ASSESSMENT 2

Task: Declare variables and a number of functions that use loops and conditionals. Also, create an array of building materials and a function that works with a material array.

Submission:
Commit and push to submit.

Build Specifications:
Create a file named `script.js`. Complete all of the following tasks in that file. *NOTE:* You may write additional code in `script.js` if you like. It will not count against you. For example, it's probably a good idea to call the functions you make to test them.

1. Declare a variable named weekdays and initialize it to an array of these three strings: Monday, Wednesday, Friday.

2. Declare a variable named rectangle and initialize it to an object with two properties: height: 20 and width: 15.

3. Declare a function named `greetMe`
    - Parameter(s): `name`
    - Functionality: logs the following greeting: "`Hi ____!`", replacing the blank with the name parameter. Don't forget the "!" at the end.

4. Declare a function named `calculateTriangleArea`
    - Parameter(s): `base`, `height`
    - Functionality: calculates and returns the triangle area based on these two parameters. The formula for triangle area is `base * height / 2`.

5. Declare a function named `countDown`
    - Parameter(s): n/a
    - Functionality: uses a loop to log, one by one, the numbers 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, and 0.

6. Declare a function named `countEvens`
    - Parameter(s): `end`
    - Functionality: uses a loop to log, one by one, the even numbers from 2 through `end`, including both 2 and `end` (if end is even).

7. Declare a function named `checkAdverb`
    - Parameter(s): `word`
    - Functionality: Check if `word` ends with "ly". If it does, return "`adverb`". Otherwise if it does not, return "`unknown`".

8. Declare a variable named `materials` that is initialized to an array with the following objects:

type	strength
straw	1
wood	5
brick	10

9. Declare a function named `printMaterials`
    - Parameter(s): `materialArray` (an array like the one in question 8)
    - Functionality: log each item from the `materialArray`, one per line, with a hyphen between the type and strength, like this: "`type` - `strength`".

10. Declare a function named `getStrengthOfMaterial`
    - Parameter(s): `materialArray`, `type`
    - Functionality: find the item in the `materialArray` that has the given `type`. Return the strength of that material. Or if there is no material with that type, return 0.