- # Introduction

Objects and arrays are useful in Javascript and any other programming languages we use nowadays. In this report, I will illustrate how I utilised the skills and knowledge I had learned from the lectures to solve all the questions given in the practical. My Firefox version is 93.0.

- # Part1.1 Question Description

In question one, we were asked to build a database of books, containing values such as ISBN, title, author, format, price, and stock. After building the database, we had to implement the required functions into the database to display to the users.

- # Part1.2 Solution

In this question, I started with building a database of books using objects and assign the properties according to the database; I then put the objects of books into an array. I also received an advice from a friend who knows coding and he suggested me to use console.table instead of console.log. Although it is not recommended by the lecturer, but with the time I was given to satisfy the requirement, I found this way made the results the easiest to read.

- # Part1.2.1 printAllBooks() Solution

I first created a loop using the condition 'i < array.length' so it would run through every element in the database. Then I assigned a variable called 'isbnHypens' with the value of the ISBN from each book; I then used subscript 'split' to split every single number of ISBN into an array. After that, I used 'splice' to insert hyphens into the right spots. Lastly, I simply used 'join' to reunite the array into a string of ISBN with hyphens again, and had it log into the console. The result is as follows:

```
» printAllBooks();
console.table()                                                                    Question 1.js:33:20
```

(index)	isbn	title	author	format	price	stock
0	978-0-744-01669-7	The Legend of Zelda: Tri Force Heroes	Prima Games	Hardcover	14.99	2
1	978-0-099-54948-2	To Kill a Mockingbird	Harper Lee	Paperback	4.99	3
2	978-0-744-01668-0	The Legend of Zelda: Tri Force Heroes	Prima Games	Paperback	9.99	1
3	978-0-062-40990-4	Go Set A Watchman	Harper Lee	Audio Book CD	10.89	2
4	978-0-099-52912-5	Catch-22	Joseph Heller	Paperback	6.29	0
5	978-1-785-15028-9	Go Set A Watchman	Harper Lee	Hardcover	9.89	3
6	978-0-554-57990-1	A Clash of Kings	George R. R. Martin	Paperback	4.95	5
7	978-1-853-26000-1	Pride and Prejudice	Jane Austin	Paperback	1.99	4
8	978-1-784-87189-4	Casino Royale	Ian Fleming	Hardcover	6.79	1

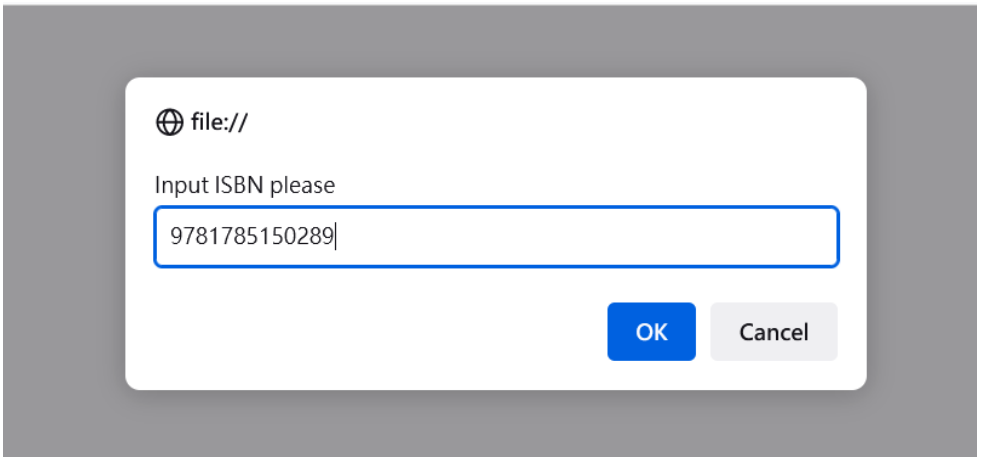
- ## Part1.2.2 sell(isbn) Solution

I created a function that asks the users to prompt an ISBN first and used Regex to ignore unnecessary punctuations so the system would be robust when the users accidentally input unnecessary punctuations into the text. Next, I created a for loop with conditional statements that breaks when the input ISBN matches the one in the database. In case no books were matched, I created a while loop as shown in the following screenshot:

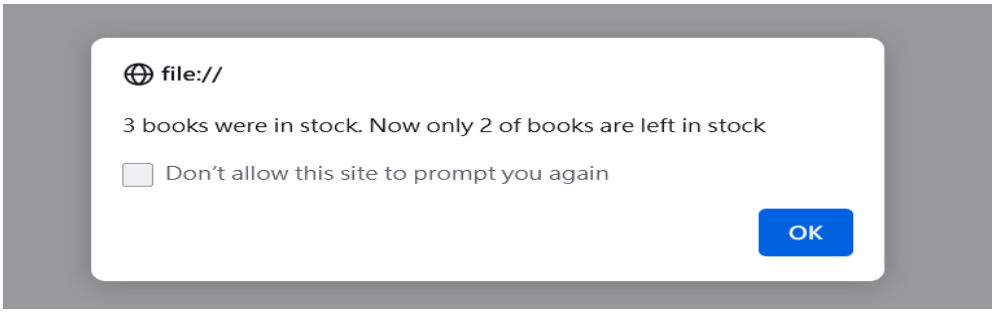
```
while (i == 9) {
    alert('No books found!');
    isbn = prompt('Input a valid ISBN').replace(/[a-z\WA-Z]+/g, '');
    for (i=0; i<list.length; i++) {
        if (isbn == list[i].isbn) {
            list[i].stock--;
            break;
        }
    }
}
```

The loop deals with any input that does not match with any books in the database, it alerts the users and then requests them to prompt a new ISBN again until it matches. The result alerts the users how many books are left in stock and pops a different alert when there are no books left in stock. On the website you will see:

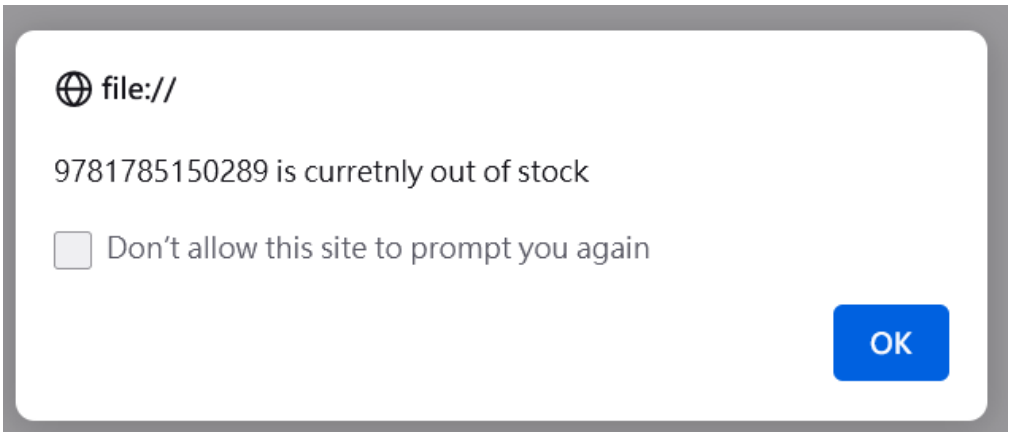
First, prompt ISBN:



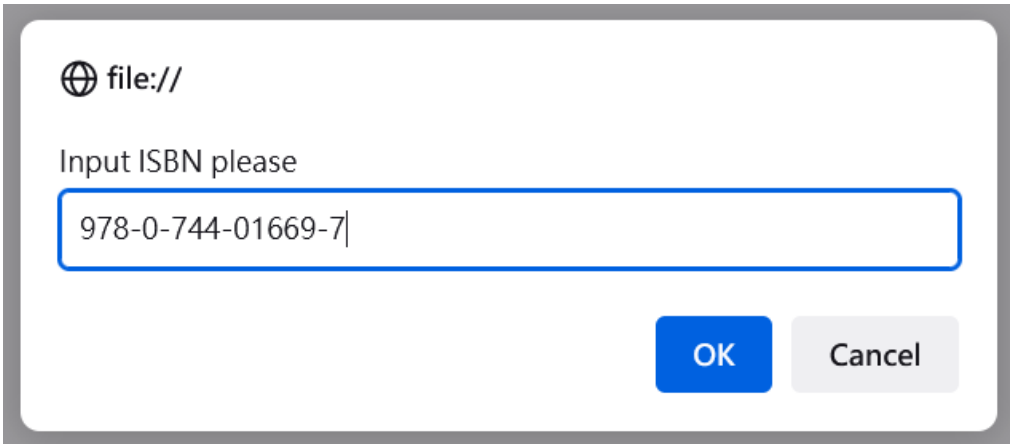
It tells the users how much books are left in stock:

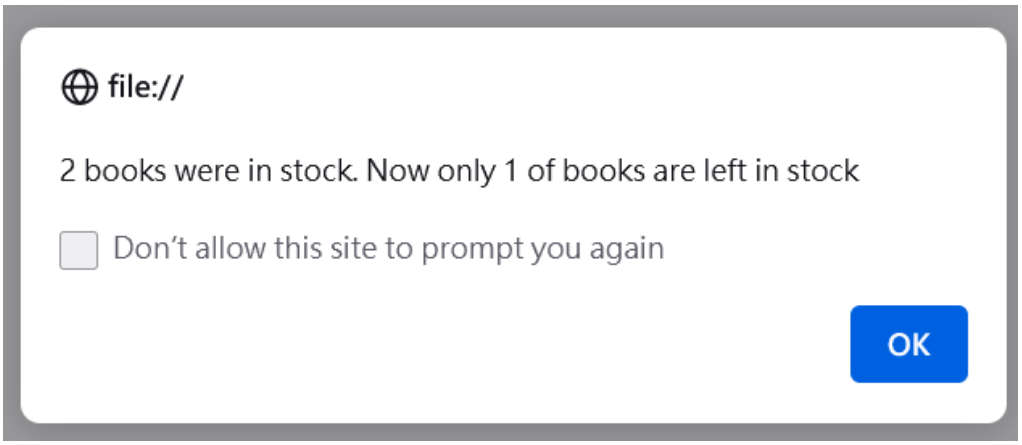


When there are no stocks left, the alert becomes:



Let's run again with hyphens:



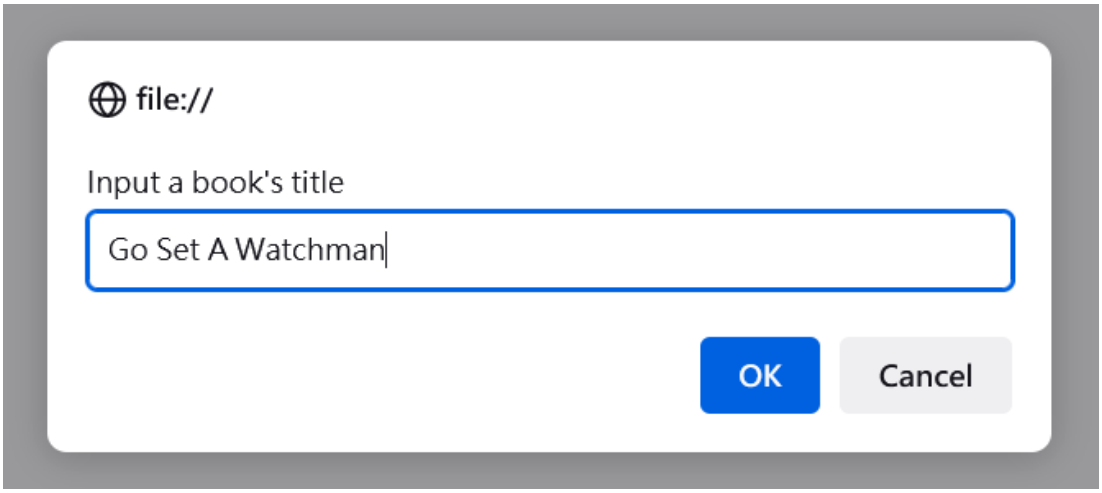


- ## Part1.2.3 printByTitle(title) Solution

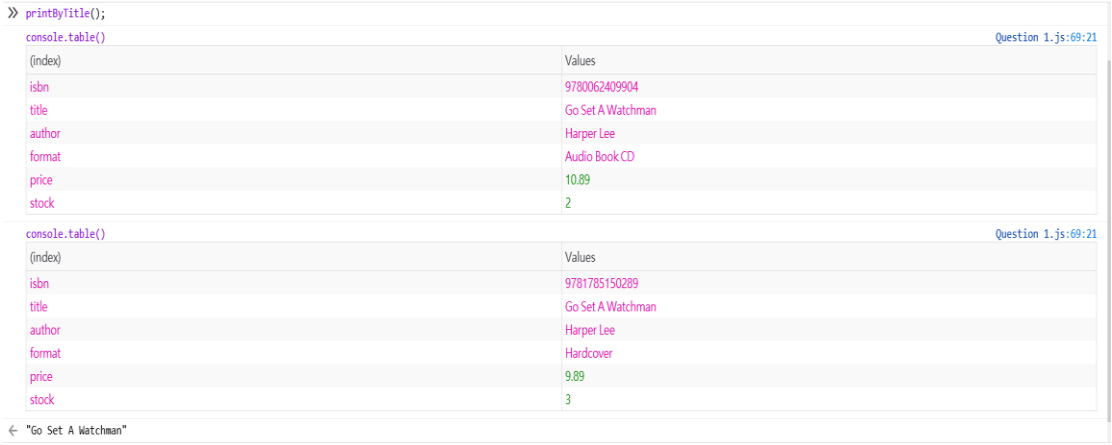
First, the function asks the users to input a book title which is similar as the previous function, then it transfers all the letters into lowercases to ignore uppercases and lowercases as required in the specification. Then it uses a loop to go through the entire database to find the results that match; Since there are books with the same title but different format in the database, we do not break

out of the loop this time until all books are searched. If no titles are matched, the function alerts the users of invalid input. The test of this function is as follows:

First, input a book's title:



It then logs the table in the console:



Check if it is still robust with incorrect cases:

```
» printByTitle();
```

console.table()

(index)	Values
isbn	9780062409904
title	Go Set A Watchman
author	Harper Lee
format	Audio Book CD
price	10.89
stock	2

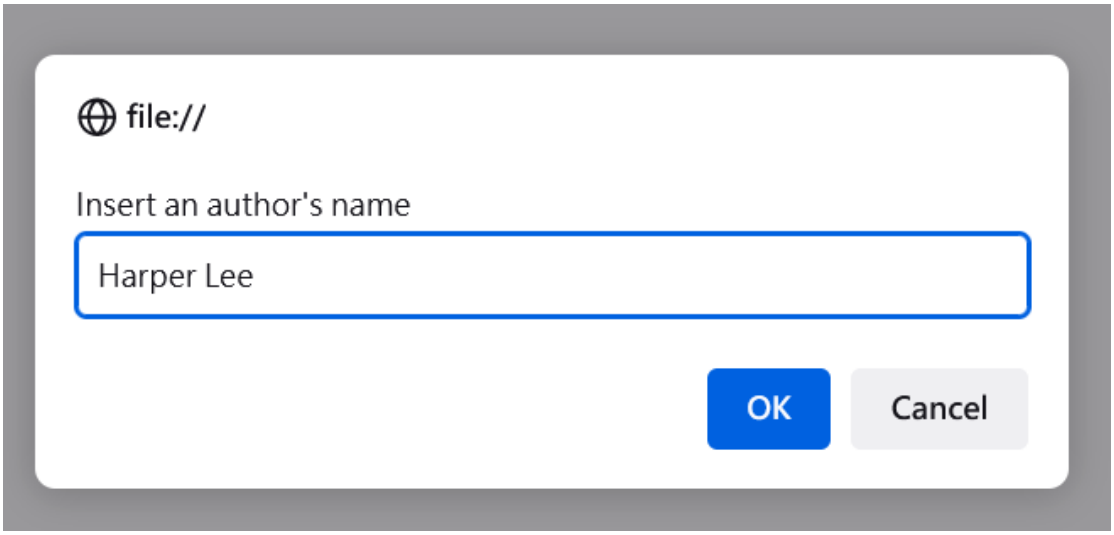
console.table()

(index)	Values
isbn	9781785150289
title	Go Set A Watchman
author	Harper Lee
format	Hardcover
price	9.89
stock	3

```
← "Go SET a WATCHmAn"
```

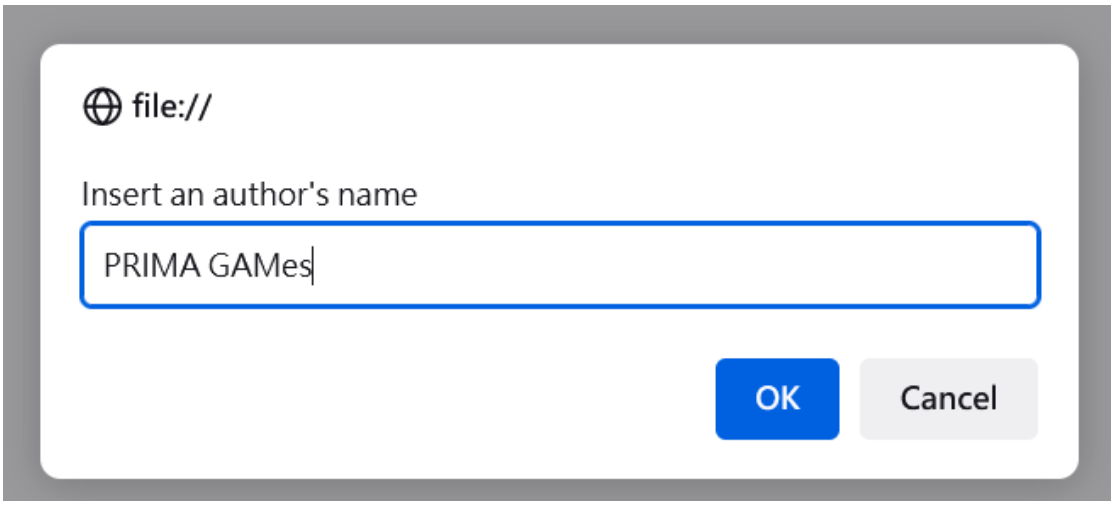
- ## Part1.2.4 printByAuthor(author) Solution

The function operates similarly as the previous function, the only difference is that this function aims to match the property of author rather than the title. The following is the test result of this function:




```
» printByAuthor();
    Harper Lee is the author of 'To Kill a Mockingbird'.
    Harper Lee is the author of 'Go Set A Watchman'.
    Harper Lee is the author of 'Go Set A Watchman'.
← "Harper Lee"
```

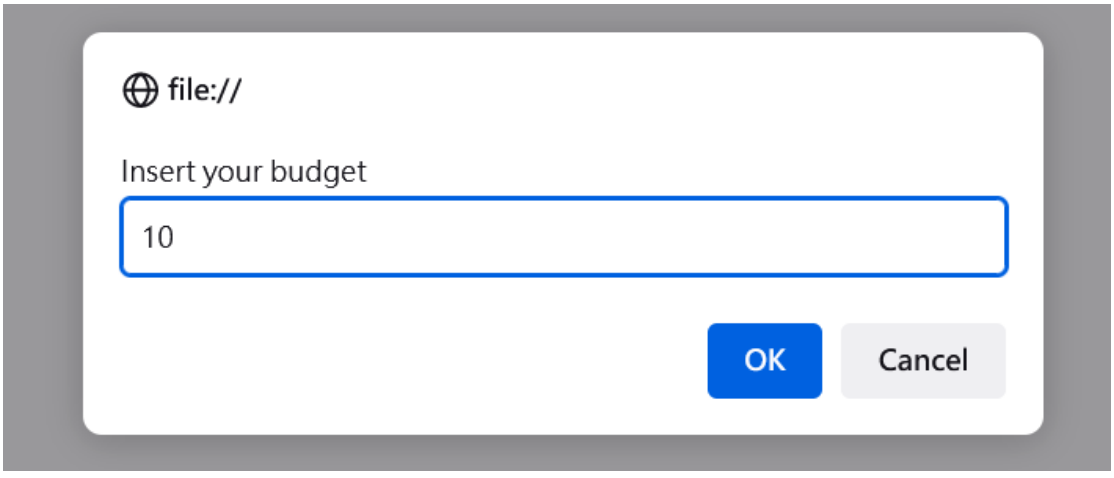
Again, with the incorrect cases:




```
>> printByAuthor();
   Prima Games is the author of 'The Legend of Zelda: Tri Force Heroes'.
   Prima Games is the author of 'The Legend of Zelda: Tri Force Heroes'.
<- "PRIMA GAMes"
```

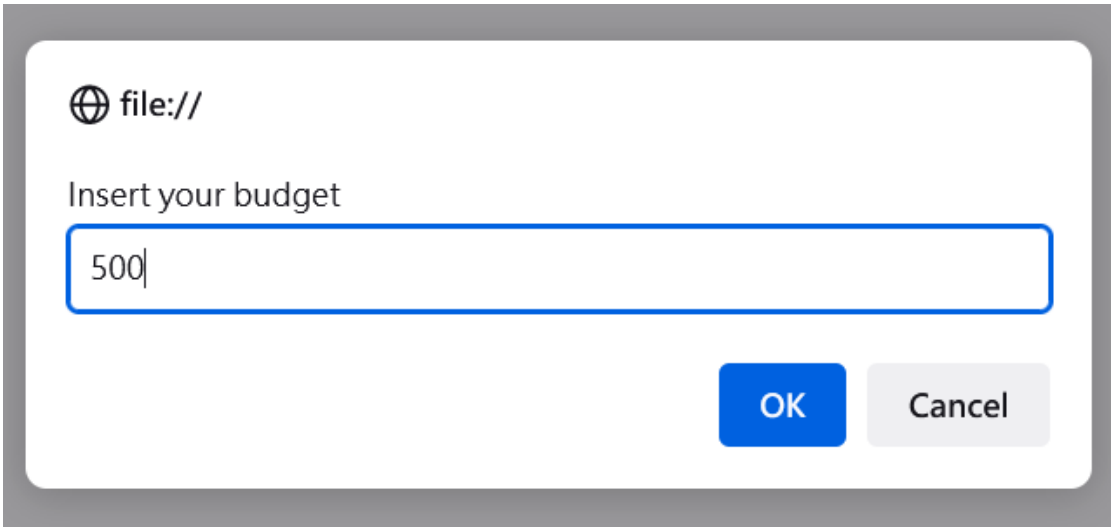
- ## Part1.2.5 printOverPrice(price) Solution

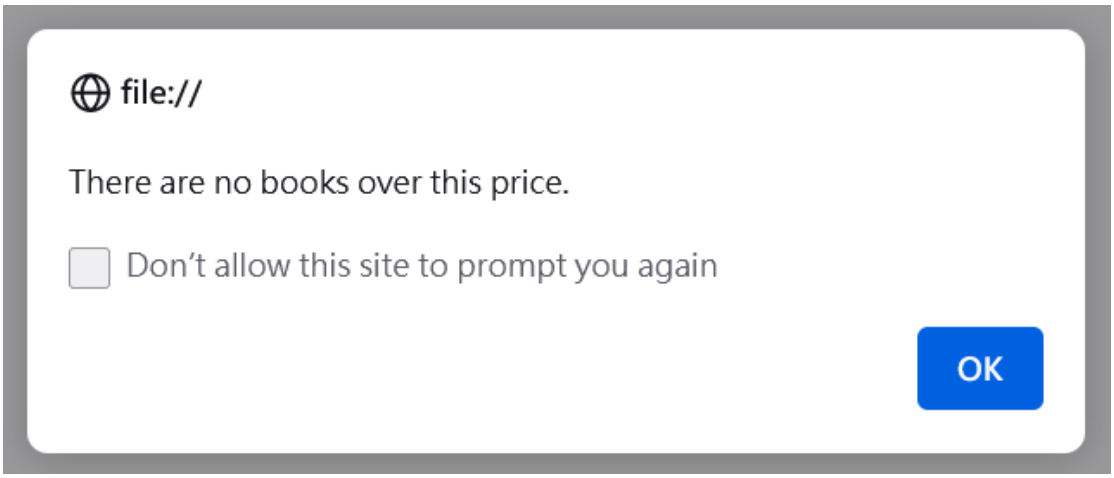
The function first asks the users to input a price then goes into the array to search for any price that is over the specified price; as there might be multiple books over the specified price, the loop once again does not break at the first matched result. Similarly, if there are no books over the specified price, the function alerts the users that no books are over the price they input.



```
>> printOverPrice();
   'The Legend of Zelda: Tri Force Heroes' is over 10 pounds, the price is 14.99
   'Go Set A Watchman' is over 10 pounds, the price is 10.89
<- "10"
```

Now see what happens if we input a price that is too high:





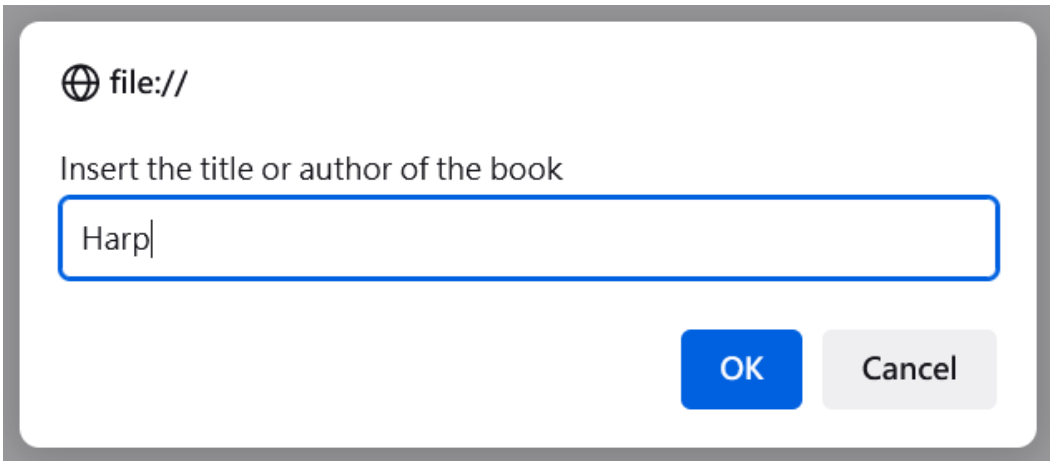
- ## Part1.2.6 search(text) Solution

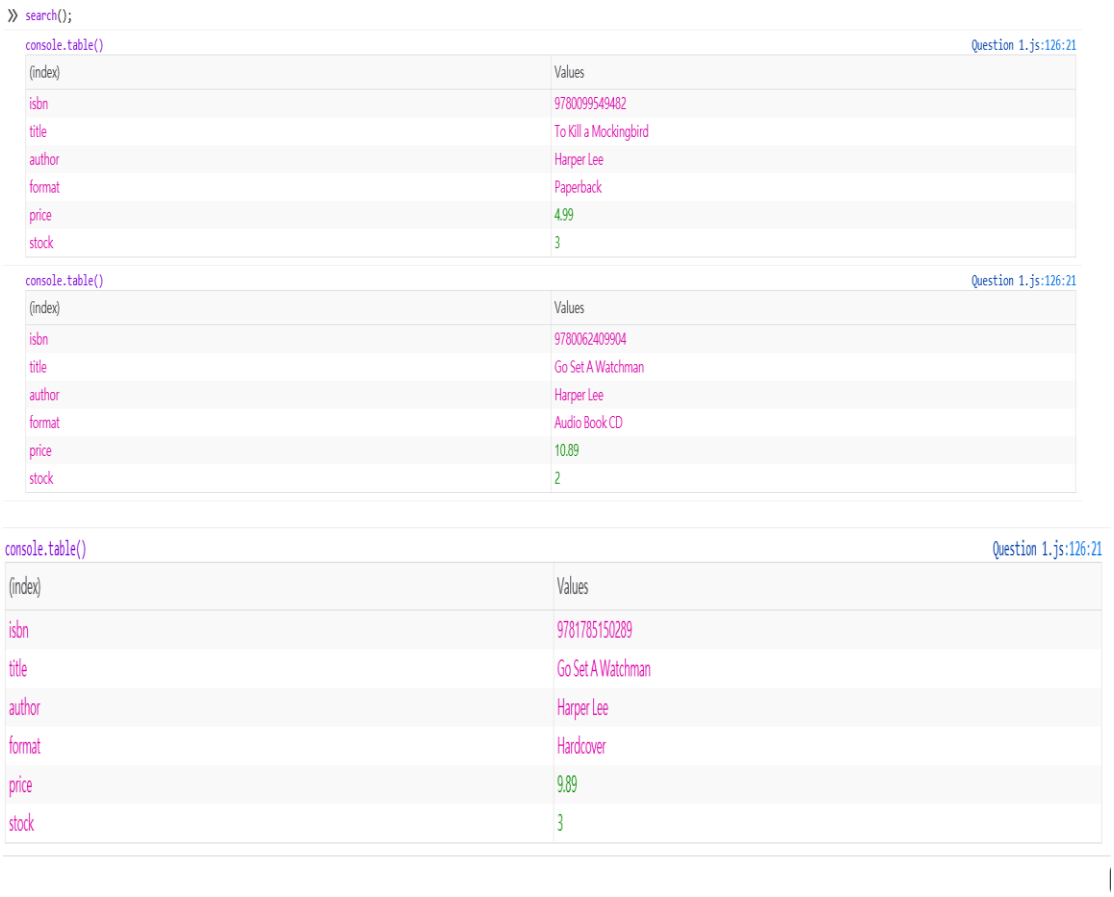
I first assigned a variable j equals to a hundred for later use; I also assigned two variables (tempTi and tempAu) with empty strings. The users would then be told to prompt a text, which would later be transferred into lowercases in the for loop I designed; the punctuation and letters would also be ignored to satisfy the requirement. Since the text should search into the property of author and title, I created two conditional statements in the loop to distinguish which property matched the text, the code of for loop is as follows:

```
for (i=0; i<list.length; i++) {
    let switchCases = text.toLowerCase().replace(/\W+/g, '');
    let tempTi = list[i].title.toLowerCase().replace(/\W+/g, '');
    let tempAu = list[i].author.toLowerCase().replace(/\W+/g, '');
    if (tempTi.indexOf(switchCases) >= 0) {
        console.table(list[i]);
        j = j * 17; //Change the number so we know whether the text is found in title or author.
        checkTi = list[i].title;
    }
    if (tempAu.indexOf(switchCases) >= 0) {
        console.table(list[i]);
        j = j * 13; //Change the number so we know whether the text is found in title or author.
        checkAu = list[i].author;
    }
}
if ((j % 17 == 0) && (text != checkTi)) {
    alert('We have found the book. However, you\'re spelling was incorrect. Please check the table.');
}
if ((j % 13 == 0) && (text != checkAu)) {
    alert('We have found the author. However, you\'re spelling was incorrect. Please check the table.');
}
if (j == 100) {
    alert('We cannot find any author or title that matches.')
}
```

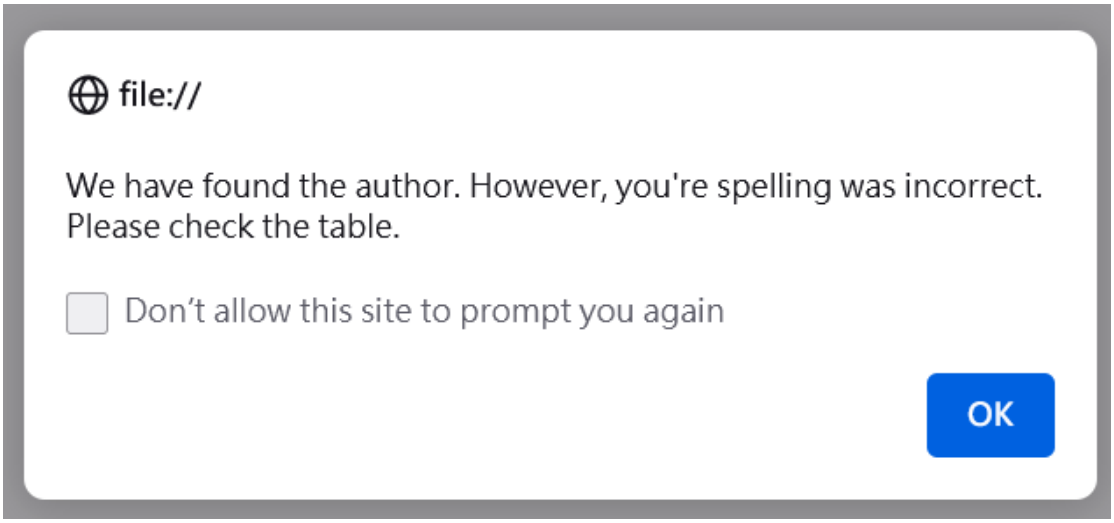
As seen in the screenshot, I also changed the value of variable j by multiplying a prime number for Javascript to distinguish what content of the alert should be when dealing with incorrect or incomplete spelling.

The following is the test result:





It also alerts the users for incomplete spelling:



- ## Part1.3 Difficulty

The difficulties I faced when solving this question were mostly about knowing which tool to use under the given circumstance. After watching the videos every week and doing some extra research by myself, I have more tools than I used to have in the beginning of the semester. The problem now is knowing when, and where to utilise them. For example, I had to refresh my memories of Regex and

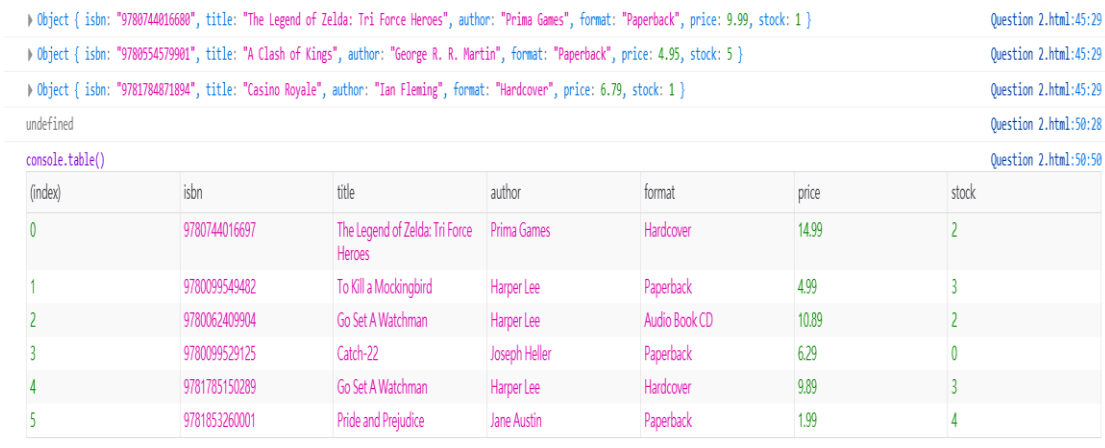
do some research on objects to satisfy some of the requirements and make the array easier to read and modify. On the other hand, I am more familiar with for loops and find it extremely useful when dealing with situations that require to repeat certain instructions until a condition is satisfied. I did not use new functions or subscript in this question, it was mostly extending the ways of applying them into the code based on previous knowledge and fundamental.

Part2.1 Question Description

This question started with how ISBN is calculated and then asked us to build a function that does the calculation and returns whether the ISBN is correct or not. The second function should be able to use the previous function and search through the whole database, list out the books with incorrect ISBN and extract them from the database.

Part2.2 Solution

To solve this question, I decided to create a nested function. The first layer of the function was only for the use of creating a string of ISBN from each book and triggering the second function. The second layer of the function, however, was more complicated. I first used 'split' to split the ISBN from each book into an array number by number, then created two for loops to multiply the odd elements of the array by one and the even elements by three respectively. Next, I replaced the last element of the array with the number calculated according to the formula given in the specification. I then returned this value back to the array and reunited the array back to a string so it could compare with the ISBN in the database. Lastly, I used a conditional statement to list out the detail of the book where its ISBN was not correct and display the database after all extraction was done. The following picture demonstrates which books have false ISBN and gets extracted out of the database:



- ## Part2.3 Difficulty

This part was rather easy, as it required more mathematical logics then coding logics. One new thing that I found out was when building a nested for loops, the two variables we assign in each loop should be different from the other otherwise the interpreter will not be able to interpret them.

- ## Part 3.1 Question description

There are two parts of the question. The first part we were asked to create a 'sort' function that is already built-in in Javascript. In this function, we should log the details of books sorted either by price, title, or author. The function should also support sorting in reverse order. The second part was to create an array with large elements using Math.random() and sorted the elements using Bubble Sort and Selection Sort.

- ## Part 3.2 Solution

In this part, I started off with the 'sort' function that is already built-in in Javascript. I looked up to MDN Web docs for information and found the code was rather easy. Simply return the value of a minus b, if it is less than zero, which means a is less than b, hence a will be sorted before b; if it equals to zero, meaning the two values are the same, no change will be made; if the value is greater than zero, meaning a is greater than b, then a will be sorted after b. If we decide to reverse the order by making it into a descending sequence, simply return the value of b minus a and the sequence will be displayed as required.

For this part I had to screenshot my results in Google Chrome browser as the one in Firefox browser kept displaying price descending sequence only.

The followings are tables for price ascending and descending:

Question 3.html:30

(index)	isbn	title	author	format	price	stock
0	'978-1-853-26000-1'	'Pride and Prejudice'	'Jane Austin'	'Paperback'	1.99	4
1	'978-0-554-57990-1'	'A Clash of Kings'	'George R. R. Martin'	'Paperback'	4.95	5
2	'978-0-099-54948-2'	'To Kill a Mockingbird'	'Harper Lee'	'Paperback'	4.99	3
3	'978-0-099-52912-5'	'Catch-22'	'Joseph Heller'	'Paperback'	6.29	0
4	'978-1-784-87189-4'	'Casino Royale'	'Ian Fleming'	'Hardcover'	6.79	1
5	'978-1-785-15028-9'	'Go Set A Watchman'	'Harper Lee'	'Hardcover'	9.89	3
6	'978-0-744-01668-0'	'The Legend of Zelda: Tr…	'Prima Games'	'Paperback'	9.99	1
7	'978-0-062-40990-4'	'Go Set A Watchman'	'Harper Lee'	'Audio Book CD'	10.89	2
8	'978-0-744-01669-7'	'The Legend of Zelda: Tr…	'Prima Games'	'Hardcover'	14.99	2

(index)	isbn	title	author	format	price	stock
0	'978-0-744-01669-7'	'The Legend of Zelda: Tr…	'Prima Games'	'Hardcover'	14.99	2
1	'978-0-062-40990-4'	'Go Set A Watchman'	'Harper Lee'	'Audio Book CD'	10.89	2
2	'978-0-744-01668-0'	'The Legend of Zelda: Tr…	'Prima Games'	'Paperback'	9.99	1
3	'978-1-785-15028-9'	'Go Set A Watchman'	'Harper Lee'	'Hardcover'	9.89	3
4	'978-1-784-87189-4'	'Casino Royale'	'Ian Fleming'	'Hardcover'	6.79	1
5	'978-0-099-52912-5'	'Catch-22'	'Joseph Heller'	'Paperback'	6.29	0
6	'978-0-099-54948-2'	'To Kill a Mockingbird'	'Harper Lee'	'Paperback'	4.99	3
7	'978-0-554-57990-1'	'A Clash of Kings'	'George R. R. Martin'	'Paperback'	4.95	5
8	'978-1-853-26000-1'	'Pride and Prejudice'	'Jane Austin'	'Paperback'	1.99	4

As for strings comparison, I first assigned two variables which represented a and b respectively, then transferred both string into lowercases. Then it became similar to the numeric value comparison, I built three conditional statements that would execute the instructions depended on the remainder. Similarly, if we would like to build a descending sequence, simply give the conditional statements into the opposite instructions and the result would be the opposite as well.

Title ascending:

(index)	isbn	title	author	format	price	stock
0	'978-0-554-57990-1'	'A Clash of Kings'	'George R. R. Martin'	'Paperback'	4.95	5
1	'978-1-784-87189-4'	'Casino Royale'	'Ian Fleming'	'Hardcover'	6.79	1
2	'978-0-099-52912-5'	'Catch-22'	'Joseph Heller'	'Paperback'	6.29	0
3	'978-0-062-40990-4'	'Go Set A Watchman'	'Harper Lee'	'Audio Book CD'	10.89	2
4	'978-1-785-15028-9'	'Go Set A Watchman'	'Harper Lee'	'Hardcover'	9.89	3
5	'978-1-853-26000-1'	'Pride and Prejudice'	'Jane Austin'	'Paperback'	1.99	4
6	'978-0-744-01669-7'	'The Legend of Zelda: Tr…	'Prima Games'	'Hardcover'	14.99	2
7	'978-0-744-01668-0'	'The Legend of Zelda: Tr…	'Prima Games'	'Paperback'	9.99	1
8	'978-0-099-54948-2'	'To Kill a Mockingbird'	'Harper Lee'	'Paperback'	4.99	3

Title descending:

(index)	isbn	title	author	format	price	stock
0	'978-0-099-54948-2'	'To Kill a Mockingbird'	'Harper Lee'	'Paperback'	4.99	3
1	'978-0-744-01669-7'	'The Legend of Zelda: Tr…	'Prima Games'	'Hardcover'	14.99	2
2	'978-0-744-01668-0'	'The Legend of Zelda: Tr…	'Prima Games'	'Paperback'	9.99	1
3	'978-1-853-26000-1'	'Pride and Prejudice'	'Jane Austin'	'Paperback'	1.99	4
4	'978-0-062-40990-4'	'Go Set A Watchman'	'Harper Lee'	'Audio Book CD'	10.89	2
5	'978-1-785-15028-9'	'Go Set A Watchman'	'Harper Lee'	'Hardcover'	9.89	3
6	'978-0-099-52912-5'	'Catch-22'	'Joseph Heller'	'Paperback'	6.29	0
7	'978-1-784-87189-4'	'Casino Royale'	'Ian Fleming'	'Hardcover'	6.79	1
8	'978-0-554-57990-1'	'A Clash of Kings'	'George R. R. Martin'	'Paperback'	4.95	5

Author ascending:

Question 3.html:64

(index)	isbn	title	author	format	price	stock
0	'978-0-554-57990-1'	'A Clash of Kings'	'George R. R. Martin'	'Paperback'	4.95	5
1	'978-0-099-54948-2'	'To Kill a Mockingbird'	'Harper Lee'	'Paperback'	4.99	3
2	'978-0-062-40990-4'	'Go Set A Watchman'	'Harper Lee'	'Audio Book CD'	10.89	2
3	'978-1-785-15028-9'	'Go Set A Watchman'	'Harper Lee'	'Hardcover'	9.89	3
4	'978-1-784-87189-4'	'Casino Royale'	'Ian Fleming'	'Hardcover'	6.79	1
5	'978-1-853-26000-1'	'Pride and Prejudice'	'Jane Austin'	'Paperback'	1.99	4
6	'978-0-099-52912-5'	'Catch-22'	'Joseph Heller'	'Paperback'	6.29	0
7	'978-0-744-01669-7'	'The Legend of Zelda: Tr…	'Prima Games'	'Hardcover'	14.99	2
8	'978-0-744-01668-0'	'The Legend of Zelda: Tr…	'Prima Games'	'Paperback'	9.99	1

Author descending:

Question 3.html:87

(index)	isbn	title	author	format	price	stock
0	'978-0-744-01669-7'	'The Legend of Zelda: Tr…	'Prima Games'	'Hardcover'	14.99	2
1	'978-0-744-01668-0'	'The Legend of Zelda: Tr…	'Prima Games'	'Paperback'	9.99	1
2	'978-0-099-52912-5'	'Catch-22'	'Joseph Heller'	'Paperback'	6.29	0
3	'978-1-853-26000-1'	'Pride and Prejudice'	'Jane Austin'	'Paperback'	1.99	4
4	'978-1-784-87189-4'	'Casino Royale'	'Ian Fleming'	'Hardcover'	6.79	1
5	'978-0-099-54948-2'	'To Kill a Mockingbird'	'Harper Lee'	'Paperback'	4.99	3
6	'978-0-062-40990-4'	'Go Set A Watchman'	'Harper Lee'	'Audio Book CD'	10.89	2
7	'978-1-785-15028-9'	'Go Set A Watchman'	'Harper Lee'	'Hardcover'	9.89	3
8	'978-0-554-57990-1'	'A Clash of Kings'	'George R. R. Martin'	'Paperback'	4.95	5

▶ Array(0)

Bubble Sort and Selection Sort required more logical thinking. According to my research, Bubble Sort starts looking for the maximum (or minimum, depends on which order is requested) value and puts the element at the right end of the array and runs the code repeatedly until it reaches the left end, the idea is to compare the elements that are next to each other and switch when necessary. Selection Sort, however, starts from the left end of the array; it runs through the whole array and assigns the minimum (or maximum, depends on which order is requested) value of the elements to the first place. After finishing the loop, the sequence will be in order. One major difference between the two is that Selection Sort is considered more efficient than Bubble Sort.

To build a Bubble Sort function, I started with a nested for loop; The first layer represented the location of element I wanted to assign in each round, hence it stopped the loop when variable i is greater than zero (We do not need to assign a value for the element on the left end as it is already the minimum value). The second loop represented how many times we wanted to compare the elements (It started comparing from the left end all the way to the right); hence the condition was to exit the loop when variable j was no longer less than i. I also built a conditional statement inside the second loop so it would switch the elements when the previous one was greater than the next one.

In the following screenshots, there will only be 10 numbers in each array as it is easier to see the result. However, I created an array with 5000 elements in the code to satisfy the requirement.

Bubble Sort result:

```
>> bubbleSort();
    ▶ Array(10) [ 8387, 1057, 6170, 3376, 3953, 9435, 7971, 2967, 749, 8568 ]
    ▶ Array(10) [ 749, 1057, 2967, 3376, 3953, 6170, 7971, 8387, 8568, 9435 ]
```

To build a Selection Sort function, I also started with a nested for loop; Some minor difference between the two was i started from the left end because we also started assigning from the left; j started next to i so it would run through the loop, find the minimum value and switch with the first element when spotted a smaller value.

Selection Sort result:

```
>> selectionSort();
    ▶ Array(10) [ 5792, 2885, 4609, 4546, 7750, 2003, 1414, 7310, 8346, 5673 ]
    ▶ Array(10) [ 1414, 2003, 2885, 4546, 4609, 5673, 5792, 7310, 7750, 8346 ]
```

Part 3.3 Difficulty

Despite having a shorter description and less requirements, I faced more difficulties in this part than the first two. The 'sort' function was new to me, so it took me awhile to figure out how it really worked on MDN Web Docs (https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/sort). Bubble Sort and Selection Sort seemed easy to understand, but it was very hard to apply them into the code. I watched a video on YouTube from a Taiwanese (https://www.youtube.com/watch?v=i-0wxW5Aun4&t=1703s) teaching Bubble Sort so I could apply the codes into my assignment. As for Selection Sort, I could not find a clear video that introduced the function so I coded on my own after reading the definition on Wikipedia, which has an example of Selection Sort in programming language C (https://en.wikipedia.org/wiki/Selection_sort).

Conclusion

I found this assignment interesting and practical; However, it still took me days to finish it. While for people who are familiar with coding it might take less than a day, I am still satisfied with my accomplishment and have a goal of improving

myself. As I referred to in the previous report, coding is something I have never done but I do enjoy the process, it is way more interesting than I thought if you have all the tools to use. You feel like a genius when you come up with ways that display as many functions with as little code as possible. Some of the codes in my program could be shorter I assume, but I believe I have done the best I could to satisfy all the requirements and hopefully as I become more experienced with coding, my code will look shorter and more efficient.