Java Script Assignments

Basic Assignment:

1. Mark and John are trying to compare their BMI (Body Mass Index), which is calculated using the formula: BMI = mass / height^2 = mass / (height * height). (mass in kg and height in meter).

Store Mark's and John's mass and height in variables

Calculate both their BMIs

Create a boolean variable containing information about whether Mark has a higher BMI than John.

Print a string to the console containing the variable from step 3. (Something like "Is Mark's BMI higher than John's? true").

2. Create a simple page that lets users type in some temperature value in the Fahrenheit scale and when the user clicks a “Show results” button, to show the temperature in Celsius scale. For example, if the user types in 32, your results should show “0 degree Celsius.”

Functionality

At the least, your program should allow for the following

User Input: One text field where the user will type in the temperature in Fahrenheit

Input Validation: When the “Show results” button is clicked, your program should check to see if the user has left the text field empty and also if what the user typed is a number and not anything else.

If there are problems (as in II. above) then your program should show appropriate error messages

Display results: If there are no errors and user input is valid, then your program should show the results to the user.

3. Write a script which uses a prompt box to get an input. Validate that the input is an Integer between 1 and 30, and then print to the page asterisks (*) to represent the number. Your script must run until you have collected 3 valid inputs as well as generate 3 outputs (one output on one line).
Example:
Input: 4
Input: 6
Input: 12
****
******
****

Control Statement:

4. John and Mike both play basketball in different teams. In the latest 3 games, John's team scored 89, 120 and 103 points, while Mike's team scored 116, 94 and 123 points.

Calculate the average score for each team

Decide which teams wins in average (highest average score), and print the winner to the console. Also include the average score in the output.

Then change the scores to show different winners. Don't forget to take into account there might be a draw (the same average score)

EXTRA: Mary also plays basketball, and her team scored 97, 134 and 105 points. Like before, log the average winner to the console. HINT: you will need the && operator to take the decision. If you can't solve this one, just watch the solution, it's no problem :)

Like before, change the scores to generate different winners, keeping in mind there might be draws.

Arrays Assignment:

5. John and his family went on a holiday and went to 3 different restaurants. The bills were $124, $48 and $268.

To tip the waiter a fair amount, John created a simple tip calculator (as a function). He likes to tip 20% of the bill when the bill is less than $50, 15% when the bill is between $50 and $200, and 10% if the bill is more than $200.

In the end, John would like to have 2 arrays:

1) Containing all three tips (one for each bill)

2) Containing all three final paid amounts (bill + tip).

(NOTE: To calculate 20% of a value, simply multiply it with 20/100 = 0.2)

Object Properties and Methods Assignment

6. Let's remember the first coding challenge where Mark and John compared their BMIs. Let's now implement the same functionality with objects and methods.

1. For each of them, create an object with properties for their full name, mass, and height

2. Then, add a method to each object to calculate the BMI. Save the BMI to the object and also return it from the method.

3. In the end, log to the console who has the highest BMI, together with the full name and the respective BMI. Don't forget they might have the same BMI.

Remember: BMI = mass / height^2 = mass / (height * height). (mass in kg and height in meter).

Loop and iterations Assignment

7. Remember the tip calculator challenge? Let's create a more advanced version using everything we learned!

This time, John and his family went to 5 different restaurants. The bills were $124, $48, $268, $180 and $42.

John likes to tip 20% of the bill when the bill is less than $50, 15% when the bill is between $50 and $200, and 10% if the bill is more than $200.

Implement a tip calculator using objects and loops:

1. Create an object with an array for the bill values

2. Add a method to calculate the tip

3. This method should include a loop to iterate over all the paid bills and do the tip calculations

4. As an output, create 1) a new array containing all tips, and 2) an array containing final paid amounts (bill + tip). HINT: Start with two empty arrays [] as properties and then fill them up in the loop.

EXTRA AFTER FINISHING: Mark's family also went on a holiday, going to 4 different restaurants. The bills were $77, $375, $110, and $45.

Mark likes to tip 20% of the bill when the bill is less than $100, 10% when the bill is between $100 and $300, and 25% if the bill is more than $300 (different than John).

5. Implement the same functionality as before, this time using Mark's tipping rules

6. Create a function (not a method) to calculate the average of a given array of tips. HINT: Loop over the array, and in each iteration store the current sum in a variable (starting from 0). After you have the sum of the array, divide it by the number of elements in it (that's how you calculate the average)

7. Calculate the average tip for each family

8. Log to the console which family paid the highest tips on average

---

Calculation Application

8. In this assignment, we will write a JavaScript web application called MyCalculator. All 3 files (the .js, .html and .css files should be in the MyCalculator folder and should be called MyCalculator.XXX).

The program should do the following:

-There should be one resultBox in the .html page and one runCalculator button. This should be clicked to run the program.

-Present a menu of 7 choices to the user, using a prompt:

1. Add
2. Subtract
3. Multiply
4. Divide
5. Exponent
6. Mean
7. Quit

- If the user selects anything other than 1-7, the program should print an error message and ask the user to select again. Use a for loop to check this.

-If the user selects 1, the program should ask the user to enter the first number and then the second, and then print out the result in the resultBox. Then it should display the prompt menu again.

-Similar behavior for choices 2,3 and 4: for subtract, subtract the second number from the first, for divide, divide the first number by the second.

-We should do isNAN checks for all numbers. Use a for loop to check that each input is a number.

-For choice 5, the program should ask for the base, and then the exponent, and then print out the answer, using the Math.pow() method or equivalent.

-For choice 6, the program should prompt the user to input a series of numbers. The series ends, when the user enters a "***". At this point, our program should print out the **mean** value, and then show the main menu prompt again.

Pizza App

9. In this assignment you will practice basic HTML, Javascript and CSS concepts

You are asked to build a page to order pizza! The page asks the user to enter information regarding the pizza order and then see a summary of what have been entered. The page is static HTML; however, part of the page will be dynamically built as it will be explained below.

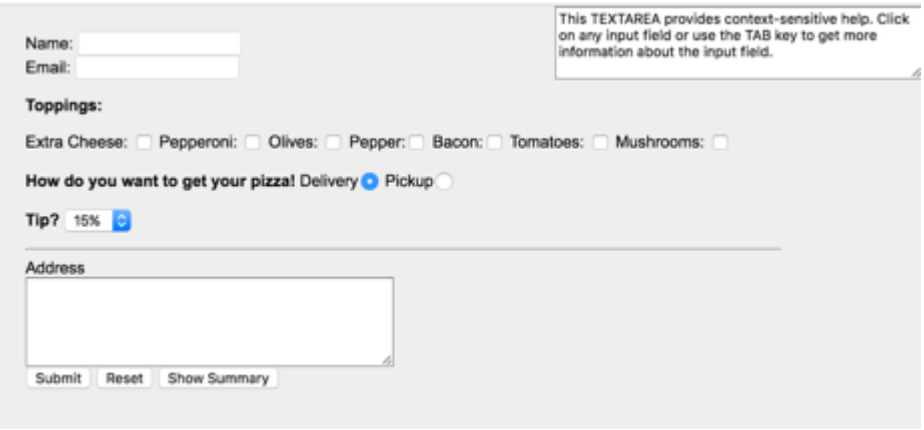

Figure 1

The page consists of the following fields:
1. Customer Name & Email (Input fields)
2. Toppings Options (Check Boxes)
3. Delivery method (Radio Buttons)
4. Tip amount (Drop Down Menu), values are 15%, 20%, and 25%
5. Address (Text Area)
6. Action Buttons
7. Help Text (Text Area)
8. Summary of Order (Table)

All the fields are static fields except the last one (Summary Table) which is built based on the values entered by the user.

CSS is used to apply the following visual properties:

- Input fields width is 600px
- Font type: arial
- Font size: 80%
- Background: light gray (#eee)
- Margin: 20px
- Table even rows: background color light gray (#eee)
- Table odd rows: background color white (#fff)
- Table header row: background color black, font color white

This page should implement the following actions (see Figure 2):

1. The Help Text (Field # 1 in Figure 2) is responsible of showing a Hint message to explaining information about the field that is currently focused (cursor is on the field). If there is no focus on any field, display a default hint message. Messages are displayed on (Table 1).
2. Submit Button should check that all fields are filled otherwise display an error message indicating that there exist missing values.
3. Clear All: Clear All fields.
4. Show Summary: This button is initially disabled. It is enables only when submit button is pressed and all fields values are valid. Clicking this button should build the Summary Table (see #2 in Figure 2). The table will summarize the pizza order based on what have been filled. The table should show:

- Customer Name
- Customer Email

• Customer Address
• Toppings selected
• Delivery option
• Total Price based on this formula: (base price + 1.5 * #of toppings + delivery fee)*1.0+tip. Where: base price= 10 and delivery is 5. For example, if 3 toppings are selected, delivery is NOT checked and 20% tip is selected, total is (10 + 1.5 * 3 + 0) * 1.2 = $17.4

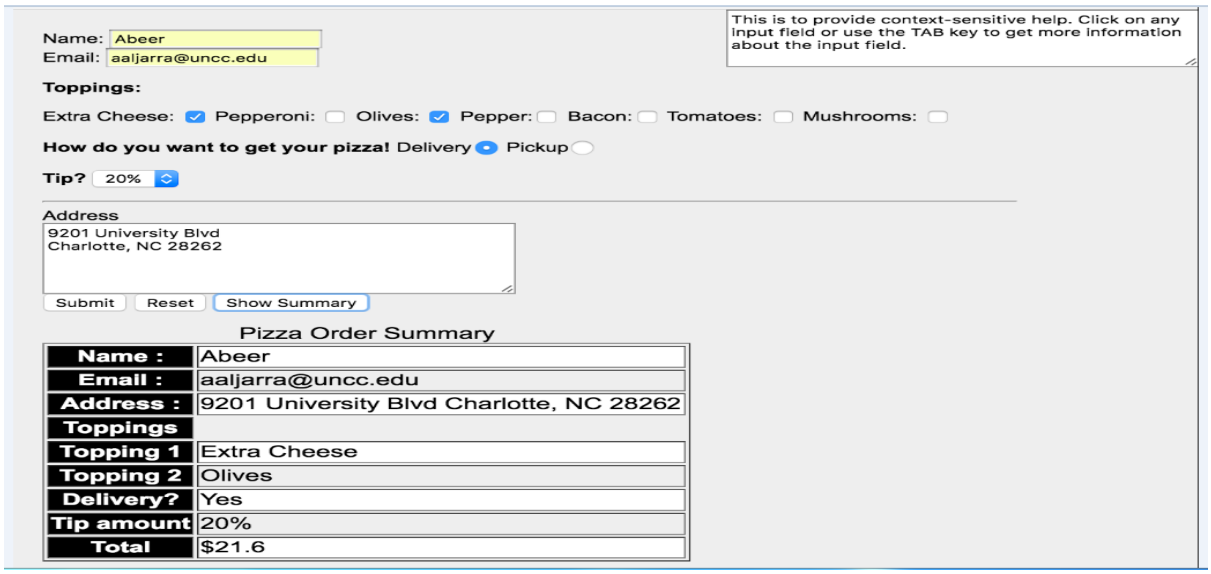

Figure 2

Simple sales app:

10. Your web site sells bags of coffee for the Save the Aardvarks Foundation. You sell regular coffee at $9/bag, you sell decaffeinated coffee at $8/bag, and you sell mocha coffee for $11/bag. If the user buys more than $100 worth, they get a $15 discount.
Use the prompt to ask the user how many bags of regular coffee s/he wants. Then use the prompt to ask about decaf, and then about mocha. Using the input, calculate the total amount the user will pay.
If the total is over $100, subtract $15 from their total. Write out their receipt to the web page, by writing out how many bags of regular they purchased and how much that works out to, then how many bags of decaf and how much that works out to, and then how many bags of mocha they purchased and how much that works out to. Finally write out how much the total will be.
So, for example, after running the script, the web page might look like this:
You purchased 3 bags of regular, totalling 27 dollars.
You purchased 2 bags of decaf, totalling 16 dollars.
You purchased 6 bags of mocha, totalling 66 dollars.
Your total purchase cost is 93 dollars.
Now, using the prompt ask the user if s/he wants to become a member of the Save the Aardvark Foundation and save 20% on their purchase (if you want to be fancy, you can tell them exactly how much they'll save). If the user says "yes", add another $50 to the total for membership. Now take 20% off that. Print out the receipt, as above, only including the membership.

Advanced Java Assignment

Class and Constructor Creation

Book Class

11. Create a script called library.js. In this file create a constructor function for a Book object. The Book object should have the following properties:

Title

Available: Boolean representing whether the book is checked out or not. The initial value should be false.

Publication Date: Use a date object

Checkout Date: Use a date object

Call Number: Make one up

Authors: Should be an array of Author objects

Author Class

Create a constructor function for an object called Author. It should have a property for the first name and last name of the author.

Patron Class

Create a constructor function for an object called Patron. This represents a person who is allowed to check out books from the library.

Give it the following properties:

Firstname

Lastname

Library Card Number (Make one up)

Books Out: Make it an array

fine: Starts a 0.00

B. Methods to add

Book Class

Add a function to the Book prototype called "checkOut". The function will change the available property of the book from true to false and set the checkout date. The

checkout date should be set to the current date minus some random number of days.

Add a function to the Book prototype called "checkIn". The function will change the available property of the book from false to true.

Add a function called isOverdue that checks the current date and the checked out date and if it's greater than 14 days it returns true

Patron Class

Add a function to the Patron prototype called "read" that adds a book to it's books out property.

Add a function to the Patron prototype called "return" that removes a book from it's books out property.

C. Test Program

Create 5 different books from the Book Class and store them in an array called catalog.

Create 5 different patrons from the Patron Class and store them in an array called patrons.

Write a loop that simulates checkouts and checkins for a 3 month period. Every day iterate over the catalog, and every person in the patrons array. If the patron currently has the book checked out then check it in. If it is not checked out then add it to the patrons list of books via the patrons read method. If the book is overdue then add a fine of $5.00 to the patron returning it. At the end of the 3 month period, display each patron, the books they have currently checked out and any fine they may have.

Prototype Assignment

12. Define a prototype Bank that provides the basic behaviour's for checking and savings accounts. All accounts have a number, balance, and support deposits and withdrawals. Define additional prototypes: CheckingAcc which extends Bank, and SavingsAcc which extends Bank. Should use javascript and html.