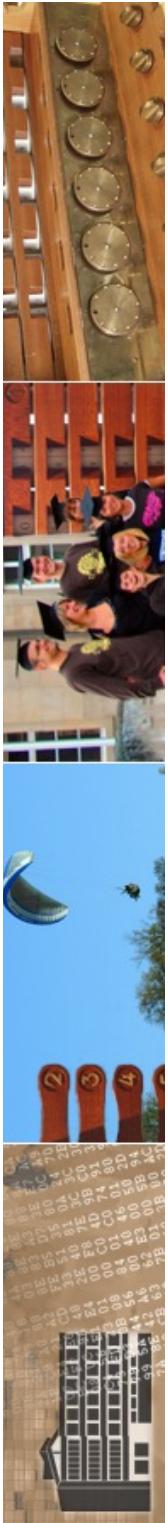
Vervollständigungen werden anhand von Faktoren wie der Häufigkeit und Aktualität von Suchbegriffen generiert. Wenn Sie eine Vervollständigung auswählen, wird nach diesem Begriff gesucht.

Woher stammen die Begriffe der Vervollständigungen?

- Aus Ihrer aktuellen Sucheingabe
- Aus relevanten Google-Suchen, die Sie in der Vergangenheit ausgeführt haben (sofern Sie in Ihrem Google-Konto angemeldet waren und [Web- & App-Aktivitäten](#) aktiviert hatten).
- Aus Suchanfragen anderer Nutzer, u. a. zu Topmeldungen . Topmeldungen sind beliebte Themen in Ihrer Region, die sich im Laufe des Tages ändern. Sie hängen nicht mit Ihrem Suchverlauf zusammen. Topmeldungen [finden Sie bei Google Trends](#) .

Hinweis: Vervollständigungen von Suchanfragen stellen keine Antwort auf Ihre Suche dar. Außerdem sind sie keine Aussagen von anderen Nutzern oder von Google zu Ihren Suchbegriffen.





technische Grundlage

- zunächst: proprietäre Lösung von Microsoft für IE 5 mit ActiveX
- allgemein:
JavaScript-Objekt
XMLHttpRequest
 - wird von allen Browsern unterstützt
 - Firefox ab 1.0, Opera ab 8, Safari ab 1.2, ...
 - auch IE ab Version 7 ☺

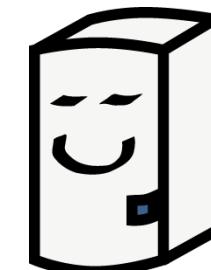


Ablauf einer Ajax-Anfrage

- Browser kann JavaScript-gesteuert kontinuierlich weitere Anfragen an Server stellen (asynchron!)
- Neuladen **eines Teils des Browsers**



Browser



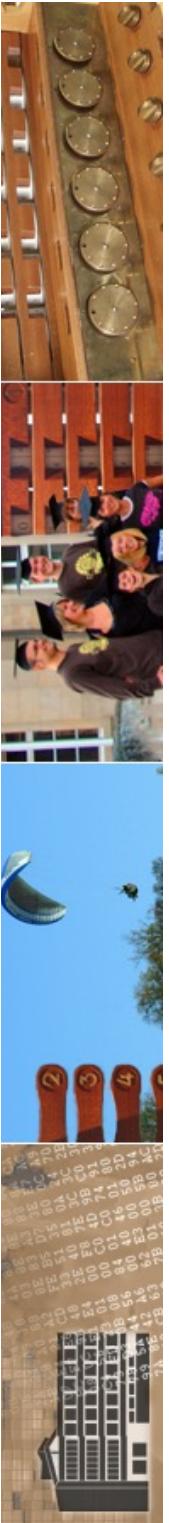
Datenhaltung

Ajax

← →

HTTP, XML, JSON, ...

Webserver



Detail der Abfrage

- drei Schritte
 - Erzeugung des XMLHttpRequest-Objektes:
`var http = new XMLHttpRequest();`
 - Methode `open` aus dieser Klasse
 - Parameter: HTTP-Methode (GET/POST), Server-URL, synchron (false) oder asynchron (true)
 - Anzeige des Ergebnisses über JavaScript-Event `onreadystatechange`

(Variable `readyState`: 0: nicht initialisiert, 1: Laden, 2: fertig, 3: warten, 4: beendet)



Beispiel 1

- Nachladen einer Datei mittels Ajax:
 - Website wird geladen
 - Ajax fügt asynchron eine Datei (HTML-Code) ein



File Edit Source Navigate Search Project Run Window Help

Java Perl JavaScript PHP Ruby Resource PyDev Web

helloworld.php ajaxtest.html

```
8
9<script type="text/javascript">!--
10    var http = null;
11    // Initialisieren http-Objekt und Abfangen alter IE
12    if (window.XMLHttpRequest)
13        http = new XMLHttpRequest();
14    else {if (window.ActiveXObject)
15        http = new ActiveXObject("Microsoft.XMLHTTP");
16        else alert( "Browser unterstuetzt kein Ajax");
17    }
18
19    if (http != null) {
20        // asynchrone http-Verbindung zum Server
21        http.open("GET", "ajaxFile.html", true);
22        // bei Aenderung von readyState wird "ausgeben" aufgerufen
23        http.onreadystatechange = ausgeben;
24        // Absenden der Anfrage
25        http.send(null);
26    } // if
27
28    function ausgeben() {
29        if (http.readyState == 4) {
30            document.getElementById("ausgabe").innerHTML =
31                http.responseText; // Rueckgabe des HTTP-Requests als Text
32        } // if
33    } // function
34 //--></script>
35</head>
36<body>
37    <hr><center><h2>Grundlagen Internet-Technologien</h2><h3>Ajax</h3><hr>
38    HTML-Ausgabe über Ajax (asynchron) vom Server:
39
40    <div id="ausgabe"></div>
41
42    </center><hr>
43</body>
44</html>
```

html/#text Writable Smart Insert 2:7



```
<!DOCTYPE html>
<html>
  <head>
    <title>Grundlagen Internet-Technologien: Ajax</title>
    <link rel="stylesheet" type="text/css"
          href="../css/webkompendium.css">

    <script type="text/javascript"><!--
        var http = null;
        // Initialisieren http-Objekt und Abfangen alter IE
        if (window.XMLHttpRequest)
            http = new XMLHttpRequest();
        else {if (window.ActiveXObject)
                http = new ActiveXObject("Microsoft.XMLHTTP");
                else alert( "Browser unterstuetzt kein Ajax");
            }

        if (http != null) {
            // asynchrone http-Verbindung zum Server
            http.open("GET", "ajaxFile.html", true);
            // bei Aenderung von readyState wird "ausgeben" aufgerufen
            http.onreadystatechange = ausgeben;
            // Absenden der Anfrage
            http.send(null);
        } // if

        function ausgeben() {
            if (http.readyState == 4) {
                document.getElementById("ausgabe").innerHTML =
                http.responseText; // Rueckgabe des HTTP-Requests als Text
            } // if
        } // function
    //--></script>
  </head>
  <body>
    <hr><center><h2>Grundlagen Internet-Technologien</h2><h3>Ajax</h3><hr>
    HTML-Ausgabe über Ajax (asynchron) vom Server:

    <div id="ausgabe"></div>

  </center><hr>
--More--(98%)
```





```
zrvwa01@infodienste =>
zrvwa01@infodienste =>
zrvwa01@infodienste => more ajaxFile.html
<p>Grundlagen Internet-Technologien:<br><b>Ajax</b>
Mehr Dynamik im Web mit Ajax</p>
<p>Tour de France 2025</p>
```



Grundlagen Internet-Technologien

Ajax

HTML-Ausgabe über Ajax (asynchron) vom Server:

Grundlagen Internet-Technologien:
Ajax Mehr Dynamik im Web mit Ajax

Tour de France 2025



Beispiel 2: Kommunikation mit Server

- Client: Ajax-Request, welcher Parameter enthält
- Server: PHP-Script zum Auslesen der Parameter, formatiertes Rücksenden an Client



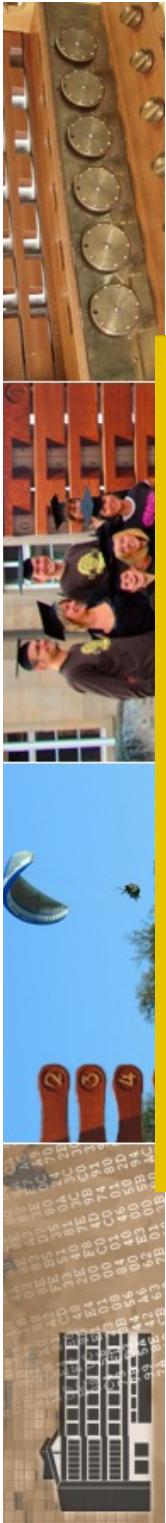
```
<!DOCTYPE html>
<html>
  <head>
    <title>Grundlagen Internet-Technologien: Parameteruebergabe mit Ajax</title>
    <link rel="stylesheet" type="text/css" href="../css/webkompendium.css">
    <link rel="shortcut icon" href="/css/favicon.ico">

    <script type="text/javascript"><!--
      var http = null;
      // Initialisieren http-Objekt und Abfangen alter IE
      if (window.XMLHttpRequest) {
          http = new XMLHttpRequest();
      } else if (window.ActiveXObject) {
          http = new ActiveXObject("Microsoft.XMLHTTP");
      }

      if (http != null) {
        // Parameteruebergabe mit GET
        http.open("POST", "parameter.php?n=42&text=Tuebingen", true);
        // bei Aenderung von readyState wird "ausgeben" aufgerufen
        http.onreadystatechange = ausgeben;
        // Ausgabe
        http.send(null);
      } // if

      function ausgeben() {
        if (http.readyState == 4) {
          document.getElementById("Ausgabe").innerHTML =
          http.responseText;
        } // if
      } // function
    //--></script>
  </head>

  <body>
    <hr><center><h2>Grundlagen Internet-Technologien<BR>
    Parameter&uuml;bergabe mit Ajax an Server (PHP)</h2><hr>
    </center>
    Nachrichten <div id="Ausgabe"></div>
  </body>
</html>
```

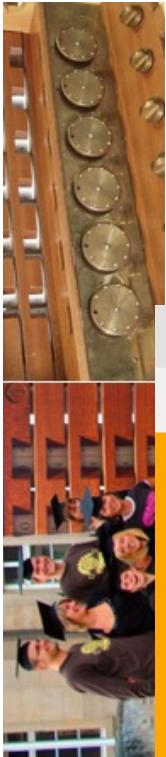


<!--

Grundlagen Internet-Technologien:
PHP-Skript zur Ausgabe aller mit GET oder POST
uebergebener Parameter

-->

```
<h3>GET</h3><code>  
<?php  
    foreach($_GET as $key => $value)  
        echo "Parameter $key hat Wert $value<BR>";  
    ?>  
</code>  
  
<h3>POST</h3><code>  
<?php  
    foreach($_POST as $key => $value)  
        echo "Parameter $key hat Wert $value<BR>";  
    ?>  
</code>
```



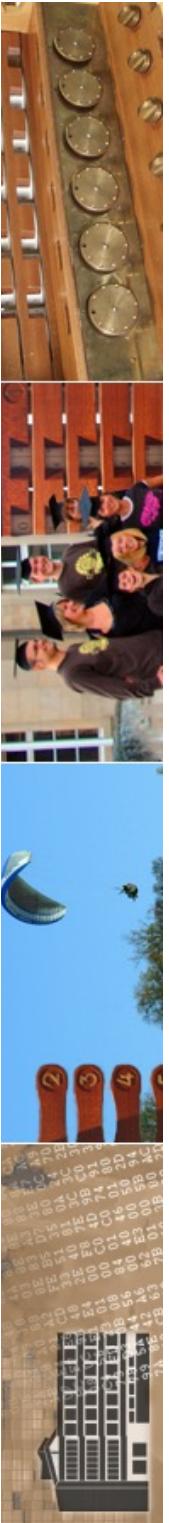
Grundlagen Internet-Technologien
Parameterübergabe mit Ajax an Server (PHP)

GET

Parameter n hat Wert 42
Parameter text hat Wert Tuebingen

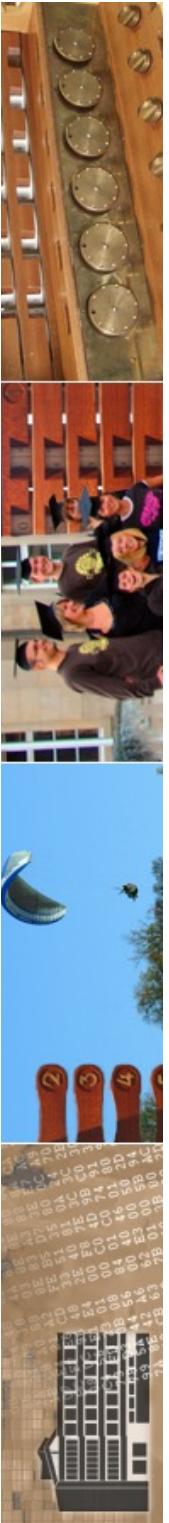
POST





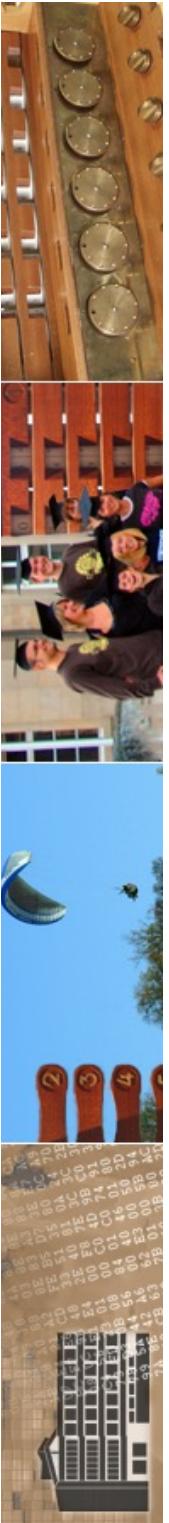
Übersicht XMLHttpRequest-Objekt

- Konstruktor
- Methoden
 - `abort`
 - `getAllResponseHeaders`
 - `getResponseHeader`
 - `open`
 - `send`
- Eigenschaften
 - `onreadystatechange`
 - `readyState`
 - `responseText`
 - `responseXML`
 - `status`
 - `statusText`



Probleme von Ajax

- Ajax hat auch Kritikpunkte
 - Usability: beispielsweise haben die Browsetasten vor/zurück nun eine unlogische Bedeutung
 - auch Ajax-Site ist nicht mehr zustandslos!
 - Links setzen: man kann nur auf den Startzustand einer Ajax-Anwendung einen Link setzen



Websockets

- Alternative zu AJAX: Websockets
 - Bestandteil von HTML5
 - ähnlich, aber anders
 - *kontinuierliche* bidirektionale Verbindung
 - typischerweise **json** als Austauschformat





Authentifizierung

- wie authentifiziert sich ein Nutzer?
- die einfache Antwort: Kennwort
 - Regeln? (“acht Zeichen mit Sonderzeichen und Ziffern”)
 - Ablauf?
- Biometrie?



Authentifizierung im Web

- besondere Herausforderung: Authentifizierung im Web
- „Shibboleth“ (SAML2): Kennwort verlässt die Heimateinrichtung nie
 - funktioniert nur mit http/https



Prinzip von Shibboleth



Benutzer



Anbieter
(Service Provider)



Login

Lokalisierungs-
dienst
(Discovery Service)

nein

Benutzer bekannt?

ok

ok

Benutzer berechtigt?

Benutzerrollen

Zugriff ja/nein

Heimat-
einrichtung
(Identity
Provider)



LOGIN @ UNI-TUEBINGEN

Shibboleth · Identity Provider

Dies ist die zentrale Anmeldeseite für das Single Sign-On der Universität Tübingen.

Ein Dienst hat angefordert, dass Sie sich über Ihr persönliches Benutzerkonto der Universität Tübingen anmelden.

Login-ID

[› Passwort vergessen?](#)

[› Hilfe benötigt?](#)

[› Nutzungsbedingungen](#)

Zentrales Passwort

Anmeldung nicht speichern

Löschen der früheren Einwilligung zur Weitergabe Ihrer Informationen an diesen Dienst.

Anmelden





An den Dienst zu übermittelnde Informationen

Benutzername	zrvwa01
Vorname(n)	Thomas
E-Mail-Adresse	thomas.walter@uni-tuebingen.de
Bereichsspezifische Zugehörigkeit	member@uni-tuebingen.de
Berechtigung	urn:mace:dir:entitlement:common-lib-terms http://moodle.uni-tuebingen.de/user
Nachname	Walter

Die oben aufgeführten Informationen werden an den Dienst **moodle02.zdv.uni-tuebingen.de** weitergegeben, falls Sie fortfahren. Sind Sie einverstanden, dass diese Informationen bei jedem Zugriff auf diesen Dienst an ihn weitergegeben werden?

Wählen Sie die Dauer, für die Ihre Entscheidung zur Informationsweitergabe gültig sein soll:

Bei nächster Anmeldung erneut fragen.

› Ich bin einverstanden, meine Informationen dieses Mal zu senden.

Erneut fragen, wenn sich die Informationen ändern, welche diesem Dienst weitergegeben werden.

› Ich bin einverstanden, dass dieselben Informationen in Zukunft automatisch an diesen Dienst weitergegeben werden.

Diese Einstellung kann jederzeit mit der Checkbox auf der Anmeldeseite widerrufen werden.



...und nun...

- haben wir wesentliche Erweiterungen für die Web-Programmierung kennen gelernt, insbesondere Ajax

- als nächstes:

Zusammenfassung

