IMY 220
Assignment 3:
JavaScript & Git

Due: Wednesday 24 August at 13:00.

The submission instructions are available on ClickUP. Any deviation from these instructions will cause a 10% deduction from your mark.

Instructions

- Create a class definition for an object in JavaScript called *EventHandler*, which contains some functionality for managing and displaying an array of events and associated information.
- Download *index.html* and *script.js* from ClickUP. These contain the basic HTML for a valid HTML5 document and a populated JS array. You must include *script.js* inside *index.html*.

You must also create a new file called *EventHandler.js* which will contain all the code for the *EventHandler* object. You must also include this file inside *index.html*.

You are also not allowed to write **any** loops for this assignment or use the *.foreach()* function. All of the functionality must be implemented with the appropriate JS Array functions as discussed in the slides and resources linked from the slides.

Part 1 – EventHandler object

Create a constructor for an object in JavaScript (check the notes to see how this is done) called *EventHandler* which takes one argument: an array of objects. You may assume this array will always be an array of objects like the *events* variable in *script.js*. This array must be saved as a member variable of the object.

This object must have four member functions, each of which is described below:

1. *getEventsBetweenDates(start, end)* – this function must receive two arguments (which must be passed as strings) and must return an array of all the objects in the member variable array that fall between the two dates specified. *Start* should be the starting date of the range (dateStart), and *end* should be the end of the range (dateEnd). Both start and end should be inclusive dates (meaning that the day specified should be included in the range).

So for example, the output to *getEventsBetweenDates("2022/02/01", "2022/02/16")* would be:

```
[{name: 'University expo', description: 'Expo to showcase University
degrees', dateStart: '2022/02/01', dateEnd: '2022/02/14'},{name: 'Hiking
trip', description: 'Hiking trip with a bunch of University friends',
dateStart: '2022/02/14', dateEnd: '2022/02/16'}]
```

2. *getByMonth(month)* – this function receives 1 argument and returns a member array with all the objects that have a starting date with the *month* specified.

So, for example, the output to *getByMonth(05)* would be:

```
[{name: 'Music festival', description: 'Weekend long music
festival with a ton of artists performing', dateStart:
'2022/05/13', dateEnd: '2022/05/15'}]
```

3. *getUniqueDateAndSort()* – this function receives no arguments and returns an array that is sorted by month of the starting date (*HINT*: use array.sort()) and returns all events with the same start and end date only appearing **once** (i.e., dates that fall in the same date range) meaning that the returned array will have no objects with repeating start and end dates. For example, given the following array of objects…

```
[{name: 'University expo', description: 'Expo to showcase University
degrees', dateStart: '2022/02/01', dateEnd: '2022/02/14'},{name: 'Music
festival', description: 'Weekend long music festival with a ton of artists
performing', dateStart: '2022/05/13', dateEnd: '2022/05/15'},{name: 'Market',
description: "Farmer's market day long event", dateStart: '2022/06/12',
dateEnd: '2022/06/12'},{name: 'Science Expo', description: 'Science expo with
sciency stuff', dateStart: '2022/06/12', dateEnd: '2022/06/21'},{name:
'Hiking trip', description: 'Hiking trip with a bunch of University friends',
dateStart: '2022/02/14', dateEnd: '2022/02/16'},{name: 'Park Picnic',
description: 'Picnic event in the park', dateStart: '2022/06/12', dateEnd:
'2022/06/12'}]
```

(so in this case the events "Market" and "Park Picnic" fall in the same date range)

…the function should return the following array:

```
[{name: 'University expo', description: 'Expo to showcase University
degrees', dateStart: '2022/02/01', dateEnd: '2022/02/14'},{name: 'Hiking
trip', description: 'Hiking trip with a bunch of University friends',
dateStart: '2022/02/14', dateEnd: '2022/02/16'},{name: 'Music festival',
description: 'Weekend long music festival with a ton of artists performing',
dateStart: '2022/05/13', dateEnd: '2022/05/15'},{name: 'Market', description:
"Farmer's market day long event", dateStart: '2022/06/12', dateEnd:
'2022/06/12'}]
```

HINT: you can return an array from *reduce()* if you supply an empty array as its initial value.

HINT: use array.sort() - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort

4. *getSummary()* – this function takes one/more optional argument(s), which can either be a single array of objects or any number of objects. The function must return an array of strings where each string is a summary of an object.
If an event is one day only (the start and end dates are the same) then the format must look like the following:

*On **[dateStart]: [name]** (**[description]**)*

if the event takes place across multiple days, the format must look as follows:

*From **[dateStart]** to **[dateEnd]: [name]** (**[description]**)*

For example, the following object…

```
{name: 'Market', description: "Farmer's market day long event", dateStart: '2022/06/12', dateEnd: '2022/06/12'}
```

…must be summarised as follows: ***"On 2022/06/12: Market (Farmer's market day long event)"***

And…

```
{name: 'University expo', description: 'Expo to showcase University degrees', dateStart: '2022/02/01', dateEnd: '2022/02/14'}
```

…must be summarised as follows: ***"From 2022/02/01 to 2022/02/14: University expo (Expo to showcase University degrees)".***

If no argument is given to the function, it must use the member variable array as the list to be summarised. If an array of objects is given, it must use that as the list.
HINT: you can check if a variable is an array with the following code: *variable.constructor === Array* (where *variable* is the name of your variable)
It can also be given any number of objects as individual arguments, in which case it must summarise each, for example, if the function is called as follows:

```
getSummary({name: 'Pizza party', description: "Pizza party at work", dateStart: '2022/07/10', dateEnd: '2022/07/10'})
```

…it must summarise each object in turn:

```
['On 2022/07/10: Pizza party (Pizza party at work)']
```

Part 2 – Extend Array functionality

After defining your *EventHandler* object, you should be able to instantiate an instance of that object and use its member functions, for example, if *handler* is instantiated as an instance of the object with the given array variable, *handler.getByMonth (06)* should return an array of all the objects that fall in the month of June. However, since the functions return arrays, the functions can't be chained, for example, the following will not work: *handler.getByMonth (06).getSummary()*, since *getByMonth()* returns an array and *getSummary()* is not defined as an Array function.

So, to allow for this chaining of functions, extend the Array class using the prototype property and add all of *EventHandler*'s member functions.

If done correctly, the output to *handler.getByMonth (06).getSummary()* would be:

```
["On 2022/06/12: Market (Farmer's market day long event)", 'From 2022/06/12
to 2022/06/21: Science Expo (Science expo with sciency stuff)', 'On
2022/06/12: Park Picnic (Picnic event in the park)']
```

Note about testing: this assignment does not have a DOM-aspect, thus you are expected to do your own testing using *console.log*. However, make sure to take out all *console.log* statements before submitting as only your class definition and Array prototype extensions will be marked.

Part 3: Git

Create a Github account (if you don't have one already) and create a (public) repository called IMY220-Assignment3. Commit the required files to this repository and save the (public) link to this repository in a text file called github.txt, for example: https://github.com/username/IMY220-Assignment2. The result of this should be that whoever is marking your work should be able to navigate to that link, click on the Clone or Download button, and click on Download ZIP to download your file for this assignment, which is the file that will be used for marking purposes. (However, you will still have to submit index.html, script.js, EventHandler.js in a zip file on ClickUP.)

Additional Information

- Refer to the slides and online resources for help

Submit only the following file(s) according to the submission instructions.

- *index.html*
- *script.js*
- *EventHandler.js*