(/)

We're moving to Discord!

In a few days, we will be leaving Slack in favor of Discord 🎉

👆 **Click here for more information (/concepts/100033)**

Javascript advanced

JavaScript

⤴ Master

👤 By: Johann Kerbrat, Engineering Manager at Uber Works

⚙ Weight: 1

☑ **Manual QA review must be done** (request it when you are done with the project)

Resources

Read or watch:

- Lexical scoping (/rltoken/gtNKxnTsTcqTHxSO7IxG1A)
- Closure (/rltoken/nkUIwlOmos3Dp5H48xB8NA)
- Call stack (/rltoken/m4N2NfiZaD3DuHT16DXndg)
- Binding (/rltoken/e4qDWCEOZPHUGqYizpR0oA)
- Callbacks (/rltoken/0FyfBzMjE_PwuHplDFD36A)

Learning Objectives

- What is lexical scoping in Javascript
- What is closure in Javascript
- How to use closure
- How to chain different closures
- How to simulate private methods with Closure
- The execution stack order with Javascript
- How to use binding
- How to use callbacks

Requirements

- Recommended editors: `Visual Studio Code`
- All your files should end with a new line
- A `README.md` file, at the root of the folder of the project, is mandatory
- Your code should use the `js` extension

Quiz questions

Great! You've completed the quiz successfully! Keep going! (Show quiz)

Tasks

0. Lexical scoping and welcome message

mandatory

Create a function named `welcome` :

- It takes two arguments: `firstName` (string) and `lastName` (string)
- It contains a variable named `fullName` , that will contains the `firstName` followed by a space and then the `lastName`
- Within the `welcome` function, write a function named `displayFullName` :
    - It should display an alert with the message `Welcome` followed by a space, then the variable `fullName` followed by an exclamation mark.
- Call the function `displayFullName` at the end of the function `welcome`

How to test

- Open your web inspector in the tab “Console”
- Copy paste your code in the console and hit enter
- Copy and paste this `welcome('Holberton', 'School');` and hhit enter to run that code. It should prompt an alert with this content: `Welcome Holberton School!`
- When you write and execute `alert(fullName)` afterwards, it should return a reference error `fullName is not defined`

Repo:

- GitHub repository: `alx_javascript`
- Directory: `Javascript_advanced`
- File: `0-welcome.js`

Help

0/12 pts

1. Closure Scope Chain

mandatory

- Create a variable named `globalVariable` with value `Welcome`
- Create a function `outer` that:
  - alerts the content of the variable `globalVariable`
  - creates a variable named `course` with value `Holberton`
  - creates a function `inner` that:
    - alerts the content of the variable `globalVariable` and `course` (concatenated)
    - creates a variable named `exclamation` with value `!`
    - creates a function `inception` that alerts the content of the variable `globalVariable` , `course` , and `exclamation` (concatenated)
    - calls the function `inception`
  - calls the function `inner`
- Call the function `outer`

Compose the code:

- Write the function `inception` within `inner`
- Write the function `inner` within `outer`
- Call the function `outer` in the main code (outside any function)
- Call the function `inner` within `outer`
- Call the function `inception` within `inner`

Requirements:

- Running the script should display three popups one by one with the text `Welcome` , `Welcome Holberton` , and `Welcome Holberton!`

Repo:

- GitHub repository: `alx_javascript`
- Directory: `Javascript_advanced`
- File: `1-nested_functions.js`

Help

0/19 pts

2. Closure

mandatory

Write a function named `welcomeMessage` :

- It accepts one argument `fullName` (string)
- It should be a closure for an alert displaying `Welcome <fullName>`

After this function definition, create three variables:

- `guillaume` contains a call `welcomeMessage` with `Guillaume` as argument
- `alex` contains a call `welcomeMessage` with `Alex` as argument

- `fred` contains a call `welcomeMessage` with `Fred` as argument

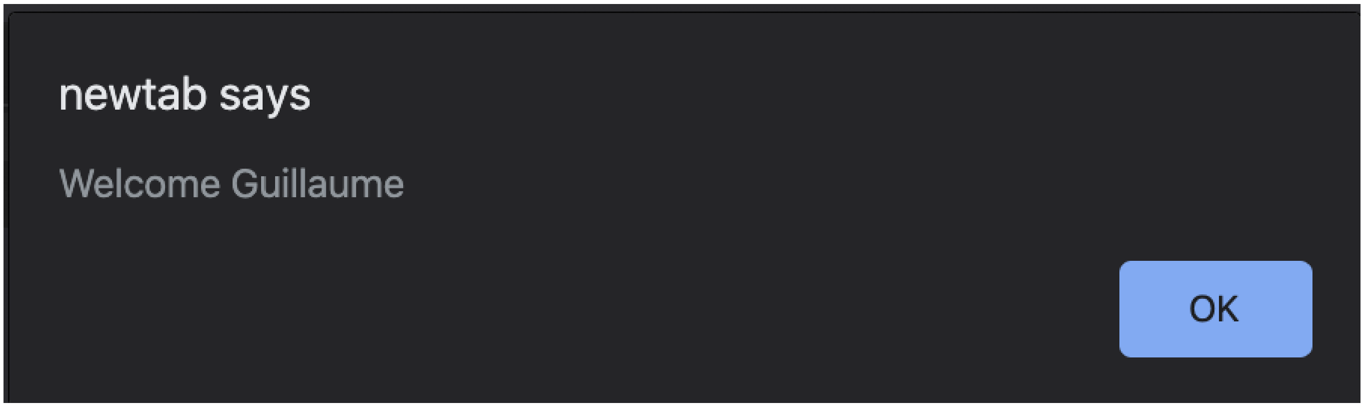
(/)

How to test:

- In your web console, cop/paste your file
- Executing the following in the console:

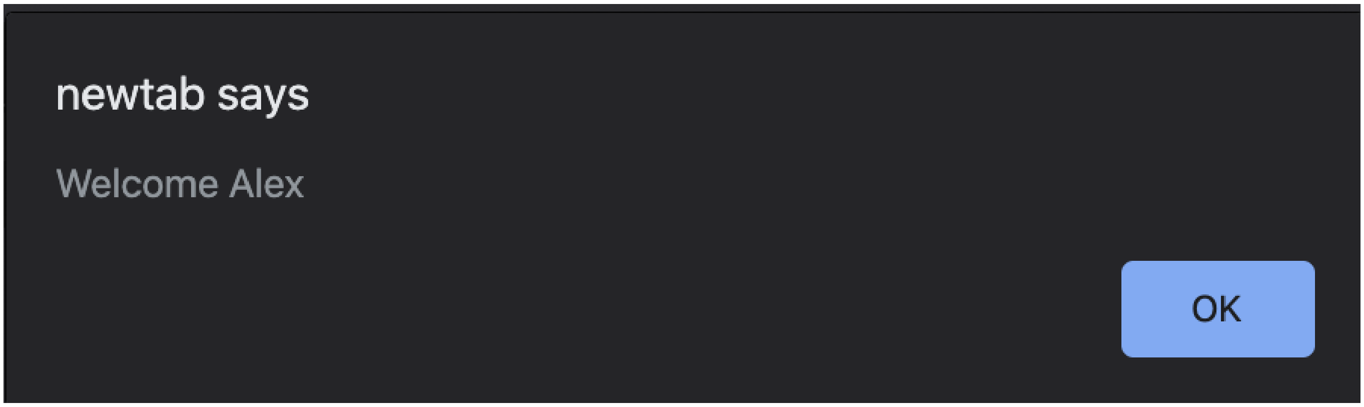
```
guillaume();
alex();
fred();
```

Should display three alerts like below:



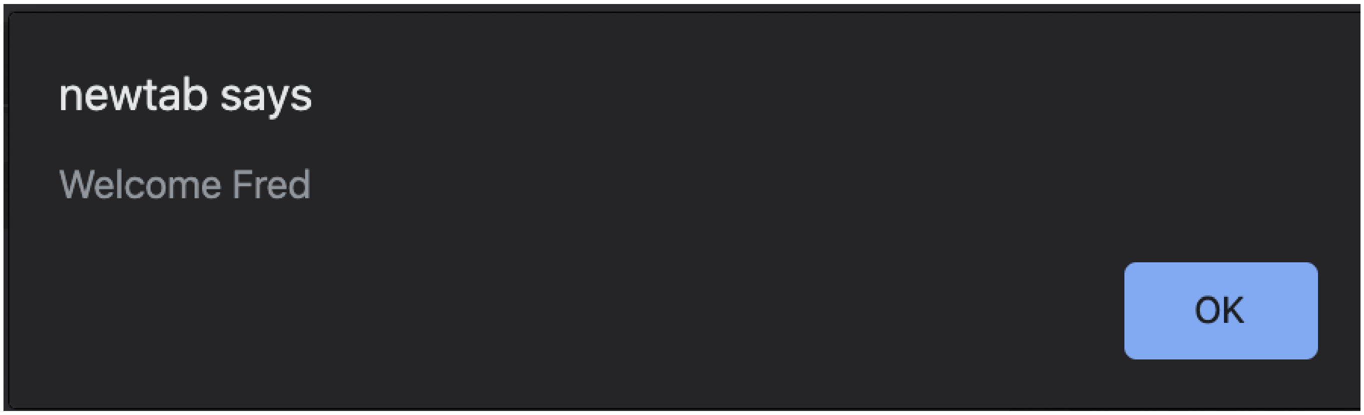
newtab says

Welcome Guillaume



newtab says

Welcome Alex



newtab says

Welcome Fred

Repo:

- GitHub repository: `alx_javascript`
- Directory: `Javascript_advanced`

- File: `2-function_me.js`

(/)

Help **0/13** pts

3. Closure and loops

mandatory

Write a function named `createClassRoom` :

- It takes into argument `numbersOfStudents` (number)
- Inside, it contains a function `studentSeat` , that takes into argument `seat` (number) and returns a function that returns the `seat` number
- After the definition of `studentSeat` , create and populate a variable `students` (array)
  - Using a loop from 0 to `numbersOfStudents` , pass the number of iteration + 1 to `studentSeat` and add its return value to the `students` array
- Returns the `students` array

Create a closure `classRoom` , calling `createClassRoom` with 10 students

Requirements:

Executing the following code:

```
console.log(classRoom[0]());
console.log(classRoom[3]());
console.log(classRoom[9]());
```

Should return

```
1
4
10
```

Repo:

- GitHub repository: `alx_javascript`
- Directory: `Javascript_advanced`
- File: `3-classrooms.js`

Help **0/12** pts

4. Complex Closure

mandatory

Create a function `divideBy` :

- It takes into argument `firstNumber` (number)

- It returns a function that takes into argument `secondNumber` (number)
    - It returns the second number divided by the first number

(/)

Create a function `addBy` :

- It takes into argument `firstNumber` (number)
- It returns a function that takes into argument `secondNumber` (number)
    - It returns the sum of the two numbers

Create four closures:

- `addBy100` , that uses the function `addBy` with the number 100
- `addBy1000` , that uses the function `addBy` with the number 1000
- `divideBy10` , that uses the function `divideBy` with the number 10
- `divideBy100` , that uses the function `divideBy` with the number 100

Requirements:

Executing the following code:

```
console.log(addBy100(20));
console.log(divideBy10(20));
console.log(divideBy100(200));
console.log(addBy1000(20));
```

Should display in the console:

```
120
2
2
1020
```

Repo:

- GitHub repository: `alx_javascript`
- Directory: `Javascript_advanced`
- File: `4-math.js`

Help **0/17** pts

5. Changing DOM with closure

mandatory

Create a function named `changeMode` :

- It accepts 5 arguments `size` (number), `weight` (string), `transform` (string), `background` (string), `color` (string)
- Using closure, it changes the style of the entire page and set the `font-size` , `font-weight` , `text-transform` , `background-color` , and `color`

Write a function named `main` :

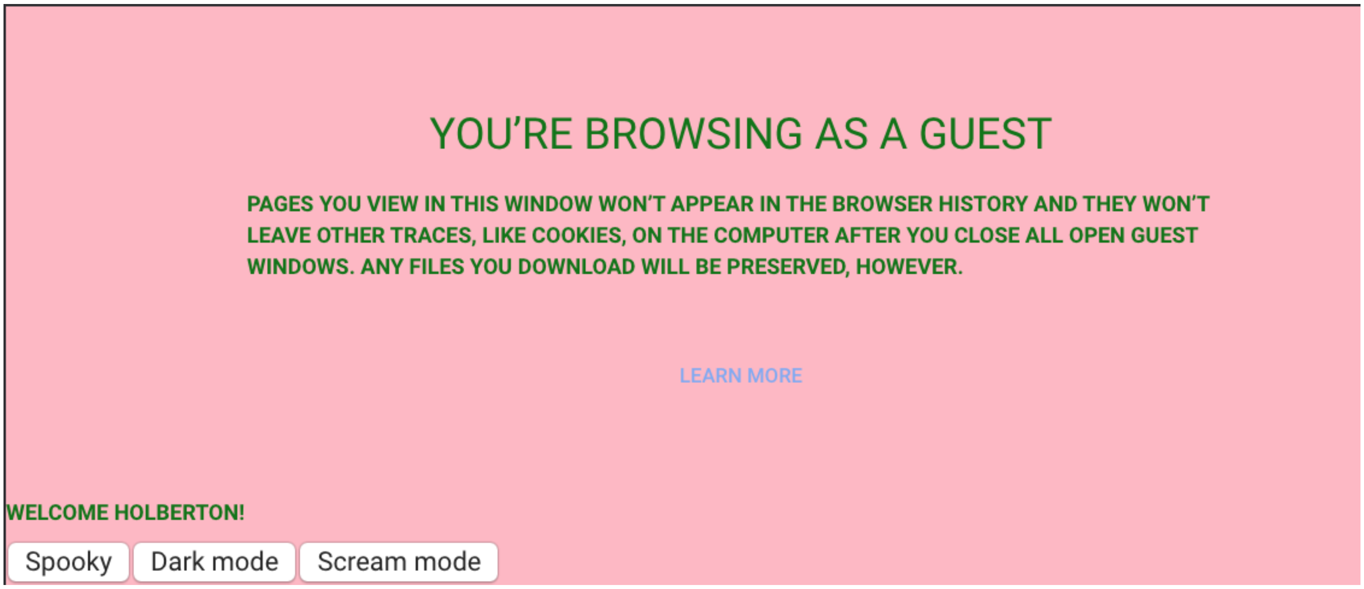
- Set a variable named `spooky` that passes the arguments `9`, `bold`, `uppercase`, `pink`, `green` to `changeMode`
- Set a variable named `darkMode` that passes the arguments `12`, `bold`, `capitalize`, `black`, `white` to `changeMode`
- Set a variable named `screamMode` that passes the arguments `12`, `normal`, `lowercase`, `white`, `black` to `changeMode`
- Add a paragraph to the body of the page with the text `Welcome Holberton!`
- Add a button to the body with the text `Spooky`
- Add a button to the body with the text `Dark mode`
- Add a button to the body with the text `Scream mode`
- When clicking on each button, the page CSS should change to the different themes you created previously

Call the `main` function

Requirements:

- Use vanilla javascript to add the elements to the DOM and change the style of the page
- JQuery not allowed!

Clicking the `Spooky` button should change the web page to look something like this:

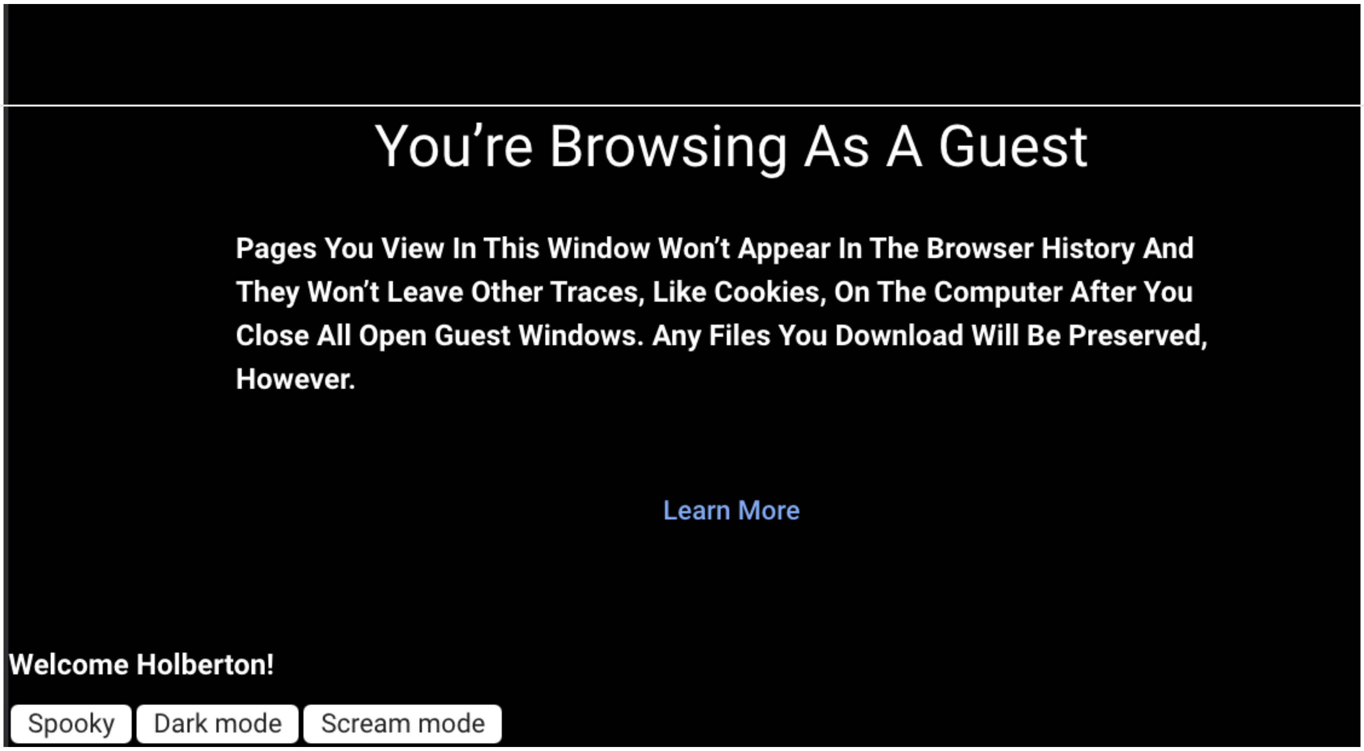


YOU’RE BROWSING AS A GUEST

PAGES YOU VIEW IN THIS WINDOW WON’T APPEAR IN THE BROWSER HISTORY AND THEY WON’T LEAVE OTHER TRACES, LIKE COOKIES, ON THE COMPUTER AFTER YOU CLOSE ALL OPEN GUEST WINDOWS. ANY FILES YOU DOWNLOAD WILL BE PRESERVED, HOWEVER.

WELCOME HOLBERTON!

Clicking the `Dark mode` button should change the web page to look something like this:

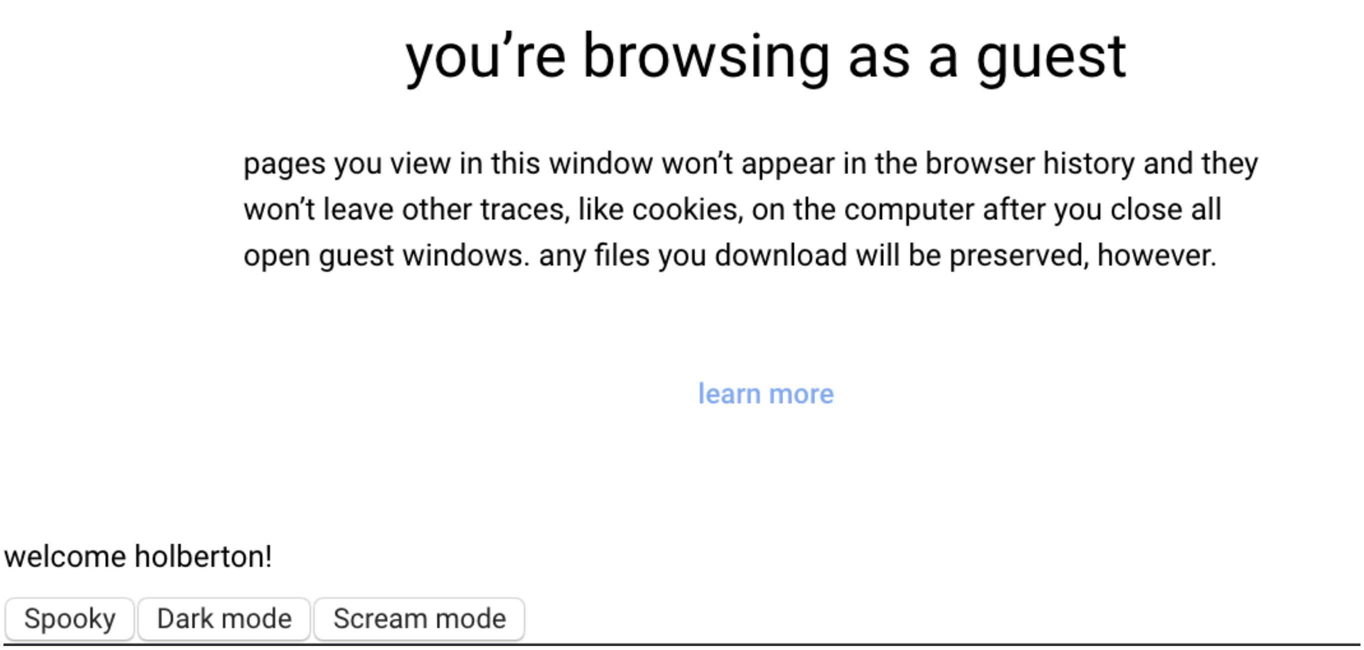


You’re Browsing As A Guest

Pages You View In This Window Won’t Appear In The Browser History And They Won’t Leave Other Traces, Like Cookies, On The Computer After You Close All Open Guest Windows. Any Files You Download Will Be Preserved, However.

Welcome Holberton!

Clicking the `Scream mode` button should change the web page to look something like this:



you’re browsing as a guest

pages you view in this window won’t appear in the browser history and they won’t leave other traces, like cookies, on the computer after you close all open guest windows. any files you download will be preserved, however.

welcome holberton!

Repo:

- GitHub repository: `alx_javascript`
- Directory: `Javascript_advanced`
- File: `5-mode.js`

Help

0/26 pts

6(/)Private methods with closure

mandatory

Write a module named `studentHogwarts` :

- It contains two variables `privateScore` set to 0, and `name` set to `null`
- It contains one private method `changeScoreBy` , it takes `points` in argument and add it to `privateScore`
- The module gives access to four public methods (return an object):
    - `setName` , it takes into argument `newName` , and set the private variable `name`
    - `rewardStudent` , it calls the method `changeScoreBy` with 1
    - `penalizeStudent` , it calls the method `changeScoreBy` with -1
    - `getScore` , it returns `name: score` (ex: `Harry: 14`)

Create one variable named `harry` , that is an instance of `studentHogwarts` :

- Set the name of the object to `Harry`
- Reward the student four times
- Log to the console the name and score

Create one variable named `draco` , that is an instance of `studentHogwarts` :

- Sets the name of the object to `Draco`
- Reward the student one time
- Penalize the student three times
- Log to the console the name and score

Requirements:

- Executing the code should display in the console

```
Harry: 4
Draco: -2
```

Repo:

- GitHub repository: `alx_javascript`
- Directory: `Javascript_advanced`
- File: `6-hogwarts.js`

Help

0/19 pts

7. Stack order and setTimeout

mandatory

Write the following commands in the following order:

- Log to the console `Start of the execution queue`
- Log to the console `Final code block to be executed` using `setTimeout` (with delay of 0)

- Using a loop that iterates 100 times, each iteration logs to the console the iteration number
- Log to the console `End of the loop printing`

Requirements:

- Your code should log to the console the following:

```
Start of the execution queue
1
2
...
100
End of the loop printing
Final code block to be executed
```

Repo:

- GitHub repository: `alx_javascript`
- Directory: `Javascript_advanced`
- File: `7-timeout.js`

Help

0/6 pts

Score

Now that you are ready to be reviewed, share your link to your peers. You can find some here.

https://intranet.alxswe.com/corrections/16931991/correct

Don't forget to review one of them (/corrections/to_review).

The next project will be available on Saturday, Sep 16th.

Skip this project

Previous project (/projects/2081)