

Tyler Moak

COP3813

Intro to Internet Computing

For the fifth project I utilized all my knowledge in JavaScript, jQuery, HTML, CSS/Bootstrap and PHP to design and implement a currency conversion app.

In this assignment I looked at the provided code and my unit conversion app to get an idea how to approach the problem of converting currency. I also took into account that my script to convert the currencies will be sitting on the server and will need proper way interact between the web server and JavaScript and HTML DOM.

Analyzing the provided currency conversion code, I used the first script which utilized the Yahoo! Finance API and the Yahoo! Query Language (YQL). The currencyConverter function worked perfectly and didn't need any adjustment to its code. Using only that function, I passed my three parameters into it and echoed the result back to the JavaScript XMLHttpRequest object.

For the JavaScript and HTML DOM, I reused the code from project 3 the units converter and modified it to meet my use case. In the DOM I changed that there is no select list to choose other units to convert to as only currency is used. I also populated two currency select lists from Google Currency. Some lesser known currencies may not work, this might have been an oversight as I am using Google's list of currencies and converting them via Yahoo! API.

To have JavaScript interact with the web server PHP script, I used AJAX to send requests to the server. This was the hardest part of the project yet it was incredibly simple to do. On the JavaScript side this code was used:

```

var xmlhttp = new XMLHttpRequest();
//If the state changes to 4 or 200, which are good states unlike 404 or
500, the response will be put into the result
xmlhttp.onreadystatechange = function () {
    if (this.readyState === 4 && this.status === 200) {
        document.getElementById("result_currency").value =
this.responseText + " " + valTo;
    }
};
//This sends a GET request to my php page and passes the currency values
by putting them in the URL
xmlhttp.open("GET", "p5.php?currency_input=" + myCurrency +
"&currency_from=" + valFrom + "&currency_to=" + valTo, true);
xmlhttp.send();

```

XMLHttpRequest handles the GET from PHP by passing three values to a URL and sending the request. Once the state of the request changes to either 4 or 200, which is a good state, the response text is placed in the result div. On the PHP side the code used to handle requests are:

```

$currency_from = $_REQUEST["currency_from"];
$currency_to = $_REQUEST["currency_to"];
$currency_input = $_REQUEST["currency_input"];
echo currencyConverter($currency_from,$currency_to,$currency_input);

```

What this piece of code does is store the request variables from the URL into local variables and calls the currency conversion function. The result of that function is echoed out back to JavaScript. It is a very simple operation and preforms smoothly.

I used brackets for designing my site. I also used two Brackets extensions that let me code faster and more efficient: <https://github.com/dingdong-io/brackets.beautify.io>, <http://www.oslibrary.in/other-information-code/brackets-php-syntax-hint>. I really like the live preview option it has. Also its minimal interface is just right for me. User extensions were also a nice feature. I'm utilizing a github extension for brackets to have source control. I used Firefox for the majority of my testing as I could see how my webpage can be responsive on different screens.

http://www.w3schools.com/php/php_ajax_php.asp