

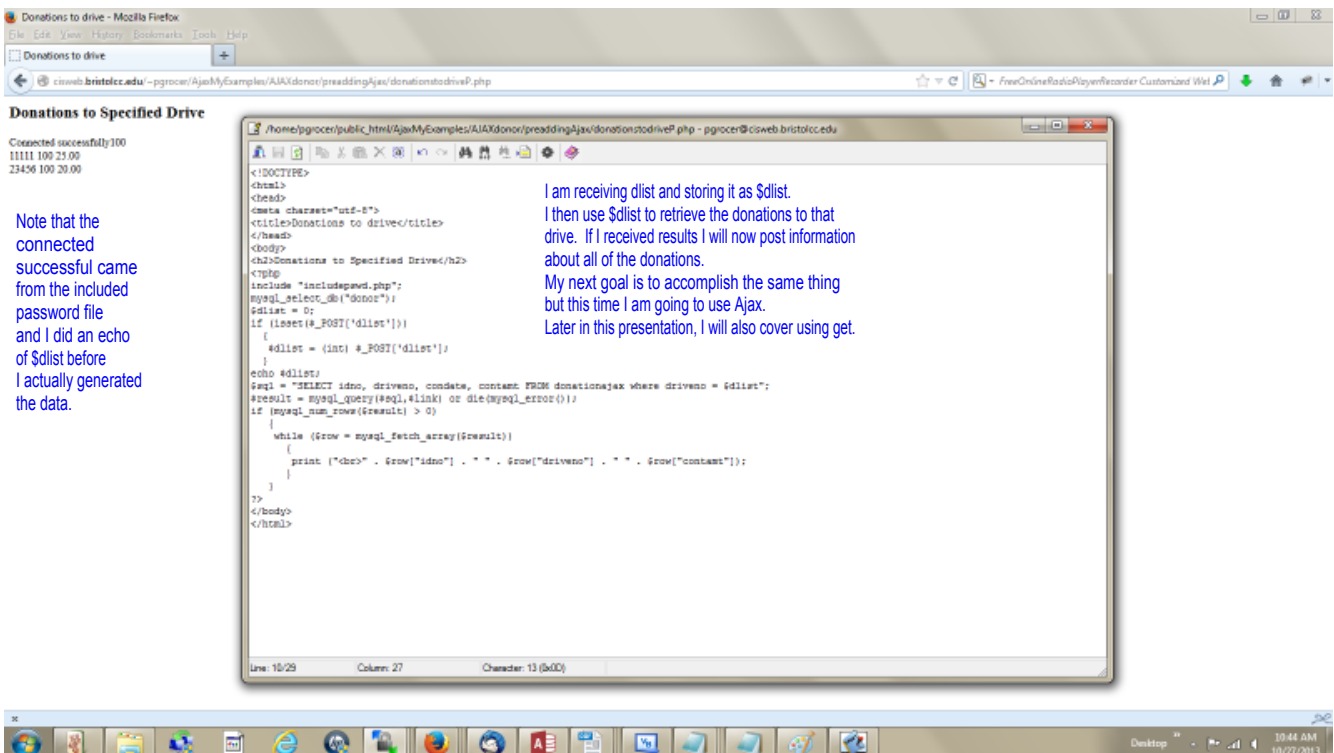
The screenshot shows a Mozilla Firefox browser window with the address bar displaying `firefox@bristolcc.edu/~pgrocer/AjaxMyExamples/AJAXdonor/preaddingAjax/donationformP.html`. The page content includes the text "This menu will allow you to select a drive and see the donations made to that drive." and a "Submit" button. A code editor window is overlaid on the page, showing the following HTML code:

```
<!DOCTYPE>
<html>
<head>
<meta charset="utf-8">
<title>Donations</title>
</head>
<body>
<h2>This menu will allow you to select a drive and see the donations made to that drive.</h2>
<form action="donationstodriveP.php" method="post" id="listdonation">
<p>
<select id="dlist" name="dlist">
<option value="100">Kids Shelter</option>
<option value="200">Animal Home</option>
<option value="300">Health Aid</option>
</select>
<input type="submit" value="Submit">
</p>
</form>
</body>
</html>
```

Below the code editor, a blue text box contains the following explanation:

First I wrote this program without Ajax. It is simply a form that is calling a php program using the post method. The data being passed is one of the values in the select and it is being passed as dlist.

The Windows taskbar at the bottom shows the system tray with the date and time: Desktop, 10:43 AM, 10/27/2013.



Note that the connected successful came from the included password file and I did an echo of \$dlist before I actually generated the data.

I am receiving dlist and storing it as \$dlist. I then use \$dlist to retrieve the donations to that drive. If I received results I will now post information about all of the donations. My next goal is to accomplish the same thing but this time I am going to use Ajax. Later in this presentation, I will also cover using get.

I should be showing the donationform.html in driveAjax - this one looks the same. The code is from the correct form.

Donations - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Donations

csweb.bristolcc.edu/~pgrocer/AjaxMyExample/AJAIdonor/preaddingAjax/donationform.html

This menu will allow you to select a drive and see the donations made to that drive.

Animal Home

```

<!DOCTYPE>
<html>
<head>
<meta charset="utf-8">
<title>Donations</title>
<script>
function getRecords() {
var requestStr = "dist=" + document.getElementById("dist").value;
var xmlhttp;
if (window.XMLHttpRequest)
    xmlhttp=new XMLHttpRequest();
else
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
xmlhttp.onreadystatechange=function()
{
if (xmlhttp.readyState==4 && xmlhttp.status==200)
    document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
}
xmlhttp.open("POST","donationtodrive.php",true);
xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
xmlhttp.send(requestStr);
}
</script>
</head>
<body>
<h2>This menu will allow you to select a drive and see the donations made to that drive.</h2>
<form id="Listdonation">
<p>
<select id="dist" name="dist">
<option value="100">Wild Shelter</option>
<option value="200">Animal Home</option>
<option value="300">Health Aid</option>
</select>
<button type="button" onclick="getRecords()">Retrieve</button>
</p>
</form>
<div id="myDiv"></div>
</body>
</html>

```

When you want to pass information to the server if you are using the POST method then you need to format the information you want to pass in a manner that the server can interpret.

The method is POST and behind the scenes in method POST a header is sent with the information from the form. With AJAX I am sending the information in the header. I execute the method .setRequestHeader and then when I execute the method .send it sends the header and the string I set up as requestStr.

The button does not have a standard submit. It is executed on an onclick event.

I have put the function getRecords() in the head and I call it on the onclick event. Then in the function I use the DOM getElementById("dist").value to make the dist available to the function.

This is coming as output from the PHP. It will then be displayed in the div with id="myDiv".

I set up a form using the select. The selection is called dist (using both id and name)

This is set up to receive the information I return.

This shows the correct form.

This menu will allow you to select a drive and see the donations made to that drive.

Animal Home

12321	200	40.00
23456	200	100.00

Once you have established the xmlhttp object you are then going to execute a series of methods. The first method is to establish what function you want to execute when the server responds to your request. You do that by setting the onreadystatechange of the xmlhttp object. All you have done is set that, you will not execute the function until the server responds to your request.

The next thing you do is open a channel to the server telling it the method you are using, the program you want it to execute and that the communication is going to be asynchronous. If you are using the post method you have to set the request header. Then you actually send the information to the server. On the server side it will receive that information and execute the program you told it to execute in the open. That program will produce output that the server will then send back to you and that output will be placed in the responseText of the xmlhttp object and the function you defined in the onreadystatechange will be executed.

```

<!DOCTYPE>
<html>
<head>
<meta charset="utf-8">
<title>Donations</title>
<script>
function getRecords() {
var requestStr = "dist=" + document.getElementById("dist").value;
var xmlhttp;
if (window.XMLHttpRequest) {
// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp=new XMLHttpRequest();
}
else {
// code for IE6, IE5
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.onreadystatechange=function()
{
if (xmlhttp.readyState==4 && xmlhttp.status==200)
{
document.getElementById("myDiv").innerHTML+xmlhttp.responseText;
}
}
xmlhttp.open("POST","donationsCode.php",true);
xmlhttp.setRequestHeader("Content-type","application/x-www-form-urlencoded");
xmlhttp.send(requestStr);
}
</script>
</head>
<body>
<h2>This menu will allow you to select a drive and see the donations made to that drive.</h2>
<form id="listdonation">
<input type="button" value="Retrieve"/>
</form>
<div id="myDiv"></div>
</body>
</html>

```

You need a xmlhttp request object that will handle that asynchronous communication with the server and depending on the browser and the age of the browser you want to use the standard request object or the older ActiveX object with older explorer versions. Once it is instantiated you use it to test readyState, status, responseText and any other Ajax functions.

The data got processed in the PHP and returned and displayed in the div with id="myDiv".

```
<?php
include "includepswd.php";
mysql_select_db("donor");
$dlist = 0;
if (isset($_POST['dlist']))
{
    $dlist = (int) $_POST['dlist'];
}
//echo $dlist;
$sql = "SELECT idno, driveno, condate, contamt FROM donationajax where driveno = $dlist";
$result = mysql_query($sql,$link) or die(mysql_error());
if (mysql_num_rows($result) > 0)
{
    while ($row = mysql_fetch_array($result))
    {
        print ("<br>" . $row["idno"] . " " . $row["driveno"] . " " . $row["contamt"]);
    }
}
?>
```

This is donationstodrive.php

It receives dlist and assigns it to \$dlist
and then retrieves the information and
sets up the output.

Notice there is no html on this page because you do not want html sent
back to you, you just want to send back the information you need.
Also note that in my password includepswd.php I eliminated the things
that outputted the reassuring comments about being connected etc.

```
~/home/pgrocer/public_html/AjaxMyExamples/AJAXdonor/driveAjax/donationstodrive.php - pgrocer@dsweb.bristolcc.edu
k:php
include "includeprwd.php";
mysql_select_db("donor");
$dlist = 0;
if (isset($_POST['dlist']))
{
    $dlist = (int) $_POST['dlist'];
}
//echo $dlist;
$sql = "SELECT idno, driveno, condate, constant FROM donationajax where driveno = $dlist";
$result = mysql_query($sql,$link) or die(mysql_error());
if (mysql_num_rows($result) > 0)
{
    while ($row = mysql_fetch_array($result))
    {
        print "<br>". $row["idno"] . " " . $row["driveno"] . " " . $row["constant"];
    }
}
?>
```

Same as previous slide.

Line 1/19 Column: 1 Character: 60 (a:3C)

Desktop 11:30 AM 10/27/2013

The screenshot shows a Mozilla Firefox browser window with the address bar displaying `csweb.bristolcc.edu/~pgrocer/AjaxMyExamples/AJAXdonor/preaddingAjax/donationform.html`. The page content includes the text "This menu will allow you to select a drive and see the donations made to that drive." and a "Submit" button. An overlaying code editor window displays the following HTML code:

```
<!DOCTYPE>
<html>
<head>
<meta charset="utf-8">
<title>Donations</title>
</head>
<body>
<h2>This menu will allow you to select a drive and see the donations made to that drive.</h2>
<form action="donationstodrive.php" method="get" id="listdonation">
  <p>
    <select id="dlist" name="dlist">
      <option value="100">Kids Shelter</option>
      <option value="200">Animal Home</option>
      <option value="300">Health Aid</option>
    </select>
    <input type="submit" value="Submit">
  </p>
</form>
</body>
</html>
```

Below the code editor, a blue text annotation reads: "This version is using the get as opposed to the post and it is NOT using AJAX."

Donations to drive - Mozilla Firefox

Donations to drive

Donations to Specified Drive

Connected successfully!100
11111 100 25.00
23456 100 20.00

```

/home/pgrocer/public_html/AjaxMyExamples/AJAXdonor/preaddingAjax/donationtodrive.php - pgrocer@ciweb.bristolcc.edu
<!DOCTYPE>
<html>
<html>
<head>
<meta charset="utf-8">
<title>Donations to drive</title>
</head>
<body>
<h2>Donations to Specified Drive</h2>
<?php
include "includepwd.php";
mysql_select_db("donor");
$dlst = 0;
if (isset($_GET['dlst']))
{
$dlst = (int) $_GET['dlst'];
}
echo $dlst;
$sql = "SELECT idno, driveno, condate, content FROM donationajax where driveno = $dlst";
$result = mysql_query($sql,$link) or die(mysql_error());
if (mysql_num_rows($result) > 0)
{
while ($row = mysql_fetch_array($result))
{
print ("<br>" . $row["idno"] . " " . $row["driveno"] . " " . $row["content"]);
}
}
?>
</body>
</html>
Line: 1/29 Column: 1 Character: 60 (D3C)

```

Note that the connected successful came from the included password file and I did an echo of \$dlst before I actually generated the data. Again this is not an AJAX version.

Desktop 11:35 AM 10/27/2013

Back to using AJAX.
 The get includes the data you are sending the server appended to the URL of the program you want to execute on the server including the ?. So here you are appending the data to the URL.
 The way the post method works it sends the data over in a header not as part of the url. So with post you need to send a header, with get you don't.
 You use xmlhttp.send() with the get. With the post you need to tell it what data to send as part of the header, with the get it know what data is being sent because it is part of the URL.
 Notice that requestStr is in the open, not in the send (as it was in the post).
 The requestStr in the post just has the data, the requestStr in the get contains the URL.

```

<!DOCTYPE>
<html>
<head>
<meta charset="utf-8">
<title>Donations</title>
<script>
function getRecords() {
  var requestStr = "donationstodrive.php?dlist=" + document.getElementById("dlist").value;
  var xmlhttp;
  if (WINDOW.XMLHttpRequest)
  {
  // code for IE7+, Firefox, Chrome, Opera, Safari
  xmlhttp=new XMLHttpRequest();
  }
  else
  {
  // code for IE6, IE5
  xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
  xmlhttp.onreadystatechange=function()
  {
  if (xmlhttp.readyState==4 && xmlhttp.status==200)
  {
  document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
  }
  }
  xmlhttp.open("GET",requestStr,true);
  xmlhttp.send();
  }
</script>
</head>
<body>
<h2>This menu will allow you to select a drive and see the donations made to that drive.</h2>
<form action="donationstodrive.php" method="get" id="listdonation">
<input type="text" value="100" />
<select id="dlist" name="dlist">
<option value="100">Kids Shelter</option>
<option value="200">Animal Home</option>
<option value="300">Health Risk</option>
</select>
<input type="button" value="Retrieve" />
</form>
<div id="myDiv"></div>
</body>
</html>
  
```

Note the major differences between using get and post. This version uses get.

```
.:/home/pgroce/public_html/AjaxMyExamples/AJAXdonor/driveAjax/donationstodrive.php - pgroce@cisweb.bristolcc.edu
<?php
include "includepwd.php";
mysql_select_db("donor");
$dlist = 0;
if (isset($_GET['dlist']))
{
    $dlist = (int) $_GET['dlist'];
}
//echo $dlist;
$sql = "SELECT idno, driveno, contdate, content FROM donationsajax where driveno = $dlist";
$result = mysql_query($sql,$link) or die(mysql_error());
if (mysql_num_rows($result) > 0)
{
    while ($row = mysql_fetch_array($result))
    {
        print "<br>". $row["idno"] . " " . $row["driveno"] . " " . $row["content"];
    }
}
?>
```

Using the GET.

This is donationAjaxPost.html. This codes uses external JavaScript and I created two scripts. The first one instantiates the XML HTTP Request Object and includes the code to check for the browser so you get a browser compatible XMLHttpRequestObject.

```

<!DOCTYPE>
<html>
<head>
<meta charset="utf-8">
<title>Donations</title>
<script src="getXMLHttpRequestObject.js" type="text/javascript"></script>
<script src="processDataPost.js" type="text/javascript"></script>
</head>
<body>
<h2>This menu will allow you to select a drive and see the donations made to that drive.</h2>
<form id="listdonation">
<p>
<select id="dlist" name="dlist">
<option value="100">Kids Shelter</option>
<option value="200">Animal Home</option>
<option value="300">Health Aid</option>
</select>
<button type="button" onclick="getRecords(dlist.value);">Retrieve</button>
</p>
</form>
<div id="myDiv"></div>
</body>
</html>

```

In the HTML you have a form with no action attribute and if you look further down you will notice that there is no submit button. What the second JavaScript will do is perform the functions of the form action and the submit button.
Now I will bring up processDataPost.js.

Note that we add an onclick event handler reference on the button tag. When we click on the button the JavaScript function getRecords (in processDataPost) is executed. It sends the value from the select on the form.

This is processDataPost.js.

The idea is that because you do not have an action or a submit you need to accomplish those functions using JavaScript and Ajax. What you need to do is format the information you want to pass to the processing program (the php program) the way it would be formatted for a post action. To do that we add an onclick event handler reference so that the getRecords function gets executed with the data that was passed.

The first thing it does is format the data that was passed into a string that is compatible with the HTML POST function.

It then use the XMLHttpRequestObject and sets the onreadystatechange property to reference the displayRecords function.

Next it opens the object to do a POST method and sets the destination to donationstodrive.php and says it is doing things asynchronously (that is the true). Then because we are emulating a POST method, we have to set to set the.setRequestHeader object. Finally you execute the send of the request object - we send the string that we formatted in requestStr. That sends the string to the PHP that we specified in the open. We then wait for something to happen. As the state of the request object changes the displayRecords function will be executed. We are only interest in a particular ready state and status of the request object that will tell us it has received a response from the PHP.

```
function getRecords(selectedValue) {
    var requestStr = encodeURI("dlist=" + selectedValue);
    var xmlhttp = getXMLHttpRequestObject();
    xmlhttp.onreadystatechange = function(){
        displayRecords(xmlhttp);
    }
    xmlhttp.open("POST", "donationstodrive.php", true);
    xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
    xmlhttp.send(requestStr);
}

function displayRecords(xmlhttp) {
    if (xmlhttp.readyState==4 && xmlhttp.status==200) {
        document.getElementById("myDiv").innerHTML=xmlhttp.responseText;
    }
}
```

Changes the innerHTML of the empty division in the HTML which is called myDiv.

This is donationstodrive.php.

The post contains 100, 200, or 300 (the value of dlist).

Remember \$_POST is an associative array and the key is dlist and the value is whatever value was chosen by the user.

We then use that value to do a select against the database and if we retrieve records we do a print which sends the information back to the request object which the JavaScript access to change the html (the myDiv empty division will now contain data).

```
<?php
include "includepswd.php";
mysql_select_db("donor");
$dlist = 0;
if (isset($_POST['dlist']))
{
    $dlist = (int) $_POST['dlist'];
}
//echo $dlist;
$sql = "SELECT idno, driveno, condate, contamt FROM donationajax where driveno = $dlist";
$result = mysql_query($sql,$link) or die(mysql_error());
if (mysql_num_rows($result) > 0)
{
    while ($row = mysql_fetch_array($result))
    {
        print ("<br>" . $row["idno"] . " " . $row["driveno"] . " " . $row["contamt"]);
    }
}
?>
```

This is donationAjaxGet.html and the only difference is I am referencing processDataGet.js instead of processDataPost.js.

```
<!DOCTYPE>
<html>
<head>
<meta charset="utf-8">
<title>Donations</title>

<script src="getXMLHttpRequestObject.js" type="text/javascript"></script>
<script src="processDataGet.js" type="text/javascript"></script>

</head>

<body>
<h2>This menu will allow you to select a drive and see the donations made to that drive.</h2>

<form id="listdonation">
<p>
<select id="dlist" name="dlist">
<option value="100">Kids Shelter</option>
<option value="200">Animal Home</option>
<option value="300">Health Aid</option>
</select>
<button type="button" onclick="getRecords()">Retrieve</button>
</p>
</form>
<div id="myDiv"></div>
</body>
</html>
```

This is processDataGet.js.

The difference between this and processDataPost.js is that it formats the request string the way a GET method would as opposed to the way the POST method would.

When you do a GET on a form it tacks the ? and the variable and then the value in quotes. The world can then see the variables.

I had to format the requestStr to be compatible with the method I am emulating. With the POST we just set up a string where we said dlist was equal to the value. In the GET, we have to set up the request string to include the URL of the PHP we want to execute and then the ? and then then the variable and the value.

We do not need the header information that we need with POST because it is not required with GET because data in URL string.

```
function getRecords() {
    var requestStr = "donationstodriveg.php?dlist=" + document.getElementById("dlist").value;
    var xmlhttp = getXMLHttpRequestObject();
    xmlhttp.onreadystatechange = function(){
        displayRecords(xmlhttp);
    }
    xmlhttp.open("GET", requestStr, true);
    xmlhttp.send();
}

function displayRecords(xmlhttp) {
    if (xmlhttp.readyState==4 && xmlhttp.status==200) {
        document.getElementById("myDiv").innerHTML+xmlhttp.responseText;
    }
}
```

[This is donationstodriveg.php](#) and it uses the \$_GET associative array.

```
<?php
include "includepswd.php";
mysql_select_db("donor");
$dlist = 0;
if (isset($_GET['dlist']))
{
    $dlist = (int) $_GET['dlist'];
}
//echo $dlist;
$sql = "SELECT idno, driveno, condate, contamt FROM donationajax where driveno = $dlist";
$result = mysql_query($sql,$link) or die(mysql_error());
if (mysql_num_rows($result) > 0)
{
    while ($row = mysql_fetch_array($result))
    {
        print ("<br>" . $row["idno"] . " " . $row["driveno"] . " " . $row["contamt"]);
    }
}
?>
```