# ASSIGNMENT 8 V.2

*Version 21/FA*

## MAPPINGS:

The following course objectives and/or outcomes are measured in this assignment:

### COURSE OBJECTIVES

- 8D: Use classes to create objects that have properties and methods.
- 8E: Use inheritance to create a new class based on an existing class.
- 8F: Use the properties and methods of the objects you create.

### COURSE OUTCOMES

- 1. Use JavaScript to create interactive web applications.
- 2. Write clean, consistent code.

## OVERVIEW:

In Assignment 8, you will be extending the existing `Date` class to create a new class named `SuperDate`. ***Do not*** modify the code in **script.js –** limit your work to the files indicated in the assignment writeup. Modifying code outside the files indicated in the assignment writeup may result up to a 60% penalty.

*Continues . . .*

## GRADING:

| Task | Points | Criteria |
|---|---|---|
| **Part 1** | | |
| **1.1 Setup** | | |
| 4a | 6 | Pass/Fail |
| 6 | 5 | Each line of the standard opening comment is worth 1 point. Subtract 1 point per missing item. |
| 7 | 5 | Pass/Fail |
| **1.2 Test the Application** | | |
| There are no points in this section of the assignment. | | |
| **1.3 Define the SuperDate Class** | | |
| 1a | 5 | Instructor's discretion. Deduct for errors or non-implementation only. |
| 1b | 5 | Instructor's discretion. Deduct for errors or non-implementation only. |
| 1bi | 10 | Instructor's discretion. Deduct for errors or non-implementation only. |
| 1c | 16 | Instructor's discretion. Deduct for errors or non-implementation only.<br>- getFullDayName() and getFullMonthName() are worth 3 points, so no more than three points should be deducted for issues with these of the methods.<br>- getShortDayName() and getShortMonthName() are worth 5 points.<br>-- 2 points are for repetitive code: if the student doesn't repeat code, award two points.<br>-- 3 points for each method are general and no more than three points should be deducted for issues with the methods themselves. |
| **1.5 Cleanup** | | |
| There are no points in this section of the assignment. | | |
| **Total** | **52** | |

| Penalties |
|---|
| Deduct 50% from entire assignment for the use of the `var` keyword in variable declarations. |
| Deduct 30% from the entire assignment is the solution does not load and/or if errors appear in the console that are not generated by explicit methods of the console object (in other words, errors that have not been troubleshot and resolved by the student prior to submission). |
| Deduct 60% from the entire assignment if JavaScript is inline/embedded instead of external. |
| Deduct up to 60% from the entire assignment if the student modified code in **script.js** |
| Deduct a maximum of 17.5% for code that does not comply with the course *Style Guide* and/or which is messy/unorganized, uncommented, or missing semicolons. |

*Continues…*

# TASK:

## MODIFY THE APPLICATION

### 1.1 SETUP

1. Review the course *JavaScript Style Guide* before starting this assignment. Part of the assignment will be graded on your adherence to the *Style Guide*.
2. Download **assignment08_starter.zip** from the *Module 8: Assignment* drop box in Canvas. The file is located beneath the heading *Assignment Resources*.
3. Extract **assignment08_starter.zip.** The file contains two files and two subfolders:
    a. a single HTML document named *index.html*.
    b. a single JavaScript file named *script.js*.
    c. a subfolder named *css*, which contains a single file named *style.css*
    d. a subfolder named *libs*, which contains a single file named *lib_convert.js*
4. Open **index.html**
    a. Link **script.js** and **libs/lib_convert.js** to the document **(6 points)**.
5. Open **libs/lib_convert.js**.
6. Add the **Standard Opening Comment** to the top of the script **(5 points**).
7. Add "use strict" beneath the **Standard Opening Comment**. **(5 points)**.

*Continues . . .*

## 1.2 TEST THE APPLICATION

1. Double-click **index.html** to launch it in your browser; examine the application's behavior.
    a. Place your cursor into the text field and input a date in MM/DD/YYYY format. You should see the date is automatically formatted as shown in Figure 1 (if this doesn't happen, check to make sure **script.js** is linked correctly).

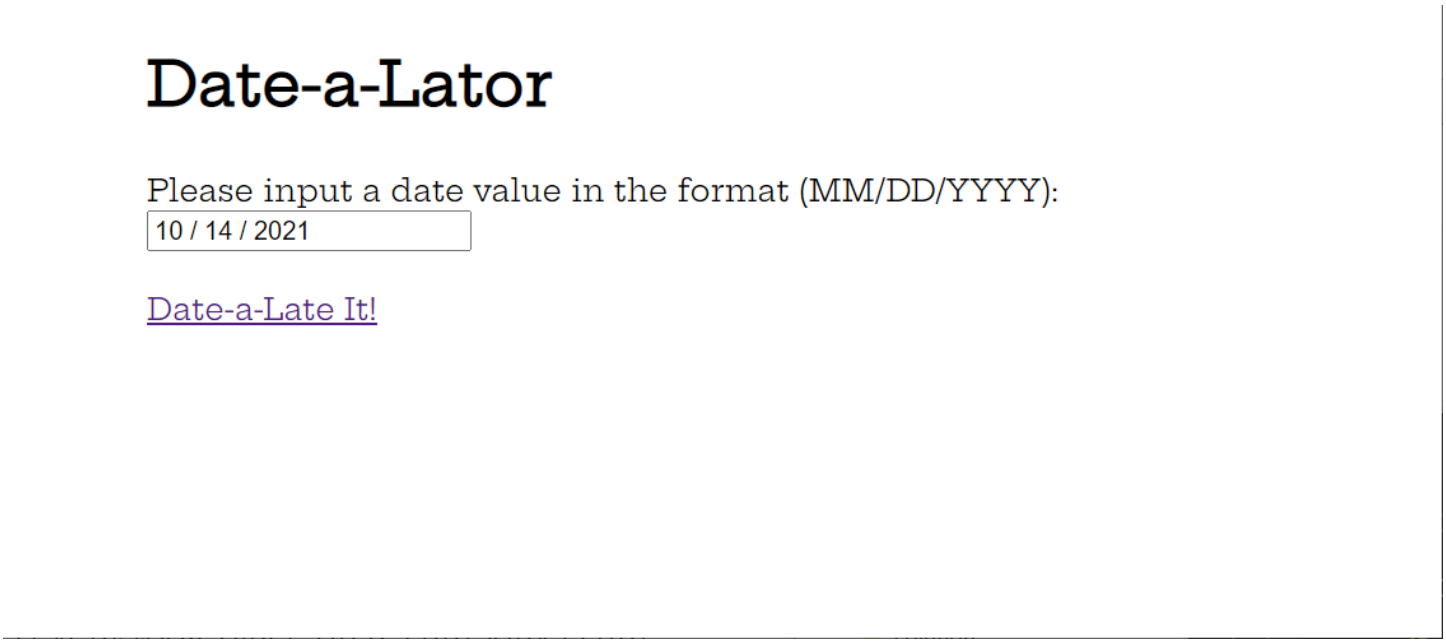



**Figure 1 (your error messages may differ)**

*Continues…*

b. If you click **Date-a-Late It!** errors will appear in the browser Console (Figure 2).


Figure 2

*Continues…*

## 1.3 Define the SuperDate Class

1. Open **lib_convert.js** and accomplish the following:
    a. Define a new class named `SuperDate`, which extends the existing JavaScript `Date` class. (**5 points**).
    b. Define a `constructor` function, which accepts a date unit to convert. **(5 points)**.
        i. Inside the body of the constructor function, you should check if the passed unit is undefined. It it is undefined, you should invoke `super()`, if the passed unit is not undefined, invoke `super()` and pass the passed unit into `super()`. This will create a new `Date` object with either the passed date value, or a new object that represents the current date. **(10 points)**.
    c. Implement the following methods: **(16 points).**
        i. getFullDayName()
            1. This method should accept no parameters.
            2. This method should return the value of `this.getDay()` converted to the full English language value for the day of the week (e.g., Tuesday).
        ii. getShortDayName()
            1. This method should accept no parameters.
            2. This method should return the first three letters of the English language day of the week (e.g., Tue).
        iii. getFullMonthName()
            1. This method should accept no parameters.
            2. This method should return the value of `this.getMonth()` converted to the full English language value for the month (e.g., March).
        iv. getShortMonthName()
            1. This method should accept no parameters.
            2. This method should return the first three letters of the English language month name (e.g., Mar).

**HINT:**

There are two additional points available for getShortDayName() and getShortMonthName() for writing DRY code (don't repeat yourself).

### 1.5 CLEANUP

1. Test your application. If necessary, use the browser's developer tools to troubleshoot any errors you may have.
    a. Sample output values may be found in Appendix A (page 8).
2. Comment and clean up your code:
    a. Make sure to document what functions and blocks of code do.
    b. Ensure consistent alignment, spacing, and carriage returns.
    c. Make sure to remove non-used code; comments should describe your code only.

## SUBMISSION

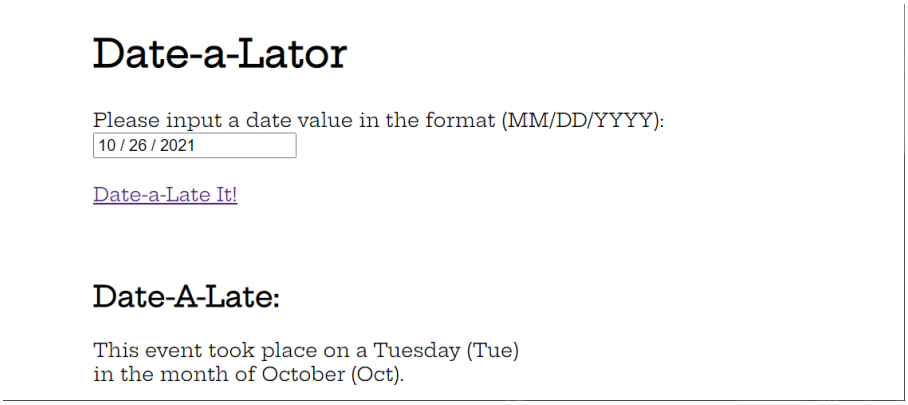
When complete, create a single ZIP file containing your solution for this assignment. The ZIP file should contain all files included in the original starter code or added as part of this assignment.

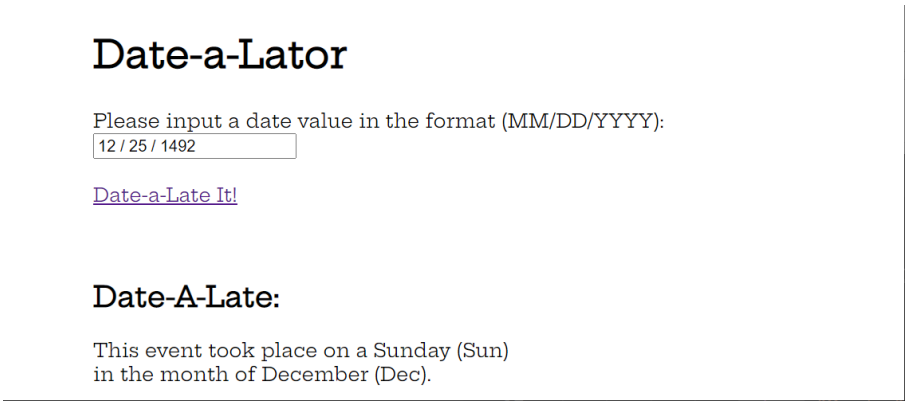
Attach and upload the ZIP file to Assignment 8 and submit.

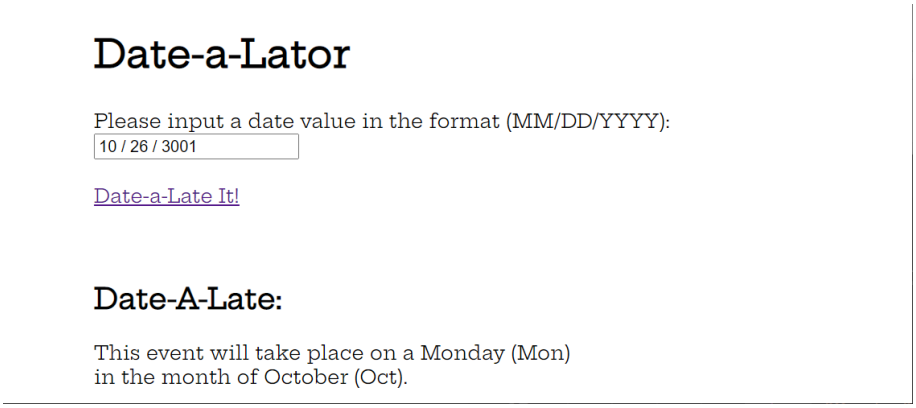
**NOTE: Canvas is configured to only accept ZIP files, DOC files, and DOCX files; it will not accept ZIPx, 7ZIP, pZip, RAR, etc.**

*End Assignment*.

## APPENDIX A


Date-a-Lator
Please input a date value in the format (MM/DD/YYYY):
10 / 26 / 2021
Date-a-Late It!
Date-A-Late:
This event took place on a Tuesday (Tue)
in the month of October (Oct).


Date-a-Lator
Please input a date value in the format (MM/DD/YYYY):
12 / 25 / 1492
Date-a-Late It!
Date-A-Late:
This event took place on a Sunday (Sun)
in the month of December (Dec).


Date-a-Lator
Please input a date value in the format (MM/DD/YYYY):
10 / 26 / 3001
Date-a-Late It!
Date-A-Late:
This event will take place on a Monday (Mon)
in the month of October (Oct).