

JavaScript α

Due Sep 24 by 11:59pm **Points** 2 **Submitting** a text entry box

Available Sep 14 at 12am - Sep 24 at 11:59pm 11 days

This assignment was locked Sep 24 at 11:59pm.

Homework: JavaScript α (2 Points)

Important: Accept the [GitHub Classroom](https://classroom.github.com/a/l5NmPoCU) [_\(https://classroom.github.com/a/l5NmPoCU\)_](https://classroom.github.com/a/l5NmPoCU) assignment for the starter code and details. Should you have any issues or questions about setting up your GitHub Classroom repository, feel free to leave your questions on [this Piazza post](https://piazza.com/class/kt08hp32l2b2p2?cid=11) [_\(https://piazza.com/class/kt08hp32l2b2p2?cid=11\)_](https://piazza.com/class/kt08hp32l2b2p2?cid=11).

This homework will introduce you to the syntax and semantics of the JavaScript language. We will be covering multiple topics, including element identification, manipulation, and styling, as well as logging, alerting, branching, looping, and even fetching from an API. By the end of this homework, you will have implemented the functionality supporting Badger Shop, and you will be able to improve the UI in JavaScript β.

Do **not** modify any files except for `script.js`. All questions are capable of being solved with [jQuery](https://api.jquery.com/) [_\(https://api.jquery.com/\)_](https://api.jquery.com/) or standard JavaScript syntax within `script.js`. Modifying any file outside of `script.js` will reduce your score by 1 point.

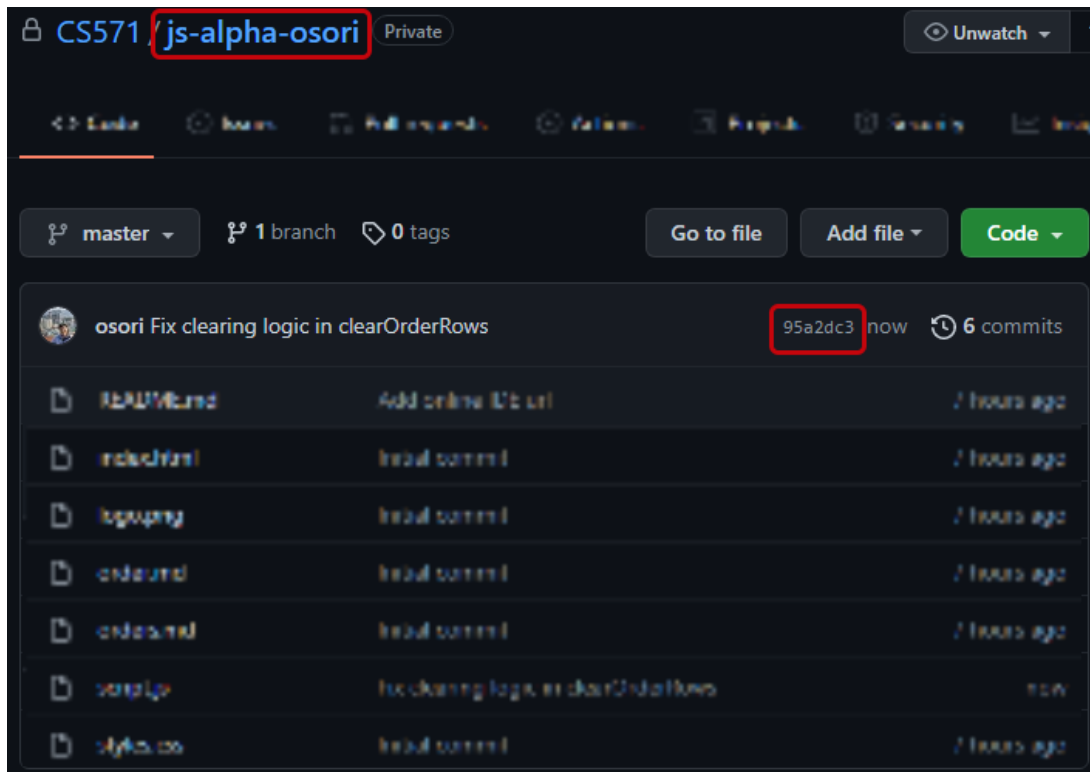
Objective: Complete each function inside of the `script.js` file with its desired functionality as described below. Do NOT modify `index.html`, `styles.css`, or add any additional files.

Submission Details

In order to submit this assignment, you will need to commit and push your code to your private repository created for you in GitHub Classroom (see above). When your code is completed and ready to grade, submit this assignment on Canvas with the name of your repository and your last commit hash, e.g.

js-alpha-osori, 95a2dc3

You can find your latest commit hash by clicking "commits" in your personal repository.



Problems

showMessageForm (0.1 Points)

Complete this function by making the message form and "Send Message" button beneath the welcome message visible.

sendMessage (0.1 Points)

Complete this function by printing the user-inputted text from the message field (revealed by the function above) to the console.

addPizzazz (0.2 Points)

Complete this function by adding some **pizzazz** [.\(https://www.merriam-webster.com/dictionary/pizzazz\)](https://www.merriam-webster.com/dictionary/pizzazz) to description text of the flash sale item; change at least **three** style attributes (e.g. color, font, font-weight, etc.) of the description text.

saveBalance (0.3 Points)

Complete this function by replacing the existing gift card balance with the user-inputted balance. Accept only rational numbers (that is, positive or negative numbers, with or without decimal points). Do not accept balances with a dollar sign, or any other illegal character. If the user enters an improper balance, print "Cannot update balance, syntax error!" to the console. Otherwise, update the balance accordingly.

An empty or whitespace-only string is **not** considered valid.

printBalance (0.1 Points)

Complete this function by printing the saved balance to the console in the format "You have {BALANCE_AMOUNT} in your gift card!"

alertBalance (0.3 Points)

Complete this function by performing the following conditional logic...

- If the saved balance is less than 0, perform a popup alert saying:

We have a special offer for you! Reload your balance now and earn back 10% bonus rewards.

- If the saved balance is between 0 and 100 (inclusive), perform a popup alert saying:

Your current balance is {BALANCE_AMOUNT}. Customers with balance greater than 100 becomes a VIP member and gets a special discount!

- If the saved balance is over 100, perform a popup alert saying:

You are our VIP member! You get a 10% discount on every purchase.

loadFamilyPlanPaymentsData (0.3 Points)

At Badger Shop, we provide Family Plan, a subscription program for your family that gives family members access to premium services! Complete this function by displaying the data from the array of objects provided within the function to the table. You **must** use a form of looping construct over the array of data!

Additionally, if a family member has a amount due of more than **20** dollars, highlight that row in red.

Code has already been added to ensure that the data is only displayed once, see `hasLoadedFamilyPlanPaymentsData`.

addOrderRows (0.4 Points)

Important: You must be signed into the [UW VPN](http://uwmadison.vpn.wisc.edu) [.\(http://uwmadison.vpn.wisc.edu\)](http://uwmadison.vpn.wisc.edu) or [Computer Sciences VPN](http://compsci.vpn.wisc.edu) [.\(http://compsci.vpn.wisc.edu\)](http://compsci.vpn.wisc.edu) in order to connect to the API below. See [this Wisc Help Article](https://kb.wisc.edu/page.php?id=90370) [.\(https://kb.wisc.edu/page.php?id=90370\)](https://kb.wisc.edu/page.php?id=90370) on how to setup GlobalProtect and connect to the VPN. When GlobalProtect asks you to enter portal address, use the following addresses for your preferred VPN service:

You can watch [our video walkthrough](#) on how to connect to the VPN and access our API endpoints (Please disregard the part that you must be connected to the CS VPN).

VPN Service	Portal Address
Computer Sciences Departmental VPN	compsci.vpn.wisc.edu
UW-Madison VPN	uwmadison.vpn.wisc.edu

Complete this function by fetching **four** orders from an API described below and appending them to the orders table. You may either make four calls to

`http://cs571.cs.wisc.edu:53706/api/badgershop/order`, which does not require a parameter, or a single call to `http://cs571.cs.wisc.edu:53706/api/badgershop/orders?amount=<am>`, which requires a parameter specifying the number of orders to return. These APIs are described in more detail in `order.md` and `orders.md` respectively.

Ordering of the rows is not a concern.

clearOrderRows (0.2 Points)

Complete this function by clearing all of the order rows. Make sure that you are still able to add additional rows after the table has been cleared!

Expected Outcomes

Below are visualizations of the expected outcomes for each function above.

Function 1: showMessageForm

Badger Shop

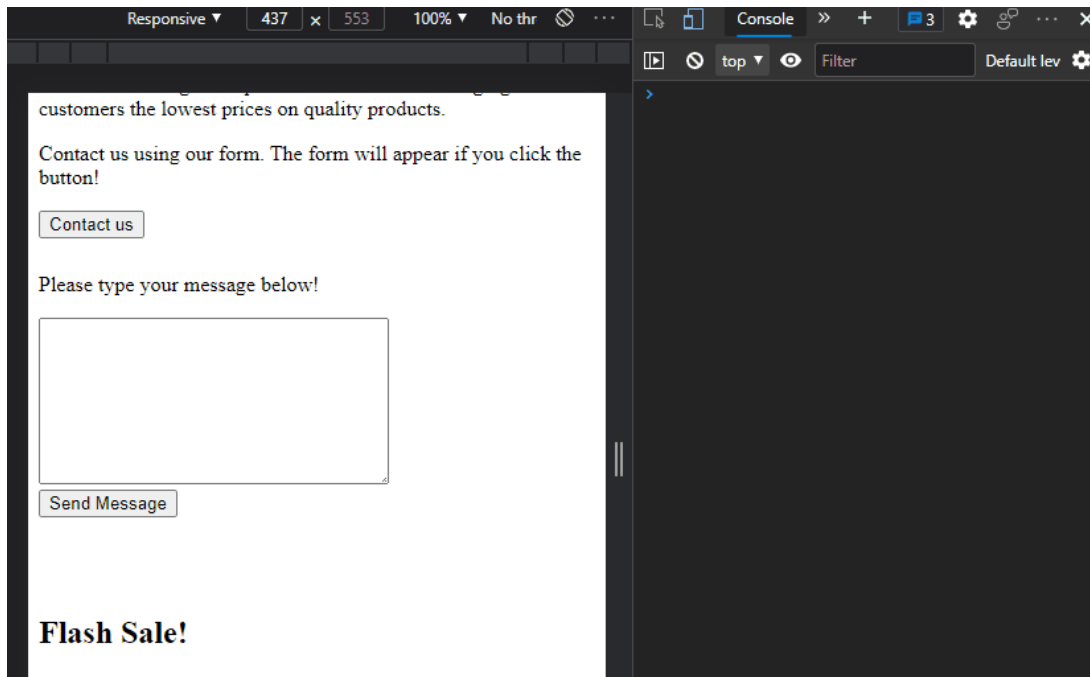
Welcome!

Welcome to Badger Shop! We are dedicated to bringing our customers the lowest prices on quality products.

Contact us using our form. The form will appear if you click the button!

Contact us

Function 2: sendMessage



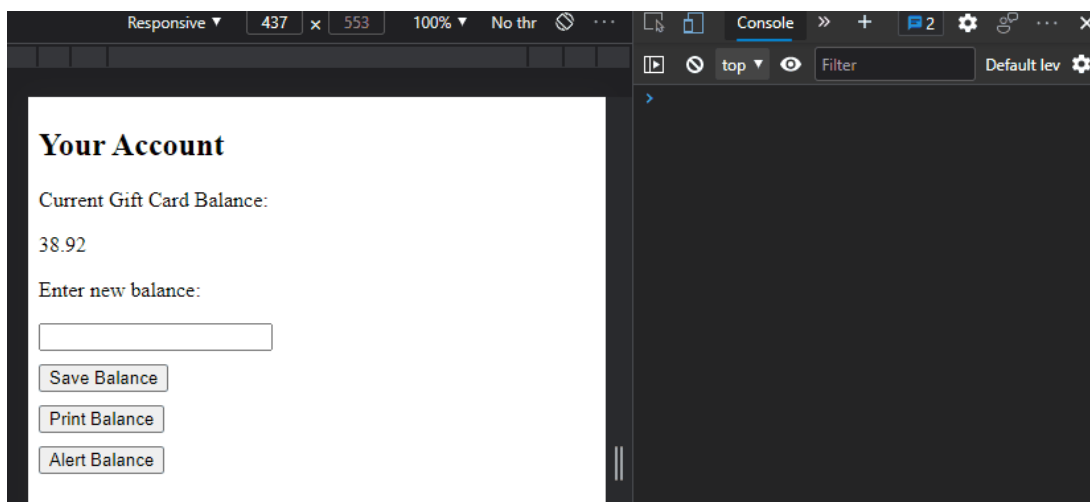
Function 3: addPizzazz

Flash Sale!

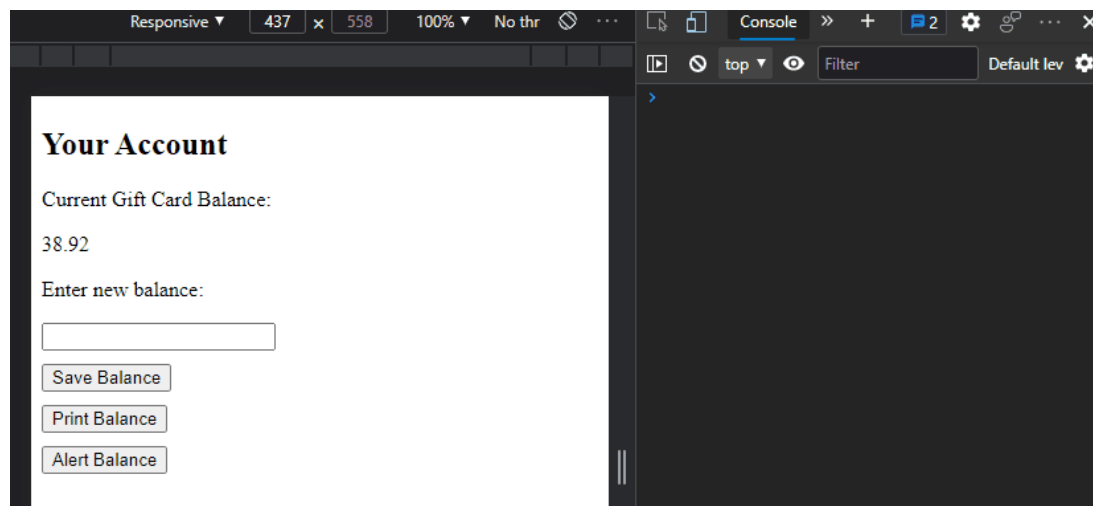
5lbs of brazil nuts for just \$9.99 today — we badgers love 'em!

Add Pizzazz

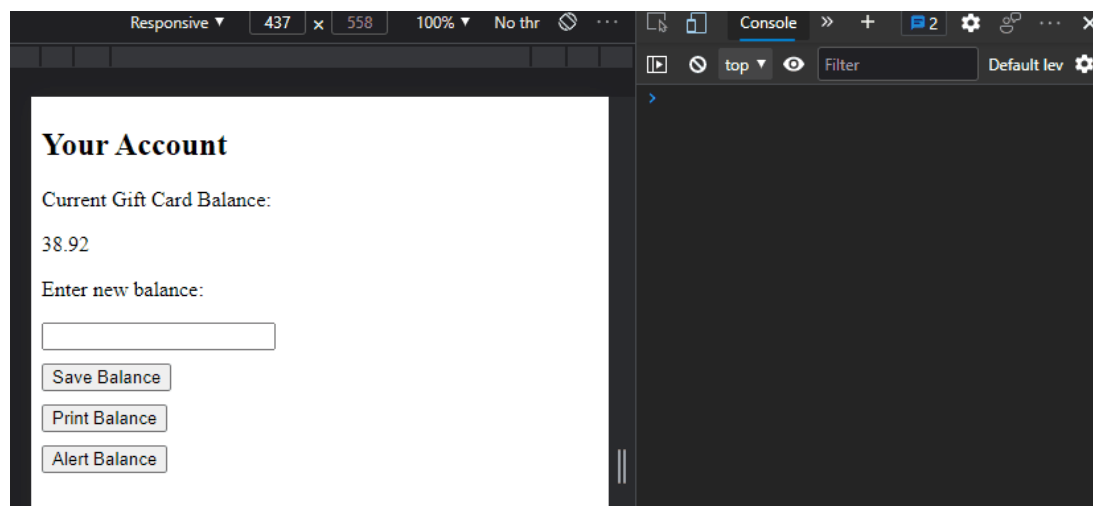
Function 4: saveBalance



Function 5: printBalance



Function 6: alertBalance



Function 7: loadFamilyPlanPaymentsData

Family Plan Payments

You are currently subscribed to our Family Plan. Below is the amount due for each family member.

Name	Amount Due
<div>Load Family Plan Payment Data</div>	

My Orders

Date Purchased	Product Name	Amount Paid
<div>Add Order Rows</div>		
<div>Clear Rows</div>		

Function 8: addOrderRows

Save Balance

Print Balance

Alert Balance

Family Plan Payments

You are currently subscribed to our Family Plan. Below is the amount due for each family member.

Name	Amount Due
<div>Load Family Plan Payment Data</div>	

My Orders

Date Purchased	Product Name	Amount Paid
<div>Add Order Rows</div>		
<div>Clear Rows</div>		

[Back to top](#)

Function 9: clearOrderRows

2020-11-11T05:29:13.162Z	Sleek Soft Fish	152.81
2020-11-07T13:02:13.018Z	Refined Fresh Bike	602.55
2020-11-14T15:25:52.159Z	Rustic Fresh Car	421.13
2020-11-18T21:47:14.302Z	Tasty Fresh Shirt	621.19
2021-03-02T21:12:56.387Z	Practical Rubber Shoes	222.92
2021-01-29T08:00:53.595Z	Generic Cotton Pizza	292.05
2021-03-28T03:52:21.047Z	Tasty Frozen Gloves	321.31
2020-10-12T20:21:25.810Z	Ergonomic Rubber Mouse	374.34
2021-05-10T17:14:02.313Z	Tasty Fresh Pizza	24.89
2021-01-17T23:57:42.129Z	Fantastic Steel Car	415.59
2021-02-06T16:44:35.715Z	Fantastic Rubber Shoes	645.52
2021-07-14T13:53:31.293Z	Incredible Wooden Chips	879.57
2020-09-26T17:26:18.809Z	Unbranded Granite Keyboard	537.13
2020-11-17T22:29:18.051Z	Practical Fresh Pizza	134.38
2021-05-09T11:58:17.607Z	Licensed Granite Gloves	188.79
2020-11-27T03:02:56.878Z	Intelligent Wooden Bacon	237.57

Add Order Rows

Clear Rows

[Back to top](#)

JavaScript α

Criteria	Ratings			Pts
Problem 1: showMessageForm	0.1 pts Full Marks Function is properly implemented.	0 pts No Marks Function is improperly implemented or no work is present.		0.1 pts
Problem 2: sendMessage	0.1 pts Full Marks Function is properly implemented.	0 pts No Marks Function is improperly implemented or no work is present.		0.1 pts
Problem 3: addPizzazz	0.2 pts Full Marks Function is properly implemented.	0.1 pts Not Enough Styling The logic for styling is present but less than three styles are applied.	0 pts No Marks Function is improperly implemented or no work is present.	0.2 pts
Problem 4: saveBalance	0.3 pts Full Marks Function is properly implemented.	0.1 pts Incomplete Saving Logic The logic for saving is present but is incomplete as it allows whitespace or strings besides rational numbers to the balance.	0 pts No Marks Function is improperly implemented or no work is present.	0.3 pts

Criteria	Ratings			Pts
Problem 5: printBalance	0.1 pts Full Marks Function is properly implemented.	0 pts No Marks Function is improperly implemented or no work is present.		0.1 pts
Problem 6: alertBalance	0.3 pts Full Marks Function is properly implemented.	0.1 pts Incomplete Alerting Logic The logic for alerting is present but improperly handles all cases of the balance.	0 pts No Marks Function is improperly implemented or no work is present.	0.3 pts
Problem 7: loadFamilyPlanPaymentsData	0.3 pts Full Marks Function is properly implemented.	0.1 pts Missing Red Text The logic for loading Family Plan payments data is present and correct but does not mark members with greater than 20 dollars in red.	0 pts No Marks Function is improperly implemented or no work is present.	0.3 pts

Criteria	Ratings			Pts
Problem 8: addOrderRows	0.4 pts Full Marks Function is properly implemented.	0.1 pts Incorrect Logic The logic for adding order data is present but does not work as intended.	0 pts No Marks Function is improperly implemented or no work is present.	0.4 pts
Problem 9: clearOrderRows	0.2 pts Full Marks Function is properly implemented.	0.1 pts Incomplete Clear Logic The logic for clearing order data is present but does not allow new rows to be added when the table has been cleared.	0 pts No Marks Function is improperly implemented or no work is present.	0.2 pts
Follows Assignment Instructions	0 pts Full Marks Does not modify files outside of script.js	0 pts Negative Marks (-1 Point) Modifies files outside of script.js		0 pts
Total Points: 2				