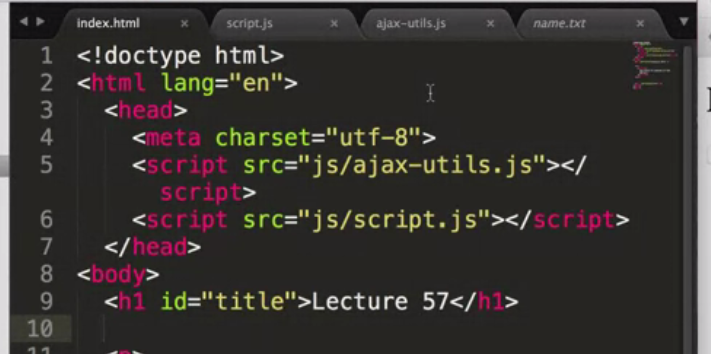


Ajax-utils.js must be placed before script.js.



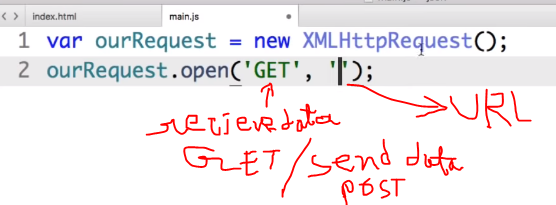
For first example: see video “basic”

1.Sending and receiving data on fly without a page reload is known

As ajax

Web beowser have a built in tool name XMLHttpRequest will establish a connection with a URL that we specify(allow-> send or receive data. )

Now web browser expect to use method called: .open(‘GET’, ‘URL’);



Here after the url we should give : , true

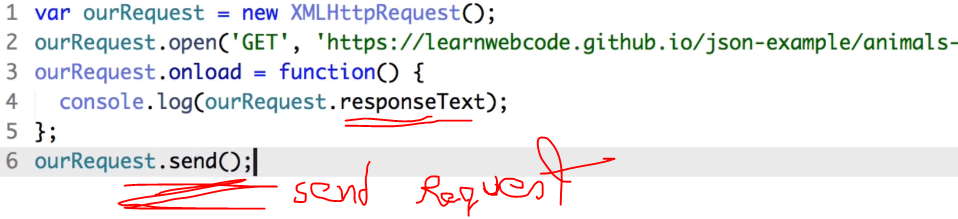
.open(‘GET’, ‘URL’, true);

To have the text which we get from the url :we use

.responseText

Finally we have to send the request.

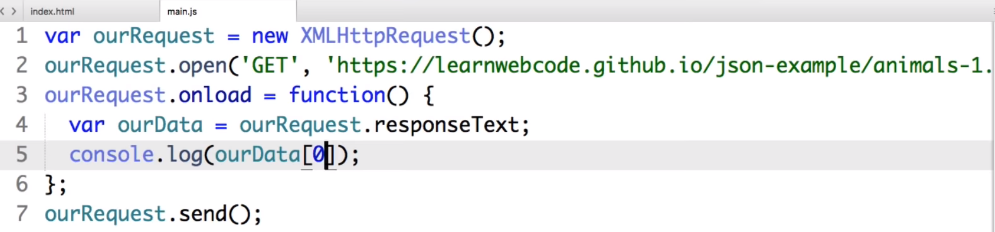
.send() or send(null);



Now if we run the function by: ourRequest.onload();

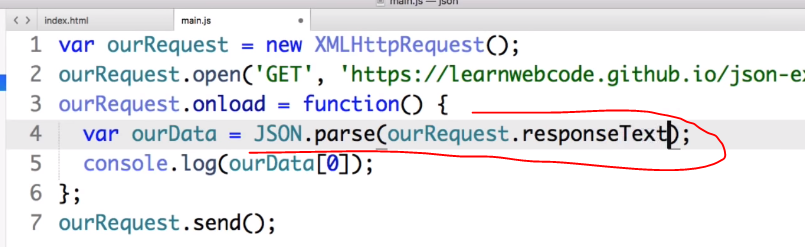
We will see the output in browser console:

Now inside onload function() we can save the data in a variable to use it.



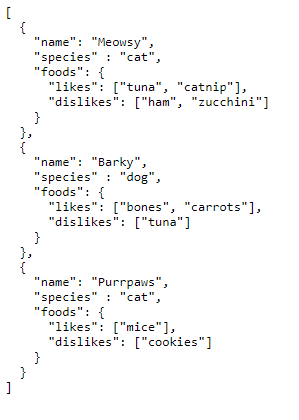
But here the output will be a plane text. Not object.  
because .respnseText allows to read the data as a plane text not as object.

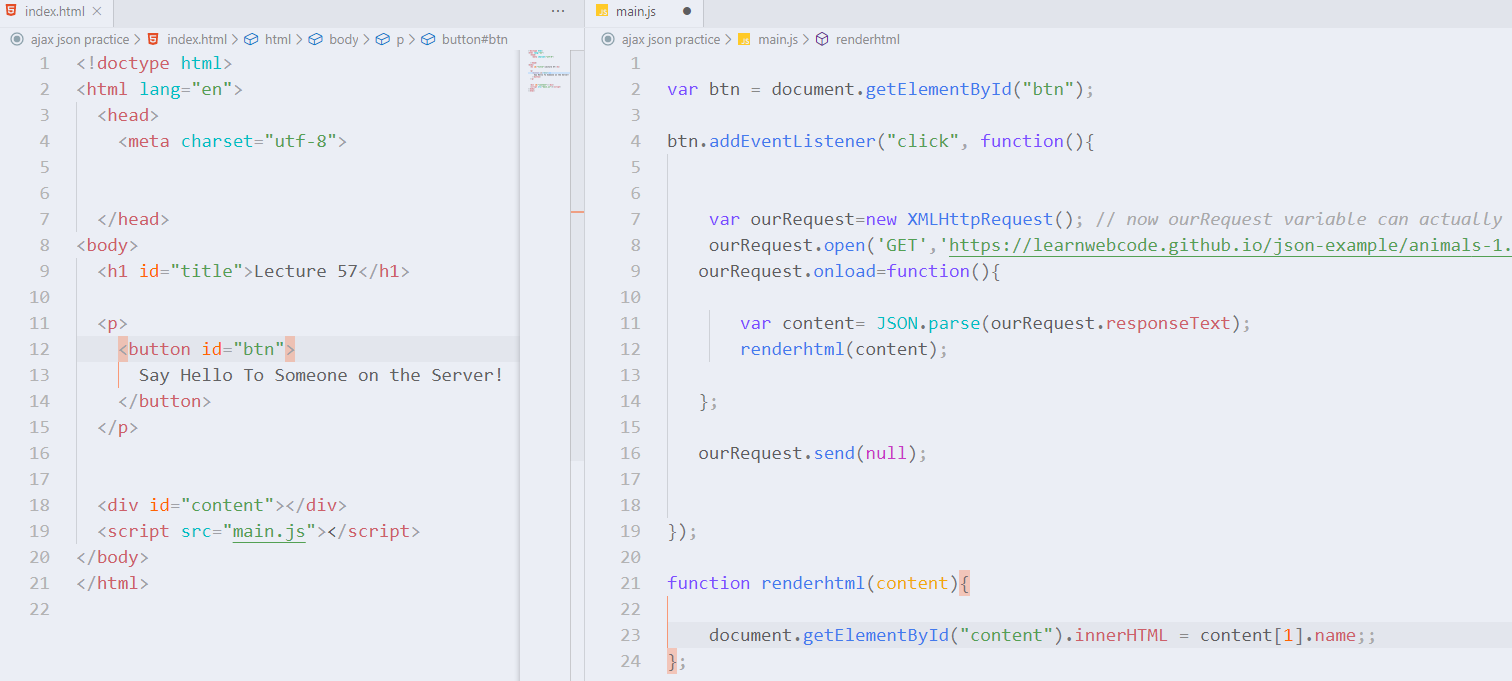
2.To read the data as object we need to read the data as .json file. For that: instead of .respnseText we have to use : JSON.parse([variable].responseText);



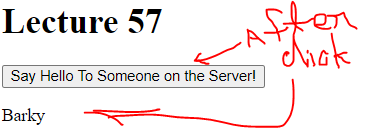
3.Now a full example of use ajax and json: see code in a file named: “ajax json practice”

Json:





Output:



And if we want to hide the button after seeing the output : we add a line bottom of the addEvntListener function:

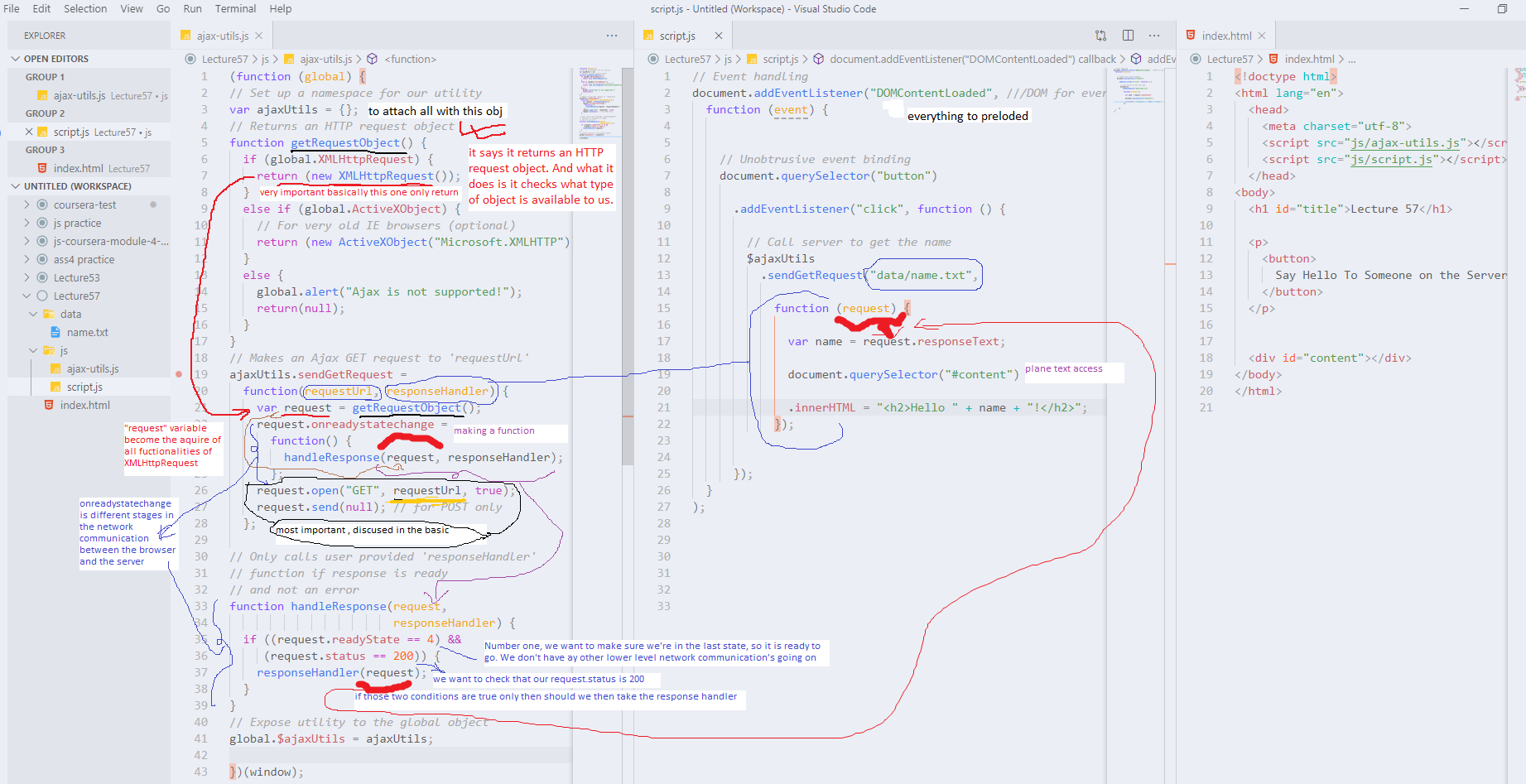
btn**.**style**.**display = "none"; ///css code inside js file

For 2nd example: see video “part 1,2,3,4”

[this is form coursera]

First without json:

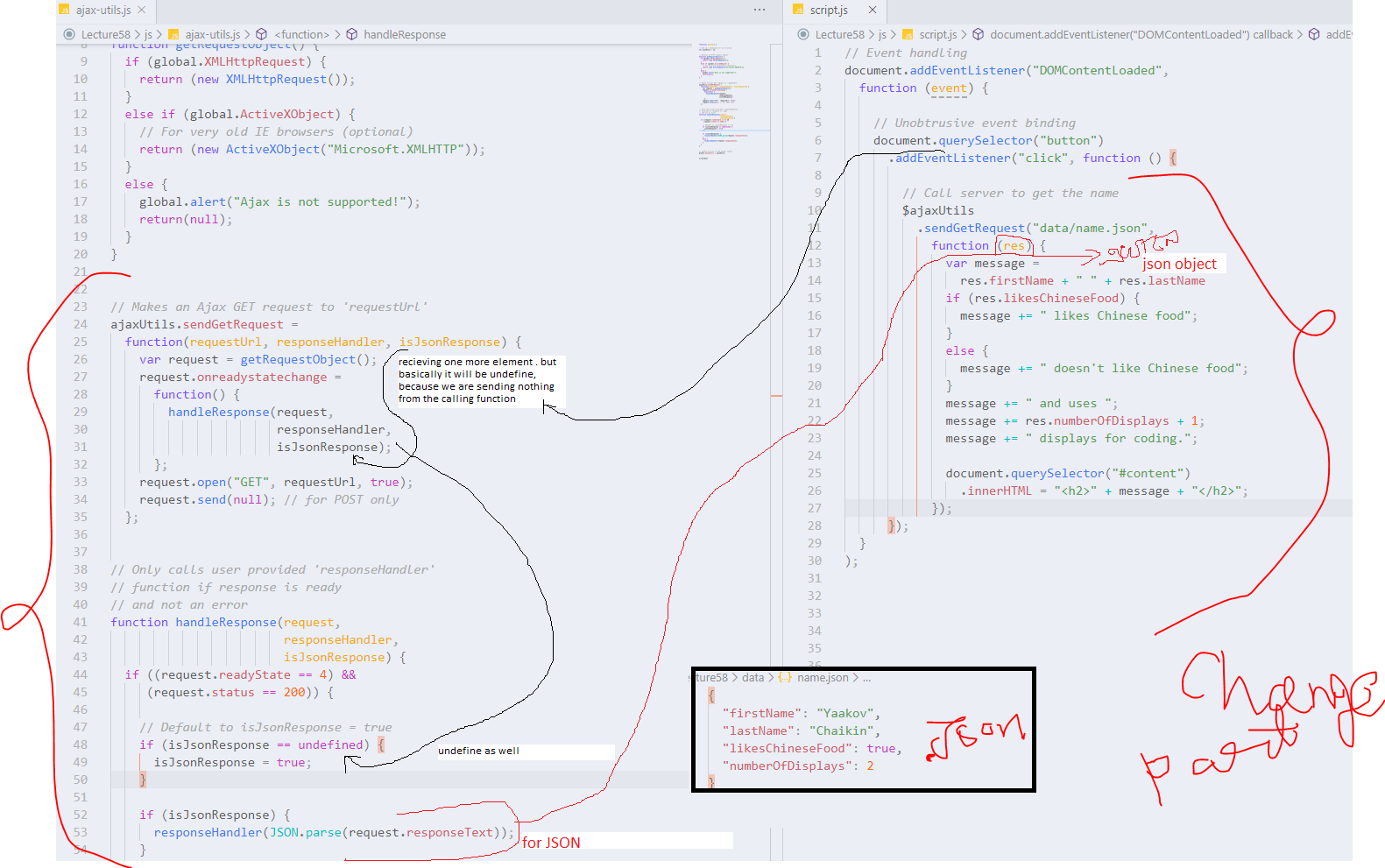
4. see img : ‘advance ajax without json.PNG’



Code:

|  |  |  |
| --- | --- | --- |
| Index.html | Script.js | Ajax.js |
| <!doctype html>  <html lang="en">    <head>      <meta charset="utf-8">      <script src="js/ajax-utils.js"></script>      <script src="js/script.js"></script>    </head>  <body>    <h1 id="title">Lecture 57</h1>      <p>      <button>        Say Hello To Someone on the Server!      </button>    </p>      <div id="content"></div>  </body>  </html> | //*Event handling*  document**.**addEventListener("DOMContentLoaded", //*/DOM for ever thing preLoaded*    function (event) {        //*Unobtrusive event binding*      document**.**querySelector("button")  **.**addEventListener("click", function () {            //*Call server to get the name*          $ajaxUtils  **.**sendGetRequest("data/name.txt",              function (request) {                var name = request**.**responseText;                document**.**querySelector("#content")  **.**innerHTML = "<h2>Hello " + name + "!</h2>";              });          });    }  ); | (function (global) {  //*Set up a namespace for our utility*  var ajaxUtils = {};  //*Returns an HTTP request object*  function getRequestObject() {    if (global**.**XMLHttpRequest) {      return (new XMLHttpRequest());    }    else if (global**.**ActiveXObject) {      //*For very old IE browsers (optional)*      return (new ActiveXObject("Microsoft.XMLHTTP"));    }    else {      global**.**alert("Ajax is not supported!");      return(null);    }  }  //*Makes an Ajax GET request to 'requestUrl'*  ajaxUtils**.**sendGetRequest =    function(requestUrl, responseHandler) {      var request = getRequestObject();      request**.**onreadystatechange =        function() {          handleResponse(request, responseHandler);        };      request**.**open("GET", requestUrl, true);      request**.**send(null); //*for POST only*    };  //*Only calls user provided 'responseHandler'*  //*function if response is ready*  //*and not an error*  function handleResponse(request,                          responseHandler) {    if ((request**.**readyState == 4) &&       (request**.**status == 200)) {      responseHandler(request);    }  }  //*Expose utility to the global object*  global**.**$ajaxUtils = ajaxUtils;  })(window); |

5. with json:



Code: html same as previous:

|  |  |
| --- | --- |
| Script.js | Ajax.js |
| //*Event handling*  document**.**addEventListener("DOMContentLoaded",    function (event) {        //*Unobtrusive event binding*      document**.**querySelector("button")  **.**addEventListener("click", function () {            //*Call server to get the name*          $ajaxUtils  **.**sendGetRequest("data/name.json",              function (res) {                var message =                  res**.**firstName + " " + res**.**lastName                if (res**.**likesChineseFood) {                  message += " likes Chinese food";                }                else {                  message += " doesn't like Chinese food";                }                message += " and uses ";                message += res**.**numberOfDisplays + 1;                message += " displays for coding.";                document**.**querySelector("#content")  **.**innerHTML = "<h2>" + message + "</h2>";              });        });    }  ); | (function (global) {  //*Set up a namespace for our utility*  var ajaxUtils = {};  //*Returns an HTTP request object*  function getRequestObject() {    if (global**.**XMLHttpRequest) {      return (new XMLHttpRequest());    }    else if (global**.**ActiveXObject) {      //*For very old IE browsers (optional)*      return (new ActiveXObject("Microsoft.XMLHTTP"));    }    else {      global**.**alert("Ajax is not supported!");      return(null);    }  }  //*Makes an Ajax GET request to 'requestUrl'*  ajaxUtils**.**sendGetRequest =    function(requestUrl, responseHandler, isJsonResponse) {      var request = getRequestObject();      request**.**onreadystatechange =        function() {          handleResponse(request,                         responseHandler,                         isJsonResponse);        };      request**.**open("GET", requestUrl, true);      request**.**send(null); //*for POST only*    };  //*Only calls user provided 'responseHandler'*  //*function if response is ready*  //*and not an error*  function handleResponse(request,                          responseHandler,                          isJsonResponse) {    if ((request**.**readyState == 4) &&       (request**.**status == 200)) {      //*Default to isJsonResponse = true*      if (isJsonResponse == undefined) {        isJsonResponse = true;      }      if (isJsonResponse) {        responseHandler(JSON**.**parse(request**.**responseText));      }      else {        responseHandler(request**.**responseText);      }    }  }  //*Expose utility to the global object*  global**.**$ajaxUtils = ajaxUtils;  })(window); |