

(/)



JavaScript - Warm up

**JavaScript**

📈 Novice

👤 By: Guillaume, CTO at Holberton School

⚙️ Weight: 1

☑️ Your score will be updated once you launch the project review.

Background Context

JavaScript is used for many things. At ALX, you will use JavaScript for 2 reasons:

- Scripting (same as we did with Python)
- Web front-end

For the moment, and for learning all basic concepts of this language, we will do some scripting.

(/)

Javascript



Please Just Work

MemeBucket.com

Resources

Read or watch:

- Writing JavaScript Code (/rltoken/APZmtL9w79192Tb5rnlZdw)
- Variables (/rltoken/uS2wx3lepQNsZv9s8QilHw)
- Data Types (/rltoken/-A8NrRXtbEQxgnsn5RcCdA)
- Operators (/rltoken/APZmtL9w79192Tb5rnlZdw)
- Operator Precedence (/rltoken/x-s3s429lZq24OcvBsmB-Q)
- Controlling Program Flow (/rltoken/RNqXLZl7X89l8j9ahmZpvQ)
- Functions (/rltoken/NfmE5LmeO3KJvSS9WgzFWw)
- Objects and Arrays (/rltoken/tii5tBRqPGmZSLdIYPphGQ)
- Intrinsic Objects (/rltoken/tii5tBRqPGmZSLdIYPphGQ)
- Module patterns (/rltoken/BO7eZ67y8m5IN73fOOBZUg)
- var, let and const (/rltoken/Hv8cxS35QWiJUcBJKMA76g)
- JavaScript Tutorial (/rltoken/R2STzQtaj4RWSsadh4cYeA)
- Modern JS (/rltoken/AM2vFBysBJNU37w8N1zZog)

Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (/ritoken/zobPXLmbSg12-dzzyNc3xQ), **without the help of Google**:

General

- Why JavaScript programming is amazing
- How to run a JavaScript script
- How to create variables and constants
- What are differences between `var`, `const` and `let`
- What are all the data types available in JavaScript
- How to use the `if`, `if ... else` statements
- How to use comments
- How to affect values to variables
- How to use `while` and `for` loops
- How to use `break` and `continue` statements
- What is a function and how do you use functions
- What does a function that does not use any `return` statement return
- Scope of variables
- What are the arithmetic operators and how to use them
- How to manipulate dictionary
- How to import a file

Requirements

General

- Recommended editors: `Visual studio code`
- All your files will be interpreted on Ubuntu 20.04 LTS using `node` (version 14.x)
- 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 be `semistandard` compliant (version 14.x.x). Rules of Standard (/ritoken/RDNP0eE9QGnCVowPzKhkAQ) + semicolons on top (/ritoken/jWkz1MC29tQb1WAZFhPm8g). Also as reference: AirBnB style (/ritoken/xejcYnyIEcDJWI1sCz5PKA)
- The length of your files will be tested using `wc`

More Info

Install Node 14

```
$ curl -sL https://deb.nodesource.com/setup_14.x | sudo -E bash -  
$ sudo apt-get install -y nodejs
```

Install semi-standard

Documentation (/rltoken/jWkz1MC29tQb1WA7FhPm8g)

```
$ sudo npm install semistandard --global
```

Intro Session for this project

Javascript 1st September 2023



Quiz questions

Great! You've completed the quiz successfully! Keep going! ([Show quiz](#)).

Tasks

0. First constant, first print

mandatory

Write a script that prints "JavaScript is amazing":

- You must create a constant variable called `myVar` with the value "JavaScript is amazing"
- You must use `console.log(...)` to print all output
- You are not allowed to use `var`

```
guillaume@ubuntu:~/ $ node 0-javascript_is_amazing.js
JavaScript is amazing
guillaume@ubuntu:~/ $
guillaume@ubuntu:~/ $ semistandard 0-javascript_is_amazing.js
guillaume@ubuntu:~/ $
```

Repo:

- GitHub repository: alx_javascript
- Directory: javascript-warm_up
- File: 0-javascript_is_amazing.js

Help

Check your code

> Get a sandbox

5/5 pts**1. 3 languages****mandatory**

Write a script that prints 3 lines:

- The first line: "C is fun"
- The second line: "Python is cool"
- The third line: "JavaScript is amazing"
- You must use `console.log(...)` to print all output
- You are not allowed to use `var`

```
guillaume@ubuntu:~/ $ node 1-multi_languages.js
C is fun
Python is cool
JavaScript is amazing
guillaume@ubuntu:~/ $
```

Repo:

- GitHub repository: alx_javascript
- Directory: javascript-warm_up
- File: 1-multi_languages.js

Help

Check your code

> Get a sandbox

5/5 pts**2. Loop to languages****mandatory**

Write a script that prints 3 lines: (like `1-multi_languages.js`) but by using an array of string and a loop

- The first line: "C is fun"

- The second line: "Python is cool"
- (/)
- The third line: "JavaScript is amazing"
- You must use `console.log(...)` to print all output
- You are not allowed to use `var`
- You are not allowed to use any `if/else` statement
- You can use only one `console.log`
- You must use a loop (`while`, `for`, etc.)

```
guillaume@ubuntu:~/ $ node 6-multi_languages_loop.js
C is fun
Python is cool
JavaScript is amazing
guillaume@ubuntu:~/ $
```

Repo:

- GitHub repository: `alx_javascript`
- Directory: `javascript-warm_up`
- File: `6-multi_languages_loop.js`

Help

Check your code

> Get a sandbox

10/10 pts**3. Object****mandatory**

Update this script to replace the value `12` with `89` :

- You are not allowed to use `var`

```
guillaume@ubuntu:~/ $ cat 12-object.js
#!/usr/bin/node
const myObject = {
  type: 'object',
  value: 12
};
console.log(myObject);
/*
YOUR CODE HERE
*/
console.log(myObject);


guillaume@ubuntu:~/ $ node 12-object.js
{ type: 'object', value: 12 }
{ type: 'object', value: 89 }
guillaume@ubuntu:~/ $
```

Repo:

- GitHub repository: alx_javascript
- (/)- Directory: javascript-warm_up
- File: 12-object.js

Help

Check your code

 Get a sandbox

6/6 pts

4. Add file

mandatory

Write a function that returns the addition of 2 integers.

- The function must be visible from outside
- The name of the function must be `add`
- You are not allowed to use `var`

Tip (/rltoken/Es7spmqqGPYa8G-7rYX5pQ)


```
guillaume@ubuntu:~/ $ cat 13-main.js
#!/usr/bin/node
const add = require('./13-add').add;
console.log(add(3, 5));
guillaume@ubuntu:~/ $ ./13-main.js
8
guillaume@ubuntu:~/ $
```

Repo:

- GitHub repository: alx_javascript
- Directory: javascript-warm_up
- File: 13-add.js

Help

Check your code

 Get a sandbox

7/7 pts

5. Const or not const

#advanced

Write a file that modifies the value of `myVar` to `333`

```
guillaume@ubuntu:~/ $ cat 100-main.js
#!/usr/bin/node
myVar = 89;
require('./100-let_me_const')
console.log(myVar);
guillaume@ubuntu:~/ $ ./100-main.js
333
guillaume@ubuntu:~/ $
```



Do you get it? Tweet! Post! Talk about it!

Hint: Scope

This exercise doesn't pass semistandard so don't worry about it.

Repo:

- GitHub repository: `alx_javascript`
- Directory: `javascript-warm_up`
- File: `100-let_me_const.js`

[Help](#)[Check your code](#)[Get a sandbox](#)**0/6 pts**

6. Call me Moby

#advanced

Write a function that executes `x` times a function.

- The function must be visible from outside
- Prototype: `function (x, theFunction)`
- You are not allowed to use `var`

```
guillaume@ubuntu:~/ $ cat 101-main.js
#!/usr/bin/node
const callMeMoby = require('./101-call_me_moby').callMeMoby;
callMeMoby(3, function () {
  console.log('C is fun');
});
guillaume@ubuntu:~/ $ ./101-main.js
C is fun
C is fun
C is fun
guillaume@ubuntu:~/ $
```


Repo:

- GitHub repository: alx_javascript
- Directory: javascript-warm_up
- File: 101-call_me_moby.js

Help

Check your code

>_ Get a sandbox

0/7 pts**7. Add me maybe**

#advanced

Write a function that increments and calls a function.

- The function must be visible from outside
- Prototype: function (number, theFunction)
- You are not allowed to use `var`

```
guillaume@ubuntu:~/ $ cat 102-main.js
#!/usr/bin/node
const addMeMaybe = require('./102-add_me_maybe').addMeMaybe;
addMeMaybe(4, function (nb) {
  console.log('New value: ' + nb);
});
guillaume@ubuntu:~/ $ ./102-main.js
New value: 5
guillaume@ubuntu:~/ $
```

Repo:

- GitHub repository: alx_javascript
- Directory: javascript-warm_up
- File: 102-add_me_maybe.js

Help

Check your code

>_ Get a sandbox

0/7 pts**8. Increment object**

#advanced

Update this script by adding a new function `incr` that increments the integer `value` .

- You are not allowed to use `var`

```
guillaume@ubuntu:~/ $ cat 103-object_fct.js
#!/usr/bin/node

const myObject = {
  type: 'object',
  value: 12
};
console.log(myObject);
/*
YOUR CODE HERE
*/
myObject.incr();
console.log(myObject);
myObject.incr();
console.log(myObject);
myObject.incr();
console.log(myObject);


guillaume@ubuntu:~/ $ ./103-object_fct.js
{ type: 'object', value: 12 }
{ type: 'object', value: 13, incr: [Function] }
{ type: 'object', value: 14, incr: [Function] }
{ type: 'object', value: 15, incr: [Function] }
guillaume@ubuntu:~/ $
```

Repo:

- GitHub repository: alx_javascript
- Directory: javascript-warm_up
- File: 103-object_fct.js

Help

Check your code

 Get a sandbox**0/6 pts****Score****100%****100%**

Congratulations! You made it!

Next project: JavaScript - Objects, Scopes and Closures



Open the next project (/projects/2080)

Previous project (/projects/2086)
(/)

Copyright © 2023 ALX, All rights reserved.