AJAX and PHP

XHTML, Javascript, AJAX, PHP

What is AJAX

AJAX

- **A**synchronous **J**avaScript **A**nd **X**ML
- **AJAX** is only a name given to a set of tools that were previously existing.
- **AJAX** is a technique for creating **fast** and **dynamic** web pages.
- **AJAX** allows web pages to be **updated asynchronously** by exchanging small amounts of data with the server **behind the scenes**. This means that it is possible to update parts of a web page, **without reloading the whole page**.
- Classic web pages, (which do not use AJAX) must reload the entire page if the content should change (this slows down your web page).

http://w3schools.com/ajax/ajax_intro.asp

AJAX

- The processing of web page formerly was only in the server-side, using web services or PHP scripts, before the whole page was sent within the network.
- Ajax allows to perform processing in the client computer (in **JavaScript**) with data taken from the server.
- "**Asynchronous**", means that the response of the server will be processed only when available; not having to wait and to freeze the display of the page.
- Examples of applications using AJAX:
  - **Google Maps (since 2005), Gmail, Youtube, and Facebook tabs**.

http://www.xul.fr/en-xml-ajax.html

AJAX SERVER??

- **No such thing!**

- Server-side applications just need to serve data using **HTTP**
- Clients using **AJAX** framework can communicate with any type of server applications
  - **PHP script**
  - **Java servlet**
  - **JSP**
  - **etc.**

Client-Side Programming

Client-Side Programming

Recall the technologies comprising **DHTML**

1. HTML (**content**)
2. Document Object Model (DOM) (**data structure**)
3. JavaScript (**behaviour**)
4. Cascading Style Sheets (**presentation**)

- **JavaScript** is a powerful tool for dynamic client-side programming

But what if we wanted to frequently communicate with the server?

Recall: Incorporating JavaScipt

Handling browsers that do not support Javascript

```
<html>
<body>
<script type="text/javascript">
  <!--
    document.write("Hello World!");
  //-->
</script>

<noscript>
   <h2>Your Browser doesn't support JavaScript! </h2>
</noscript>
</body>
</html>
```

- use **HTML comments** so that browsers that do not support JavaScript do not display your code

Client-Server Communication

Client-Server Communication

- **JavaScript** runs inside a sandbox attached to the browser

Sequence of steps:

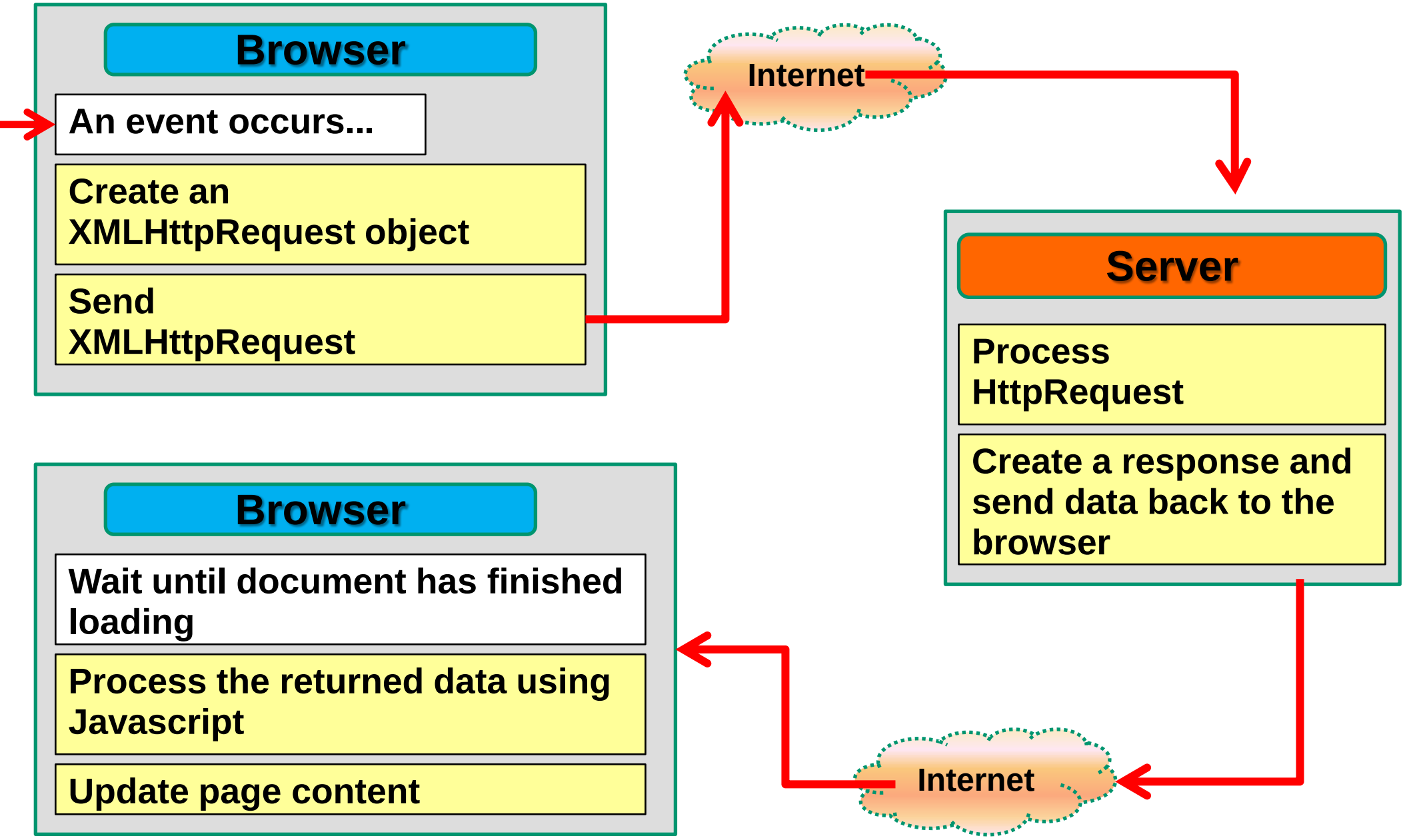
1. **JavaScript** code uses DOM to build **HTML** document.
2. **GET/POST** string is formed.
3. **Browser** encodes **HTML** and **GET/POST** queries into an **HTTP** string.
4. **Browser** establishes contact with server and sends the **HTTP** request.
5. **Browser** receives **HTTP** response from the server and displays the body of the page.

It would be nice if one could write JavaScript code that can **directly communicate** with the **server**

How does AJAX work?

- **AJAX** uses a programming model with **display** and **events**.
- These events are **user actions**, they call **functions** associated with **elements of the web page**.
- **Interactivity** is achieved with **forms** and **buttons**.
- **DOM** allows to link elements of the page with **actions** and also to extract data from **XML** or Text files provided by the server.

How does AJAX work?



How does AJAX work?

To get data on the server, **XMLHttpRequest** provides two methods:

- **open**: create a connection.
- **send**: send a request to the server.

- **Data furnished by the server** will be found in the attributes of the **XMLHttpRequest object**:
  - **responseXml** for an **XML** file or
  - **responseText** for a **plain text**.

XMLHttpRequest object

readyState	The value ranges from **0 to 4,** 4 means "ready".
status	**200** means OK **404** if the page is not found.
responseText	holds loaded data as a string of characters.
responseXml	holds an XML loaded file, DOM's method allows to extract data.
onreadystatechange	property that takes a **function** as value that is invoked when the **readystatechange event** is dispatched.

These are the **properties** of an **XMLHttpRequest object** that we are going to utilise to retrieve a response from the server.

http://www.xul.fr/en-xml-ajax.html

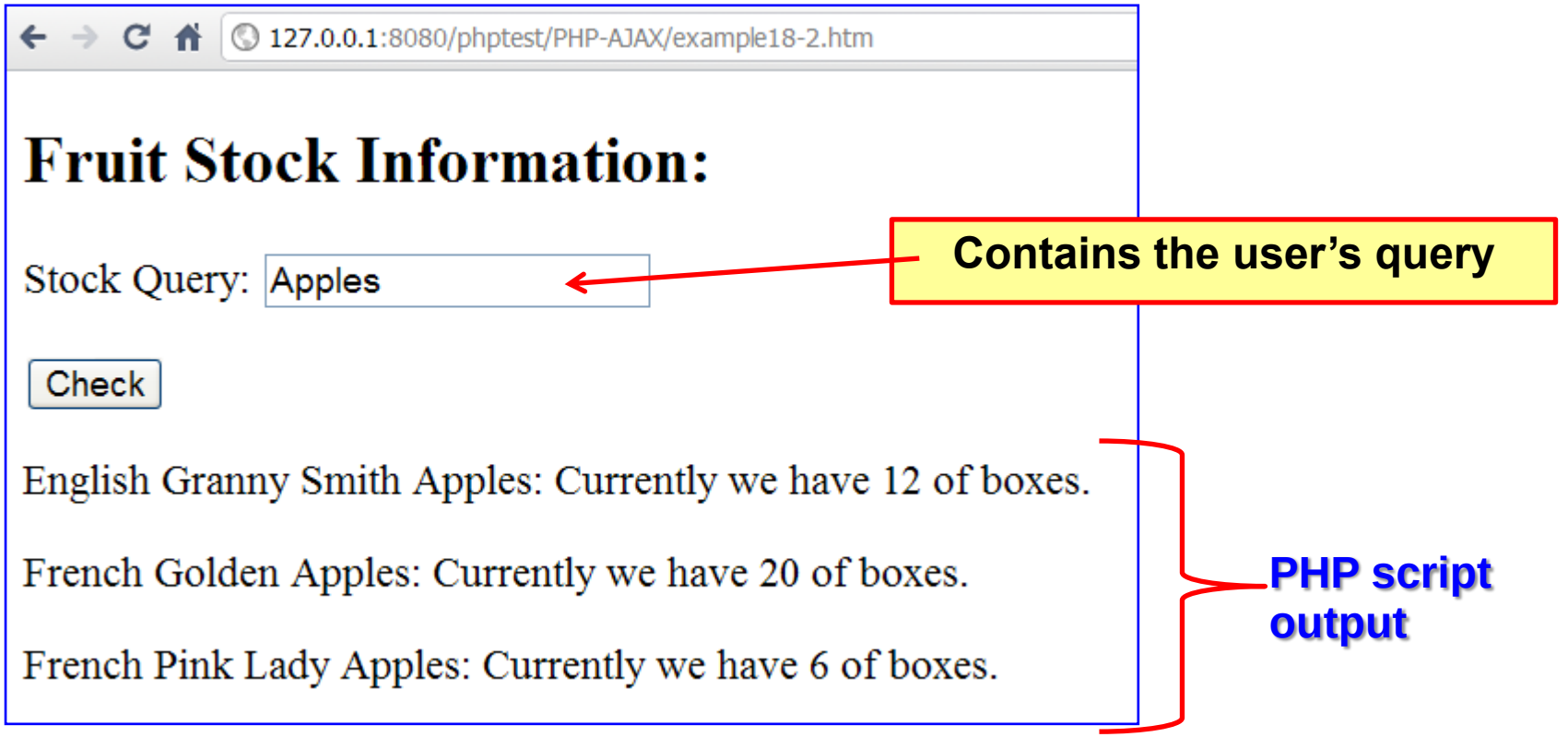
PHP-MySQL-AJAX Example #1

Files

- MySQL database (*.sql)
- PHP script (*.php)
- HTML document (*.htm)
- Javascript (*.js)

Communicates with the MySQL server to retrieve records based on a user's query

Database Stock Example

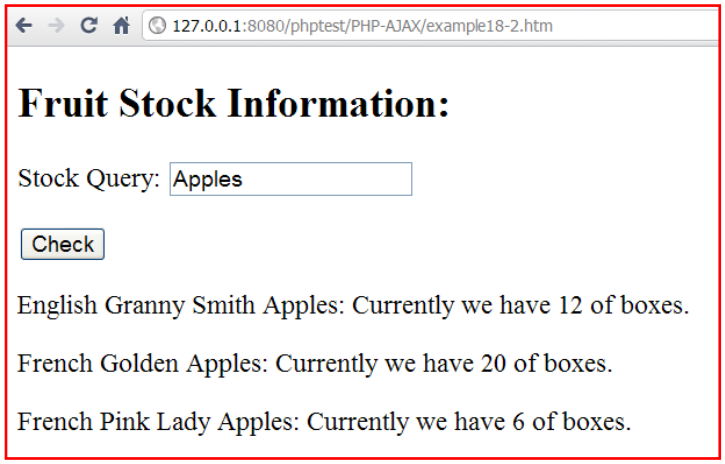


AJAX can be used to **run** **PHP scripts** that obtain up-to-the-minute information stored on a database.

The database is **queried** when the **user interacts** with the application, delivering accurate information without the inconvenience of a page reload.


HTML Document

Database Stock Example

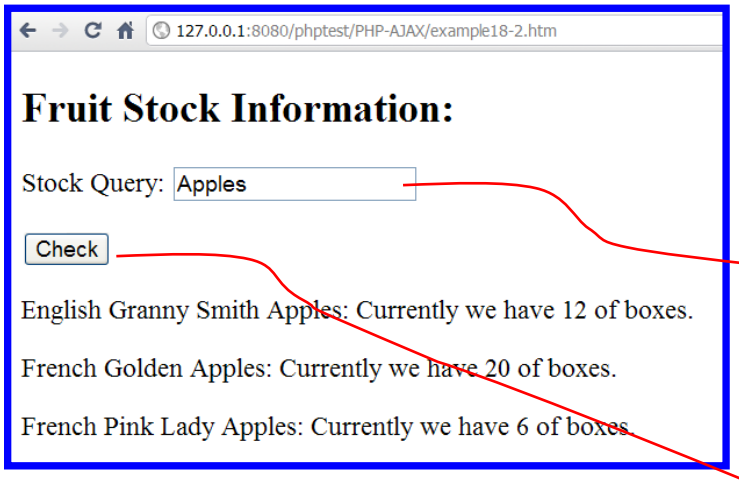


example18-2.htm

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

<head>
<title>Stock Script</title>
<meta http-equiv="Content-Type" content="text/html;
charset=ISO-8859-1" />

<script type="text/javascript" src="getxmlhttprequest.js">
</script>
<script type="text/javascript" src="example18-2.js">
</script>
</head>
...
```

We have two **Javascript files** in our example. They are loaded in the **<head>** section of our HTML file.

Database Stock Example



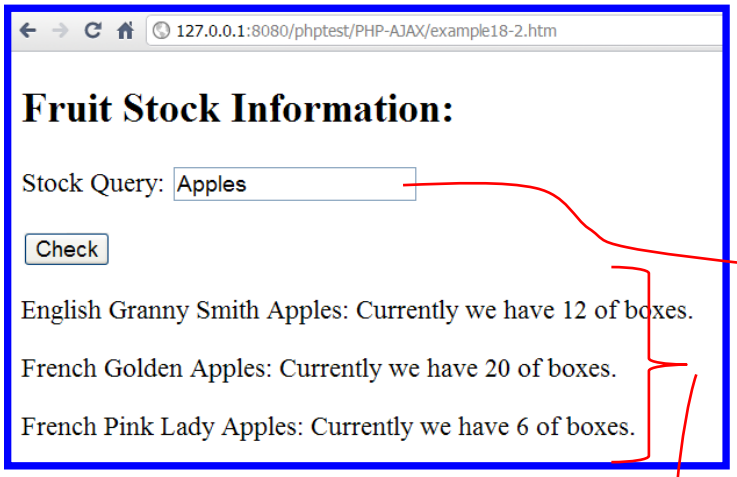
example18-2.htm (continuation…)

```
<body>

<h2>Fruit Stock Information:</h2>
<form action="" method="post">
<p>
<label for="strStock">Stock Query: </label>
<input type="text" name="strStock" id="strStock"/>
</p>

<p>
<input type="button" value="Check" onclick="startJS();"/>
</p>
…
```

We have a query input **text field** and a **submit button**

The **submit button** includes an **onclick event** which invokes the **startJS()** function when clicked **(example18-2.js).**

Database Stock Example



example18-2.htm (continuation…)

```
<body>

<h2>Fruit Stock Information:</h2>
<form action="" method="post">
<p>
<label for="strStock">Stock Query: </label>
<input type="text" name="strStock" id="strStock"/>
</p>

<p>
<input type="button" value="Check" onclick="startJS();"/>
</p>
<div id="strStockResult"></div>
</form>

</body>
</html>
```

The ** ** element defines a **section** used to display the **output** from the **PHP script**.

AJAX (Javascript)

AJAX – connect to server, send request

example18-2.js

```
function startJS() {
    xhrequest = null;
    try {
        xhrequest = getXMLHttpRequest();
    }
    catch(error) {
        document.write("Cannot run Ajax code using this browser");
    }

    if(xhrequest != null) {
    // get form values
      var strStock = document.getElementById("strStock").value;

      var strUrl = "example18-2.php?strStock=" + strStock;

      xhrequest.onreadystatechange = changePage;
      xhrequest.open("GET", strUrl, true);
      xhrequest.send(null);
    }
}
...
```

Checks if AJAX is supported. It checks if the xmlhttprequest object can be created.

Obtain query data entered on the form

PHP script file + User's query

Sets a function that obtains the data output from PHP script

Open a connection to the PHP script, then pass the data

Null because we have appended the query parameters already

startJS() is invoked when the button is clicked.

AJAX – obtain output from server

example18-2.js (continuation…)

Check if data is available

```
function changePage() {
    if (xhrequest.readyState == 4  &&  xhrequest.status == 200) {
        var strResponse = xhrequest.responseText;
        document.getElementById("strStockResult").innerHTML = strResponse;
    }
}
```

Retrieve response from the server

changePage() obtains the data output from the PHP script then stores it into a variable named **strResponse**.

The data is then injected into the **strStockResult ** section defined in the HTML. This is accomplished using the **innerHTML method**.

XMLHttpRequest object

• an **object** used both for making the **XMLHttpRequest** and retrieving the server's **response**

• We have to wait for the data to be available to process it.

• In this purpose, the **state of availability of data** is given by the **readyState** attribute of **XMLHttpRequest**.

States of readyState

0: not initialized.
1: connection established.
2: request received.
3: answer in process.
✓ **4**: finished.

only the last one is really useful!

getXMLHttpRequest() – user-defined

getxmlhttprequest.js

```
function getXMLHttpRequest() {
  var xhrequest = null;

  if(window.XMLHttpRequest) {

   // If IE7, Mozilla, Safari, etc: Use native object
   try  {
      xhrequest = new XMLHttpRequest();
      return xhrequest;
   }
   catch(exception)  {
       // OK, just carry on looking
   }
}
```

The window object represents an open window in a browser.

Check if this property is present

Use native scripting

Continued...

Our Javascript needs to be able to acquire the appropriate type of **XMLHttpRequest** object, depending on the browser the script is running in.

getXMLHttpRequest() – user-defined

getxmlhttprequest.js

```
else {
 // ...otherwise, use the ActiveX control for IE5.x and IE6

  var IEControls = ["MSXML2.XMLHttp.5.0","MSXML2.XMLHttp.4.0",
                    "MSXML2.XMLHttp.3.0","MSXML2.XMLHttp"];
  for(var i=0; i<IEControls.length; i++) {

   try {
      xhrequest = new ActiveXObject(IEControls[i]);
      return xhrequest;
   }
   catch(exception) {
    // OK, just carry on looking
   }

  }
 // if we got here we didn't find any matches
 throw new Error("Cannot create an XMLHttpRequest");
 }
}
```

Testing is done starting from the most recent backwards.

Microsoft has developed different implementations of the XMLHttpRequest object over time.

ActiveXObject is an older version implementation.

PHP Script

PHP Script

example18-2.php

```
<?php
$strStock = $_GET["strStock"];

$dbLocalhost = mysql_connect("localhost", "root", "") or die("Could not connect: " . mysql_error());

mysql_select_db("stock", $dbLocalhost )  or die("Could not find database: " . mysql_error());

$dbRecords = mysql_query("SELECT * FROM stock WHERE Name='$strStock' ",
$dbLocalhost ) or die("Problem reading table: " . mysql_error());

$intRecords = mysql_num_rows($dbRecords );

if ($intRecords == 0)
      echo "<p>Stock Item '$strStock' Unknown.</p>";
else {
      while ($arrRecords = mysql_fetch_array($dbRecords)) {
            $strDescription = $arrRecords ["Description"];
            $intQuantity = $arrRecords["Quantity"];
            echo "<p>$strDescription: Currently we have $intQuantity of boxes.</p>";
      }
}
?>
```

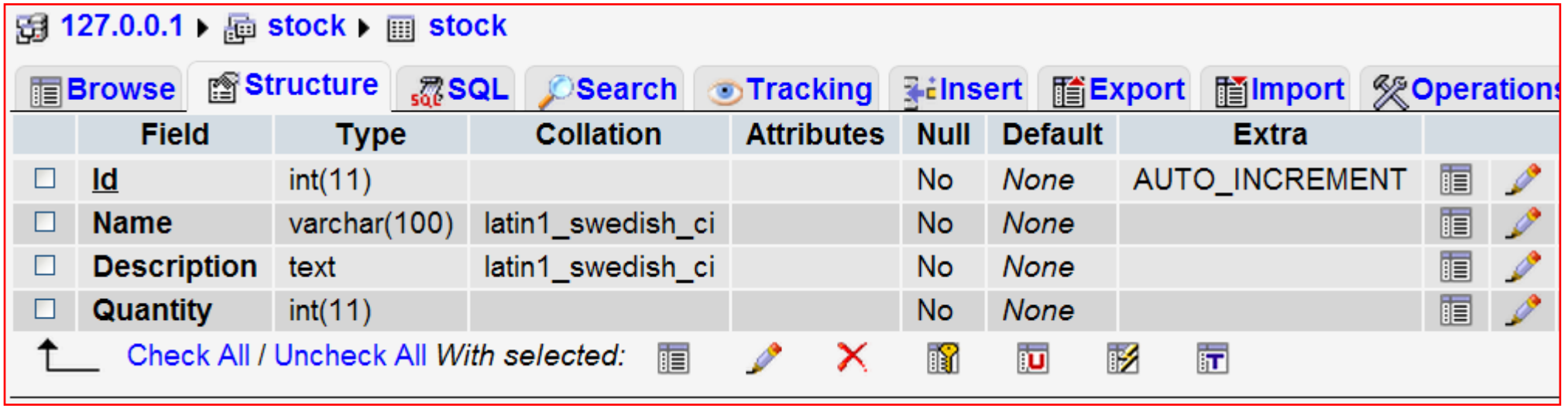
Contains the user's query

Table named stock

- Queries the database and outputs the corresponding records found

MySQL Database

Stock Table (Structure)



127.0.0.1 ▸ stock ▸ stock

Browse | Structure | SQL | Search | Tracking | Insert | Export | Import | Operations

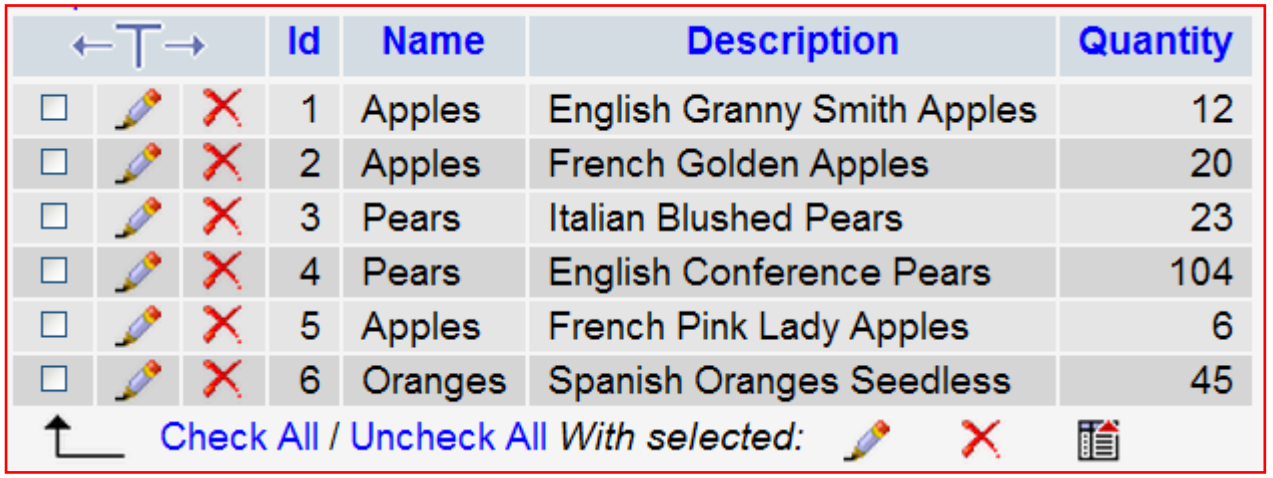
	Field	Type	Collation	Attributes	Null	Default	Extra
☐	Id	int(11)			No	*None*	AUTO_INCREMENT
☐	Name	varchar(100)	latin1_swedish_ci		No	*None*	
☐	Description	text	latin1_swedish_ci		No	*None*	
☐	Quantity	int(11)			No	*None*	

Check All / Uncheck All *With selected:*

Id is a **primary key**, also set to **auto_increment.**

You need to **create** your database first using **phpMyAdmin**, then import the **stock.sql** file containing the structure and data entries.

Stock Table (data)



	Id	Name	Description	Quantity
☐	1	Apples	English Granny Smith Apples	12
☐	2	Apples	French Golden Apples	20
☐	3	Pears	Italian Blushed Pears	23
☐	4	Pears	English Conference Pears	104
☐	5	Apples	French Pink Lady Apples	6
☐	6	Oranges	Spanish Oranges Seedless	45

Check All / Uncheck All *With selected:*

- You can populate the database easily using **phpMyAdmin**.
- You can import the **stock.sql** file containing the structure and initial data entries.
- You can select the **INSERT** tag to add more data entries.

Example #2: thumbnails with zoom-in feature

Files

- Images (*.jpg)
- PHP script (*.php)
- HTML document (*.htm)
- Javascript (*.js)

Zooming photo thumbnail application

- user is presented with a **series** of **small thumbnails** of photos
- Upon moving a **mouse over** one of the thumbnails, a **larger image** is displayed.

Zooming photo thumbnail application

• Using standard Javascript and (X)HTML would require all of the images to be downloaded whenever a user moves a mouse over an image.

• Using **AJAX**, only the images that the user wishes to zoom in on are downloaded and a **full page refresh is avoided**.


CSS-style

CSS float: applied on an Image

Image_float.htm

```
<html>
<head>
<style type="text/css">
img
{
   float:right;
}
</style>
</head>

<body>
<p>In the paragraph below, we have added an image with style <b>float:right</b>. The result is that the image will float to the right in the paragraph.</p>
<p>
<img src="logocss.gif" width="95" height="84" />
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>
</body>
</html>
```

CSS-style definition

Image file

- Elements are floated horizontally, this means that an element can only be **floated left** or **right**, not up or down.

CSS float: applied on an Image

In the paragraph below, we have added an image with style **float:right**. The result is that the image will float to the right in the paragraph.

This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.


HTML Document

html

Example18-3.htm

```
<script type="text/javascript" src="getxmlhttprequest.js">
</script>
<script type="text/javascript" src="example18-3.js">
</script>

<style type="text/css">

#big {
   float: left;
}

#small {
   float: left;
   width: 320px;
}

</style>
</head>
```

CSS-style definition

Image file

• Embedded CSS-style definition is included to define the Style of the small thumbnail photos and the larger zoomed image.

html

Example18-3.htm (continuation…)

```
<h2>Zooming Pictures:</h2>
<div id="small">
<img src="graphics/1s.jpg" onmouseover="startJS(1);" alt="small picture"/>
<img src="graphics/2s.jpg" onmouseover="startJS(2);" alt="small picture"/>
<img src="graphics/3s.jpg" onmouseover="startJS(3);" alt="small picture"/>
<img src="graphics/4s.jpg" onmouseover="startJS(4);" alt="small picture"/>
<img src="graphics/5s.jpg" onmouseover="startJS(5);" alt="small picture"/>
<img src="graphics/6s.jpg" onmouseover="startJS(6);" alt="small picture"/>
<img src="graphics/7s.jpg" onmouseover="startJS(7);" alt="small picture"/>
<img src="graphics/8s.jpg" onmouseover="startJS(8);" alt="small picture"/>
<img src="graphics/9s.jpg" onmouseover="startJS(9);" alt="small picture"/>
<img src="graphics/10s.jpg" onmouseover="startJS(10);" alt="small picture"/>
<img src="graphics/11s.jpg" onmouseover="startJS(11);" alt="small picture"/>
<img src="graphics/12s.jpg" onmouseover="startJS(12);" alt="small picture"/>
</div>

<div id="big"><img src="graphics/1l.jpg" width='600' alt="large picture"/></div>
</body>
</html>
```

Large image

The alt attribute provides alternative information for an image if a user for some reason cannot view it

- Each of the thumbnail images is displayed within a division with an id “small”.

AJAX (Javascript)

AJAX – connect to server, send request

example18-2.js

```
function startJS(intPicture) {
    xhrequest = null;
    try {
        xhrequest = getXMLHttpRequest();
    }
    catch(error) {
        document.write("Cannot run Ajax code using this browser");
    }

    if(xhrequest != null) {
    // get form values


      var strUrl = "example18-3.php?intPicture=" + intPicture;

      xhrequest.onreadystatechange = changePage;
      xhrequest.open("GET", strUrl, true);
      xhrequest.send(null);
    }
}
...
```

Contains a number representing the image the mouse has moved over.

Checks if AJAX is supported. It checks if the xmlhttprequest object can be created.

PHP script file + User's query

Sets a function that obtains the data output from PHP script

Open a connection to the PHP script, then pass the data

Null because we have appended the query parameters already

startJS() is invoked when the button is clicked.

AJAX – obtain output from server

example18-3.js (continuation…)

Check if data is available

```
function changePage() {
    if (xhrequest.readyState == 4  &&  xhrequest.status == 200) {
        var strResponse = xhrequest.responseText;
        document.getElementById("big").innerHTML = strResponse;
    }
}
```

Retrieve response from the server

changePage() obtains the data output from the PHP script then stores it into a variable named **strResponse**.

The data is then injected into the **strStockResult ** section defined in the HTML. This is accomplished using the **innerHTML method**.

PHP Script

PHP Script

example18-3.php

```
<?php

$intPicture = $_GET["intPicture"];

echo "<img src='graphics/$intPicture" . "l.jpg' width='600' />";

?>
```

Name retrieved via GET

Append an Extension to the file name

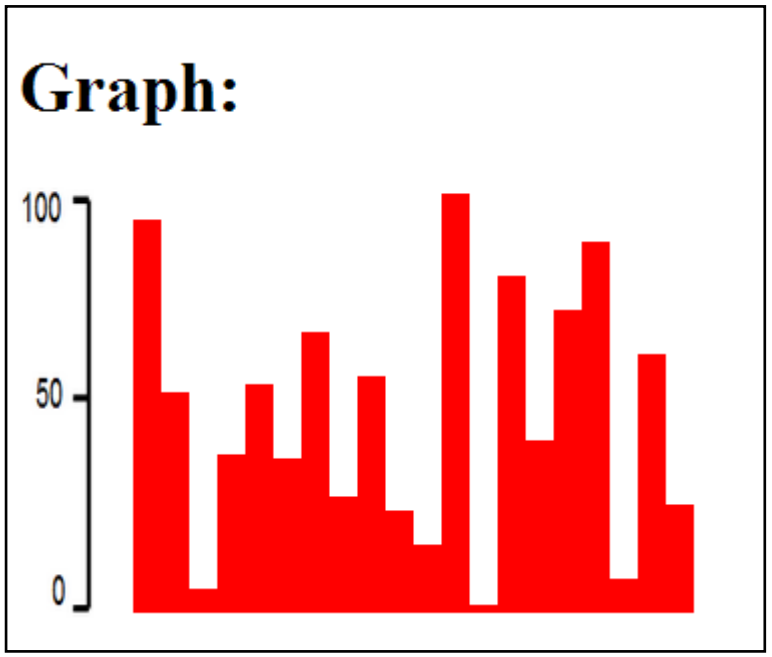
- the PHP script obtains the value of the image the mouse has moved over, passed via the **GET method** and stores it in a variable called **$intPicture**.

- It then outputs the (X)HTML element pointing to the corresponding large image.

Example #3: dynamic histogram

Files

- Images (*.jpg)
- MySQL (*.sql)
- PHP script (*.php)
- HTML document (*.htm)
- Javascript (*.js)



Consider developing a website that displays a **graph** showing **in almost real-time** some data about your customers, sales, etc.

Using **AJAX**, we can **minimise the effect** of that continuously updating graph on the users of your website.


HTML Document

html

Example18-4.htm

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

<head>
<title>Graph Script</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1" />

<script type="text/javascript" src="getxmlhttprequest.js">
</script>
<script type="text/javascript" src="example18-4.js">
</script>
<style type="text/css">
#graphBars {
    float: left;
}

#graphScale {
        float: left;
        width: 40px;
}
</style>
</head>
```

CSS-style definition

Defines a CSS-style for an Image file that will display the scale of the chart

- Embedded CSS-style definition is included to define the Style of the graphing area

html

Example18-4.htm (continuation…)

```
<body onload="startJS();">

    <h2>Graph:</h2>

    <div id="graph">

    </div>

</body>
</html>
```

when the web page is loaded, an onload event invokes the startJS() function

a section that defines the graphing area

- The ** ** section will be populated programmatically via a combination of php script and AJAX.

AJAX (Javascript)

AJAX – connect to server, send request

example18-4.js

```
function startJS( ) {
    xhrequest = null;
    try {
        xhrequest = getXMLHttpRequest();
    }
    catch(error) {
        document.write("Cannot run Ajax!");
    }

    if(xhrequest != null) {
    // get form values
        var objDate = new Date();
        var intSecs = objDate.getTime();

        var strUrl = "example18-4.php?intSecs=" + intSecs;
            xhrequest.onreadystatechange = changePage;
            xhrequest.open("GET", strUrl, true);
            xhrequest.send(null);
            setTimeout("startJS()", 500);
        }   }
}
...
```

Checks if AJAX is supported. It checks if the xmlhttprequest object can be created.

A Data object is created and used to return the number of seconds elapsed since 1/1/1970 using the getTime() function

We want to force the browser to obtain new data via the PHP script

Sets a function that obtains the data output from PHP script

Open a connection to the PHP script, then pass the data

Call startJS() function (itself!) after 500 milliseconds

startJS() is invoked when the button is clicked.

AJAX – obtain output from server

example18-4.js (continuation…)

Check if data is available

```
function changePage() {
    if (xhrequest.readyState == 4  &&  xhrequest.status == 200) {
        var strResponse = xhrequest.responseText;
        document.getElementById("graph").innerHTML = strResponse;
    }
}
```

Retrieve response from the server

changePage() obtains the data output from the PHP script then stores it into a variable named **strResponse**.

The data is then injected into the **graph ** section defined in the HTML. This is accomplished using the **innerHTML method**.

PHP Script

PHP Script

example18-4.php

```
<?php
$dbLocalhost = mysql_connect("localhost", "root", "") or die("Could not connect: " . mysql_error());
mysql_select_db("graph", $dbLocalhost )  or die("Could not find database: " . mysql_error());

srand((double) microtime() * 1000000);
$intPercentage = rand(0,99);

$dbWriteRecords= mysql_query("INSERT INTO percentageValues VALUES ('',
'$intPercentage')", $dbLocalhost) or die("Problem reading table: " . mysql_error());

$dbRecords = mysql_query("SELECT * FROM percentageValues", $dbLocalhost )
or die("Problem reading table: " . mysql_error());
$intCount = mysql_num_rows($dbRecords );

if ($intCount > 20) {
   $intStart = $intCount - 20;
   $dbRecords = mysql_query(" SELECT * FROM percentageValues LIMIT $intStart, 20",
$dbLocalhost)  or die("Problem reading table: " . mysql_error());
}
...
```

Generate a random number between [0, 99]

Table named percentageValues

Last 20 entries

Continued...

• the PHP script queries (INSERT & SELECT) the database and outputs the corresponding records found.

PHP Script

example18-4.php (continuation…)

```
$arrPercent = array (0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0);
$intSize = count($arrPercent);
$intCount = 0;
while ($arrRecords= mysql_fetch_array($dbRecords)) {
 $arrPercent[$intCount++] = $arrRecords["Percentage"];
}
graph($arrPercent, $intSize);

function graph($arrData, $intSize) {
    $intBarWidth = 10;
//    $intBarSpacing = 10;
    $intMultiplier = 1.5;

    $intSize = count($arrData);
    echo "<div id='graphScale'><img src='graphics/scale.gif' width='27' height='150' /></div>";
    echo "<div id='graphBars'>";
    echo "<img src='graphics/hiddenbar.gif' width='0' height='" . 99 * $intMultiplier . "'>";
    for($intCount=0; $intCount<$intSize; $intCount++) {
        echo "<img src='graphics/redbar.gif' width='$intBarWidth' height='" . $arrData[$intCount]
            * $intMultiplier . "'>";
    }
    echo "</div>";
}
?>
```

Fetch the records

A field in the table

Generate a graph by displaying an image repetitively

- the graph function uses 2 sections to define the scale and the graphing area.


jquery

jquery

- **jQuery** is a library of JavaScript Functions.
- **jQuery** is a lightweight "write less, do more" **JavaScript library**.
- The **jQuery** library contains the following features:
  - HTML element selections
  - HTML element manipulation
  - CSS manipulation
  - HTML event functions
  - JavaScript **Effects** and **animations**
  - HTML DOM traversal and modification
  - **AJAX**
  - Utilities

http://w3schools.com/jquery/jquery_intro.asp

jquery

- The jQuery library is stored in a **single** **JavaScript file**, containing all the **jQuery functions**.

- It can be added to a web page with the following mark-up:

```
<head>
    <script type="text/javascript" src="jquery.js"></script>
</head>
```

Jquery library

Note: **the <script> tag should be inside the page's <head> section.**

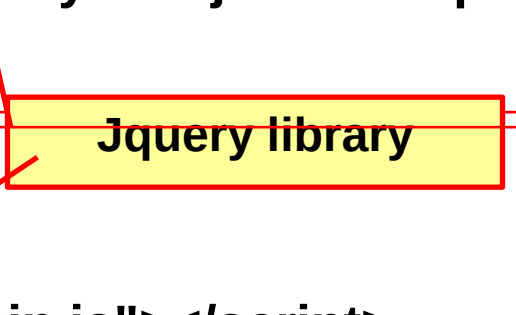
http://w3schools.com/jquery/jquery_intro.asp

Jquery library

If you don't want to store the jQuery library on your own computer, you can use the hosted jQuery library from **Google** or **Microsoft**.

```
<head>
<script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js"></script>
</head>
```

```
<head>
<script type="text/javascript"
src="http://ajax.microsoft.com/ajax/jquery/jquery-1.4.2.min.js"></script>
</head>
```



Note: the <script> tag should be inside the page's <head> section.

http://w3schools.com/jquery/jquery_intro.asp

Recall: CSS, ID selector, class selector

jQuery uses **CSS selectors** to select **HTML elements**.

Recall: 1. Incorporating CSS

Example: External style sheet

mystyle.css

```
hr {color:sienna;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
```

myHTML.htm

```
<head>
   <link rel="stylesheet" type="text/css" href="mystyle.css”>
</head>
```

Recall: 1. Applying CSS to an HTML element

•You can apply CSS-style formatting to an HTML element either by using an **ID selector** or a **Class selector**.

ID SELECTOR

•The id selector is used to specify a **style for a single, unique element**.
•The id selector uses the id attribute of the HTML element, and is defined with a "#".
•The style rule below will be applied to the element with id="para1":
•Do NOT start an ID name with a number! It will not work in Mozilla/Firefox.

General Syntax:

```
#ID-name
{
  Property:value;
  Property:value;
  /*... and so on.. */
}
```

ID selector application:

```
<h1 id=”ID-name”>  Internet programming </h1>
```

Recall: 2a. Applying CSS to an HTML element

•You can apply CSS-style formatting to an HTML element either by using an **ID selector** or a **Class selector**.

class SELECTOR

• The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements.
• This allows you to set a particular style for any HTML elements with the same class.
• The class selector uses the HTML class attribute, and is defined with a "."

General Syntax:

```
.class-name
{
   Property:value;
   Property:value;
   /*... and so on.. */
}
```

class selector application:

```
<h1 class="class-name"> Internet programming </h1>
<p class="class-name"> Dynamic HTML: CSS, HTML, DOM, Javascript </p>
```

Recall: 2b. Applying CSS to an HTML element

•You can apply CSS-style formatting to an HTML element either by using an **ID selector** or a **Class selector**.

class SELECTOR

• You can also specify that only specific HTML elements should be affected by a class.

General Syntax:

```
/* this class selector is only applicable to paragraphs*/
p.class-name
{
   Property:value;
   Property:value;
   /*... and so on.. */
}
```

class selector application:

```
<p class="class-name">  Dynamic HTML: CSS, HTML, DOM, Javascript </p>
```

jquery

- The jQuery library is stored a **single** **JavaScript file**, containing all the **jQuery functions**.

$(this).hide()
Demonstrates the jQuery hide() function, hiding the current HTML element.

$("#test").hide()
Demonstrates the jQuery hide() function, hiding the element with id="test".

$("p").hide()
Demonstrates the jQuery hide() function, hiding all <p> elements.

$(".test").hide()
Demonstrates the jQuery hide() function, hiding all elements with class="test".

http://w3schools.com/jquery/jquery_syntax.asp

Document Ready Function

- to prevent any jQuery code from running before the document is finished loading (**is it ready?**)

```
$(document).ready(function(){

   // jQuery functions go here...

});
```

Jquery example #1

jquery

jquery1.htm

```
<html>
<head>

<script type="text/javascript" src="jquery-1.4.2.min.js"></script>
<script type="text/javascript">
$(document).ready( function() {
  $("button").click( function() {
    $("div").load('test1.txt');
  } );
} );
</script>
</head>
<body>

<div><h2>Let AJAX change this text</h2></div>
<button>Change Content</button>

</body>
</html>
```

Jquery library

On click event, load the contents from test1.txt into the div section

Modify the contents of the section

http://docs.jquery.com/Downloading_jQuery#Current_Release

jquery

- The jQuery library is stored as a **single** **JavaScript file**, containing all the **jQuery functions**.

The jQuery syntax is tailor made for selecting HTML elements and perform some action on the element(s).

Basic syntax is: **$(selector).action()**

- **$** (dollar sign) to define jQuery
- **(selector)** to "query (or find)" **HTML elements**
- jQuery **action()** to be performed on the element(s)

http://w3schools.com/jquery/jquery_syntax.asp

Jquery example #2

jquery

Jquery2-slide.htm

```
<html>
<head>
<script type="text/javascript" src="jquery-1.4.2.min.js"></script>
<script type="text/javascript">
$(document).ready(function(){
$(".flip").click(function(){
    $(".panel").slideToggle("slow");
  });
});
</script>
 <style type="text/css">
div.panel,p.flip
{
margin:0px;
padding:5px;
text-align:center;
background:#e5eecc;
border:solid 1px #c3c3c3;
}
div.panel
{
height:120px;
display:none;
}
</style>
</head>
 …
```

Jquery library

On click event, (toggle) slide-up or slide-down the html element of class panel

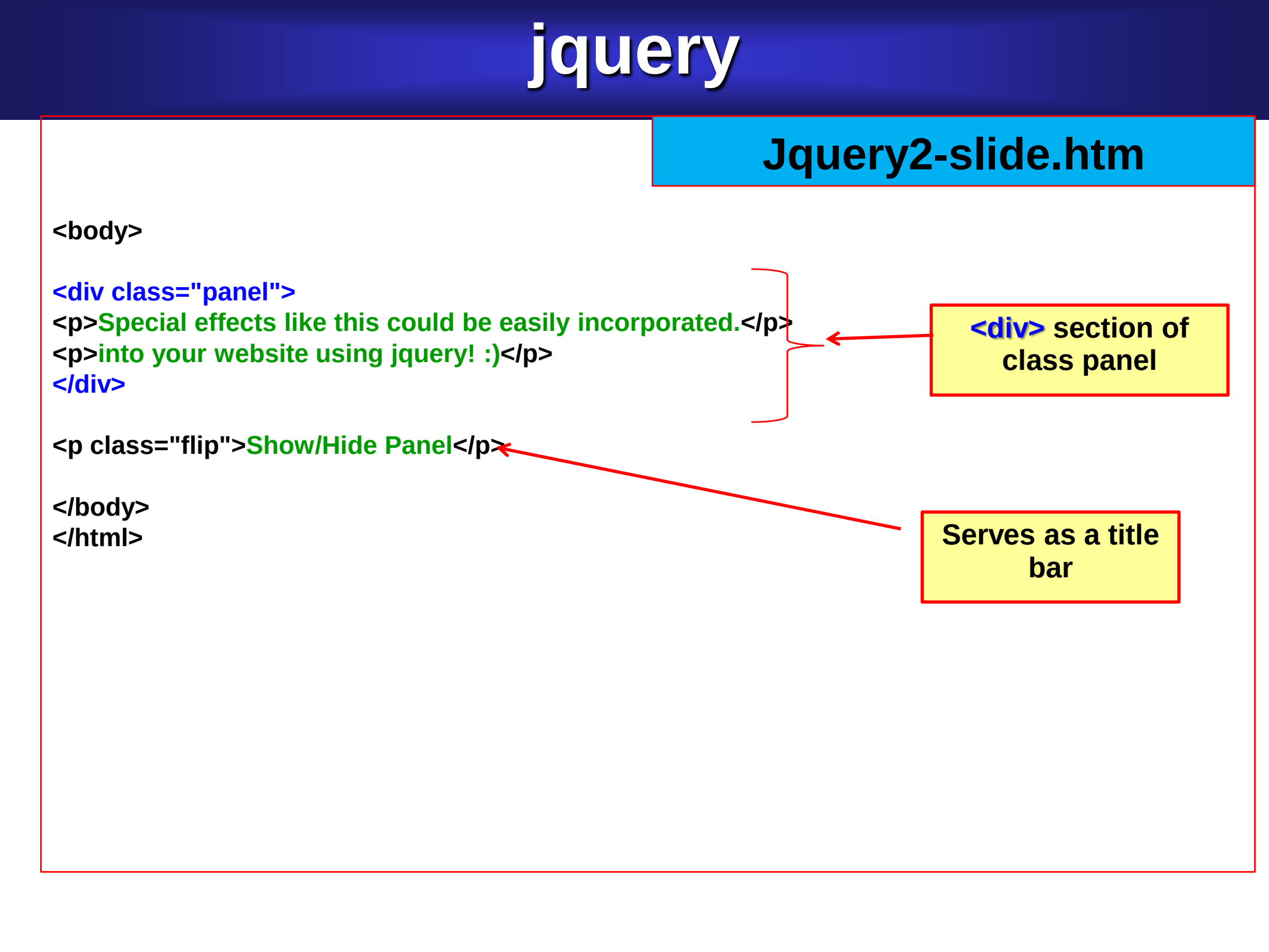
Settings for both the panel and flip classes

Do not show the panel initially

jquery

Jquery2-slide.htm

```
<body>

<div class="panel">
<p>Special effects like this could be easily incorporated.</p>
<p>into your website using jquery! :)</p>
</div>

<p class="flip">Show/Hide Panel</p>

</body>
</html>
```



jquery

- It is recommended that you put your jQuery functions in a separate **.js** file

References

Dynamic Web Application Development using PHP and MySQL – by Stobart and Parsons

http://w3schools.com/ajax/ajax_intro.asp

http://www.xul.fr/en-xml-ajax.html

http://www.jquery.com

http://w3schools.com/jquery