* function loadDoc(url, cFunction) {

var xhttp;

xhttp=new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

cFunction(this);

}

};

xhttp.open("GET", url, true);

xhttp.send();

}

function myFunction(xhttp) {

document.getElementById("demo").innerHTML =

xhttp.responseText;

}

* **Initialize Variables**
  + if (window.XMLHttpRequest) {

//code for modern browsers

xmlhttp = new XMLHttpRequest();

}

else {

//code for old IE browsers

xmlhttp = new ActiveXObject(“Mircosoft.XMLHTTP”);

}

* **GET vs POST**
  + GET is simpler and faster than POST, and can be used in most cases.

However, always use POST requests when:

* + - A cached file is not an option (update a file or database on the server).
    - Sending a large amount of data to the server (POST has no size limitations).
    - Sending user input (which can contain unknown characters), POST is more robust and secure than GET.
  + GET
    - xhttp.open("GET", "demo\_get.asp", true);  
      xhttp.send();//might get a cached result
    - xhttp.open("GET", "demo\_get.asp?t=" + Math.random(), true);  
      xhttp.send();//avoid cached results with unique ID
    - xhttp.open("GET", "demo\_get2.asp?fname=Henry&lname=Ford", true);  
      xhttp.send();//How to send information with the GET method
  + POST
    - xhttp.open("POST", "demo\_post.asp", true);//true means asynchronous  
      xhttp.send();
    - xhttp.open("POST", "ajax\_test.asp", true);  
      xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");  
      xhttp.send("fname=Henry&lname=Ford");
* **.onreadystatechange**
  + xhttp.onreadystatechange = function() {

if(this.readyState == 4 && this.status == 200) {

document.getElementById(“demo”).innerHTML = this.responseText;

}

}//executes when the request receives an answer

* **.readyState**
  + this.readyState == /\*The status of the XMLHttpRequest\*/
    - 0: request not initilaized
    - 1: server connection established
    - 2: request received
    - 3: processing request
    - 4: request finished and response is ready
  + The onreadystatechange function is called every time the readyState changes.
* **.status**
  + this.status == /\*The status of the XMLHttpRequst object\*/
    - 200: “OK”
    - 403: “Forbidden”
    - 404: “Page not found”
    - <https://www.w3schools.com/tags/ref_httpmessages.asp>
* **Callback Function**
  + If you have more than one AJAX task in a website, you should create one function for executing the XMLHttpRequest object, and one callback function for each AJAX task.
  + <https://www.w3schools.com/js/js_ajax_http_response.asp>
* **Server Response Properties**
  + responseText - get the response data as a string
    - document.getElementById(“demo”).innerHTML = xhttp.responseText;
  + responseXML - get the response data as XML data
    - var xhttp, xmlDoc, txt, x, i;

xhttp = new XMLHttpRequest();

xhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

xmlDoc = this.responseXML;

txt = "";

x = xmlDoc.getElementsByTagName("ARTIST");

for (i = 0; i < x.length; i++) {

txt = txt + x[i].childNodes[0].nodeValue + "<br>";

}

document.getElementById("demo").innerHTML = txt;

}

};

xhttp.open("GET", "cd\_catalog.xml", true);

xhttp.send();

* **childNodes**
  + The nodes in the collection are sorted as they appear in the source code and can be accessed by index numbers.
  + For statement is the only way to display all o f the nodes
    - c = document.body.childNodes;

for statement

txt = txt + c[ i ].nodeName + “<br>”;

* + Basically line numbers, including whitespace lines if:

var c = document.getElementById(“myDIV”).childNodes.length;

* + var c = document.getElementById(“mySelect”).childNodes;

document.getElementById(“demo”).innerHTML = c[2].text;

* + xmlDoc = this.responseXML;

x = xmlDoc.getElementsByTagName(“ARTIST”);

for (i = 0; i < x.length; i++) {

txt = txt + x[ i ].childNodes[0].nodeValue + “<br>”;

}

document.getElementById(“demo”).innerHTML = txt;

* + getAllResponseHeaders()
    - returns all the header information of a resource, like length….
    - document.getElementById(“demo”).innerHTML = this.getAllResponseHeaders();//all the header information.
  + getResponseHeader()
    - return specific header information from a resource
    - $(#demo).innerHTML = this.getResponseHeader(“Last-Modified”);
* Ajax Methods
  + animate()
    - $(“button”).click(function() {

$(“#box”).animate({height:”300px”});

});

* + - (selector).animate({styles}, speed, easing, callback);
    - <https://www.w3schools.com/jquery/eff_animate.asp>
  + clearQueue()
    - $(“button”).click(function() {

$(“div”).clearQueue();

});

* + - Removes all items from the queue that have not yet been run. Example, if you have multiple animate items in a que it will stop after the current method stops.
  + delay()
    - $(“#div1”).delay(“slow”/”fast”/\*milliseconds\*).fadeIn();//will have a delay before the .fadeIn() executes
* **PHP Example**
  + <html>  
     <head>  
     <script>  
     function showHint(str) {  
     if (str.length == 0) {   
     document.getElementById("txtHint").innerHTML = "";  
     return;  
     } else {  
     var xmlhttp = new XMLHttpRequest();  
     xmlhttp.onreadystatechange = function() {  
     if (this.readyState == 4 && this.status == 200) {  
     document.getElementById("txtHint").innerHTML = this.responseText;  
     }  
     };  
     xmlhttp.open("GET", "gethint.php?q=" + str, true);  
     xmlhttp.send();  
     }  
     }  
     </script>  
     </head>  
     <body>  
       
     <p><b>Start typing a name in the input field below:</b></p>  
     <form>   
     First name: <input type="text" onkeyup="showHint(this.value)">  
     </form>  
     <p>Suggestions: <span id="txtHint"></span></p>  
     </body>  
    </html>
  + <?php  
     // Array with names  
     $a[] = "Anna";  
     $a[] = "Brittany";  
     $a[] = "Cinderella";  
     $a[] = "Diana";  
     $a[] = "Eva";  
     $a[] = "Fiona";  
     $a[] = "Gunda";  
     $a[] = "Hege";  
     $a[] = "Inga";  
     $a[] = "Johanna";  
     $a[] = "Kitty";  
     $a[] = "Linda";  
     $a[] = "Nina";  
     $a[] = "Ophelia";  
     $a[] = "Petunia";  
     $a[] = "Amanda";  
     $a[] = "Raquel";  
     $a[] = "Cindy";  
     $a[] = "Doris";  
     $a[] = "Eve";  
     $a[] = "Evita";  
     $a[] = "Sunniva";  
     $a[] = "Tove";  
     $a[] = "Unni";  
     $a[] = "Violet";  
     $a[] = "Liza";  
     $a[] = "Elizabeth";  
     $a[] = "Ellen";  
     $a[] = "Wenche";  
     $a[] = "Vicky";  
       
     // get the q parameter from URL  
     $q = $\_REQUEST["q"];//how to get stuff out of the url  
       
     $hint = "";  
       
     // lookup all hints from array if $q is different from ""   
     if ($q !== "") {  
     $q = strtolower($q);  
     $len=strlen($q);  
     foreach($a as $name) {  
     if (stristr($q, substr($name, 0, $len))) {  
     if ($hint === "") {  
     $hint = $name;  
     } else {  
     $hint .= ", $name";  
     }  
     }  
     }  
     }  
      
     // Output "no suggestion" if no hint was found or output correct values   
     echo $hint === "" ? "no suggestion" : $hint;  
    ?>