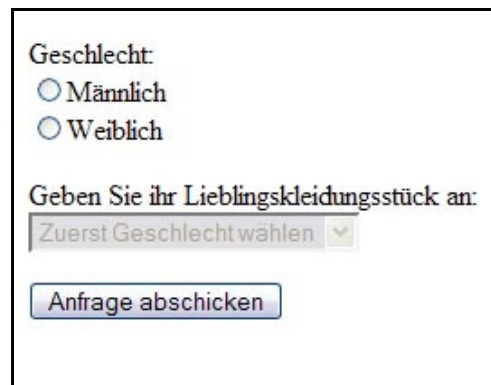


4. Kurzbeispiel zur Erklärung der Funktionsweise

Im Folgenden möchte ich anhand eines Kurzbeispiels die Funktionsweise von AJAX vorstellen und die einzelnen Komponenten erläutern.



Die Ansicht im Browser vor Eingabe

Eine lauffähige Version findet man auch im WWW unter <http://ajax.eifelartig.de/kurz.htm> .

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-tran
nsitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>Kurzbeispiel</title>
6 <script language="javascript" src="kurz.js"></script>
7 </head>
8 <body>
9 <form id="form1" name="form1" method="post" action="">
10 <p>Geschlecht:<br />
11 <input type="radio" name="ger" value="m" onclick="sndreq('m');"/>Männlich<br />
12 <input type="radio" name="ger" value="w" onclick="sndreq('w');"/>Weiblich</p>
13 <p id="form2">Geben Sie ihr Lieblingskleidungsstück an:<br />
14 <select name="kleid" disabled="disabled">
15 <option selected="selected" disabled="disabled">Zuerst Geschlecht wählen</option>
16 </select></p>
17 <input type="submit" />
18 </form>
19 </body>
20 </html>
```

Quelltext kurz.htm

Wie man sehen kann, handelt es sich hier um eine gewöhnliche HTML- Seite. Sie enthält ein Formular, auf dem man zunächst mittels eines Radio- Buttons das Geschlecht auswählen muss. Im weiteren Verlauf wird ein Drop- Down- Auswahlmenü angezeigt, welches aber deaktiviert ist.

Im head wird die externe JavaScript- Datei kurz.js eingebunden. Des Weiteren findet man an den Radiobuttons zur Auswahl des Geschlechts onclick- Eventhandler, die die Funktion sndreq mit einem Übergabeparameter aufrufen.

```

1 // JavaScript Document
2 //Ajax Funktionen
3 function erzXMLHttpRequestObject(){
4     var resObjekt = null;
5     try {
6         resObjekt = new ActiveXObject("Microsoft.XMLHTTP");
7     }
8     catch(Error){
9         try {
10            resObjekt = new ActiveXObject("MSXML2.XMLHTTP");
11        }
12        catch(Error){
13            try {
14                resObjekt = new XMLHttpRequest();
15            }
16            catch(Error){
17                alert("Erzeugung des XMLHttpRequest-Objekts ist nicht möglich");
18            }
19        }
20    }
21    return resObjekt;
22 }
23 function sndreq(gen){
24     resObjekt.open('post','kurz.php',true);//Methode, URL, asynchron?
25     resObjekt.onreadystatechange=handleResponse;
26     resObjekt.setRequestHeader("Content-Type","application/x-www-form-urlencoded");//zusätzliche Header Tr
27     resObjekt.send("gen="+gen);//Post wird mit Werten belegt
28 }
29 function handleResponse(){
30     if(resObjekt.readyState==4){
31         document.getElementById("form2").innerHTML=resObjekt.responseText;
32     }
33     else{
34         document.getElementById("form2").innerHTML="Bitte warten";
35     }
36 }
37 resObjekt=erzXMLHttpRequestObject();

```

Quelltest kurz.js

Die JavaScript- Quelltextdatei kurz.js enthält drei Funktionen.

Die erste Funktion, erzXMLHttpRequestObject, ist die Browserweiche und erzeugt das jeweils für den benutzten Browser passende XMLHttpRequest-Objekt. Die beiden ersten beziehen sich auf Versionen des Microsoft Internet Explorers, die letzte auf alle anderen Browser. Als Rückgabewert wird die Referenz auf das gerade erzeugte Objekt übergeben, Fehlermeldungen bei einer fehlgeschlagenen Erzeugung werden abgefangen.

Die zweite Funktion, sndreq, erwartet einen Übergabewert und sendet ihn an ein PHP- Skript in dem gleichen Verzeichnis des Webservers. Die Funktion bedient sich Eigenschaften und Methoden des XMLHttpRequest- Objekts. Die open- Methode öffnet eine Verbindung zum Server. Der erste Parameter legt die Art der Datenübertragung per Http fest, in diesem Falle also POST. Der zweite Parameter ist die URL des Skripts, welches auf Serverseite beim Senden aufgerufen werden soll. Bei GET- Übertragung nutzt man an dieser Stelle eine Pseudo- URL mit angefügtem Querystring. Der dritte Parameter legt fest, ob das Skript weiterlaufen soll während auf die Antwort vom Server gewartet wird. Die Eigenschaft onreadystatechange ist ein Eventhandler, der jedes Mal aufgerufen wird, wenn sich der Ready State des XMLHttpRequest- Objekts ändert. An dieser Stelle wird die Funktion handleResponse als Callbackfunktion

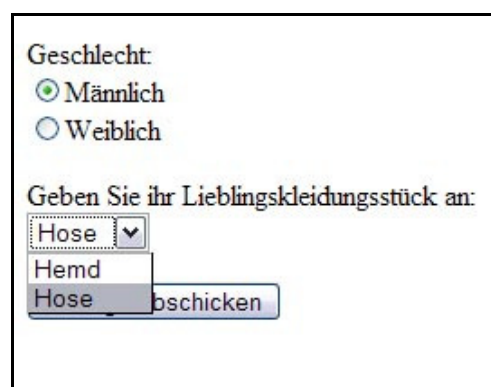
ohne Klammern aufgerufen, d.h. `handleResponse` wird direkt von dem Eventhandler referenziert und hat keinen Einfluss auf den restlichen Programmablauf. Am Ende kümmert sich die Methode `send` darum, dass die Daten übertragen werden. Wird POST benutzt, muss man hier den Querystring als Parameter übergeben.

Die letzte Funktion, `handleResponse`, legt fest was passiert, wenn das `XMLHttpRequest`- Objekt seinen Status ändert. Ist die Seite fertig geladen, wird der Antworttext, gespeichert in der Eigenschaft `responseText`, in den DOM Baum an der Stelle mit der ID „form2“ eingehängt. Ist die Seite noch nicht fertig geladen, dann erscheint an dieser Stelle „Bitte warten“.

```
1 <?php
2 $gen=$_POST["gen"];
3 echo("Geben Sie ihr Lieblingskleidungsstück an:<br /><select name=\"kleid\">");
4 if($gen=="m"){
5     echo <<<TEXT
6         <option selected="selected">Hemd</option>
7         <option selected="selected">Hose</option>
8     TEXT;
9 }
10 else{
11     echo <<<TEXT
12         <option selected="selected">Bluse</option>
13         <option selected="selected">Rock</option>
14     TEXT;
15 }
16 echo("</select>");
17 ?>
```

Quelltext kurz.php

Auf der Serverseite findet man obiges Skript. Es übernimmt einfach die Variable „gen“ aus dem POST- Array und erzeugt dann, je nach Wert der Variable das passende Drop- Down- Menü.



The screenshot shows a web form with two sections. The first section is titled "Geschlecht:" and contains two radio buttons: "Männlich" (which is selected) and "Weiblich". The second section is titled "Geben Sie ihr Lieblingskleidungsstück an:" and contains a dropdown menu with "Hose" selected. Below the dropdown menu is a button labeled "abschicken".

Die Ansicht im Browser nach Angabe des Geschlechts