

(/)



Curriculum

Short Specializations

Average: 97.3%

0x02. ES6 classes

OOP

JavaScript

ES6

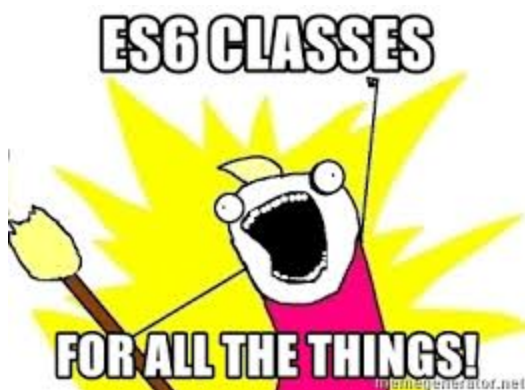
By: Johann Kerbrat, Engineering Manager at Uber Works

Weight: 1

 Project over - took place from Nov 30, 2023 6:00 AM to Dec 1, 2023 6:00 AM☒ An auto review will be launched at the deadline

In a nutshell...

- **Auto QA review:** 28.0/33 mandatory & 3.0/3 optional
- **Altogether: 169.7%**
 - Mandatory: 84.85%
 - Optional: 100.0%
 - Calculation: $84.85\% + (84.85\% * 100.0\%) == 169.7\%$



Resources

Read or watch:

- Classes (/rltoken/ke2dSL31JbpAUBW0qWE9WA)
- Metaprogramming (/rltoken/6OgF5QGbYclp_cwATfq-0g)



Learning Objectives

At the end of this project, you are expected to be able to explain to anyone (rltoken/HZvBI09eHoGwvN8jqIYO-g), **without the help of Google**:

- How to define a Class
- How to add methods to a class
- Why and how to add a static method to a class
- How to extend a class from another
- Metaprogramming and symbols

Requirements

- All your files will be executed on Ubuntu 18.04 LTS using NodeJS 12.11.x
- Allowed editors: `vi`, `vim`, `emacs`, `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
- Your code will be tested using `Jest` and the command `npm run test`
- Your code will be verified against lint using `ESLint`
- Your code needs to pass all the tests and lint. You can verify the entire project running `npm run full-test`

Setup

Install NodeJS 12.11.x

(in your home directory):

```
curl -sL https://deb.nodesource.com/setup_12.x -o nodesource_setup.sh
sudo bash nodesource_setup.sh
sudo apt install nodejs -y
```

```
$ nodejs -v
v12.11.1
$ npm -v
6.11.3
```

Install Jest, Babel, and ESLint

in your project directory, install Jest, Babel and ESLint by using the supplied `package.json` and run `npm install`.



Configuration files

Add the files below to your project directory

package.json

Click to show/hide file contents

babel.config.js

Click to show/hide file contents

.eslintrc.js

Click to show/hide file contents

and...

Don't forget to run `$ npm install` when you have the `package.json`

Tasks

0. You used to attend a place like this at some point

mandatory

Score: 100.0% (Checks completed: 100.0%)

Implement a class named `ClassRoom` :

- Prototype: `export default class ClassRoom`
- It should accept one attribute named `maxStudentsSize` (Number) and assigned to `_maxStudentsSize`

```
bob@dylan:~$ cat 0-main.js
import ClassRoom from "./0-classroom.js";

const room = new ClassRoom(10);
console.log(room._maxStudentsSize)

bob@dylan:~$
bob@dylan:~$ npm run dev 0-main.js
10
bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x02-ES6_classes`



- File: 0-classroom.js (/)

☒ Done!

1. Let's make some classrooms

mandatory

Score: 100.0% (Checks completed: 100.0%)

Import the `ClassRoom` class from `0-classroom.js`.

Implement a function named `initializeRooms`. It should return an array of 3 `ClassRoom` objects with the sizes 19, 20, and 34 (in this order).

```
bob@dylan:~$ cat 1-main.js
import initializeRooms from './1-make_classrooms.js';

console.log(initializeRooms());

bob@dylan:~$
bob@dylan:~$ npm run dev 1-main.js
[
  ClassRoom { _maxStudentsSize: 19 },
  ClassRoom { _maxStudentsSize: 20 },
  ClassRoom { _maxStudentsSize: 34 }
]
bob@dylan:~$
```

Repo:

- GitHub repository: alx-backend-javascript
- Directory: 0x02-ES6_classes
- File: 1-make_classrooms.js

☒ Done!

2. A Course, Getters, and Setters

mandatory

Score: 100.0% (Checks completed: 100.0%)



Implement a class named `HolbertonCourse` :

- Constructor attributes:
 - `name` (String)

- length (Number)
- (/) ◦ students (array of Strings)

- Make sure to verify the type of attributes during object creation
- Each attribute must be stored in an "underscore" attribute version (ex: name is stored in `_name`)
- Implement a getter and setter for each attribute.

```
bob@dylan:~$ cat 2-main.js
import HolbertonCourse from './2-hbtn_course.js';

const c1 = new HolbertonCourse("ES6", 1, ["Bob", "Jane"])
console.log(c1.name);
c1.name = "Python 101";
console.log(c1);

try {
  c1.name = 12;
}
catch(err) {
  console.log(err);
}

try {
  const c2 = new HolbertonCourse("ES6", "1", ["Bob", "Jane"]);
}
catch(err) {
  console.log(err);
}

bob@dylan:~$
bob@dylan:~$ npm run dev 2-main.js
ES6
HolbertonCourse {
  _name: 'Python 101',
  _length: 1,
  _students: [ 'Bob', 'Jane' ]
}
TypeError: Name must be a string
...
TypeError: Length must be a number
...
bob@dylan:~$
```

Repo:


- GitHub repository: `alx-backend-javascript`
- Directory: `0x02-ES6_classes`
- File: `2-hbtn_course.js`



☒ Done!

Help

Check your code

 Get a sandbox

QA Review

3. Methods, static methods, computed methods names..... MONEY

mandatory

Score: 100.0% (Checks completed: 100.0%)

Implement a class named `Currency` :

- - Constructor attributes:
 - `code` (String)
 - `name` (String)
- Each attribute must be stored in an "underscore" attribute version (ex: `name` is stored in `_name`)
- Implement a getter and setter for each attribute.
- Implement a method named `displayFullCurrency` that will return the attributes in the following format `name (code)` .

```
bob@dylan:~$ cat 3-main.js
import Currency from './3-currency.js';

const dollar = new Currency('$', 'Dollars');
console.log(dollar.displayFullCurrency());

bob@dylan:~$
bob@dylan:~$ npm run dev 3-main.js
Dollars ($)
bob@dylan:~$
```


Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x02-ES6_classes`
- File: `3-currency.js`

☒ Done!

Help

Check your code

 Get a sandbox

QA Review

4. Pricing

mandatory

Score: 100.0% (Checks completed: 100.0%)



Import the class `Currency` from `3-currency.js`

Implement a class named `Pricing` :

- Constructor attributes:
 - amount (Number)
 - currency (Currency)
- Each attribute must be stored in an “underscore” attribute version (ex: name is stored in `_name`)
- Implement a getter and setter for each attribute.
- Implement a method named `displayFullPrice` that returns the attributes in the following format `amount currency_name (currency_code)` .
- Implement a static method named `convertPrice` . It should accept two arguments: `amount` (Number), `conversionRate` (Number). The function should return the amount multiplied by the conversion rate.

```
bob@dylan:~$ cat 4-main.js
import Pricing from './4-pricing.js';
import Currency from './3-currency.js';

const p = new Pricing(100, new Currency("EUR", "Euro"))
console.log(p);
console.log(p.displayFullPrice());

bob@dylan:~$
bob@dylan:~$ npm run dev 4-main.js
Pricing {
  _amount: 100,
  _currency: Currency { _code: 'EUR', _name: 'Euro' }
}
100 Euro (EUR)
bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x02-ES6_classes`
- File: `4-pricing.js`

☒ Done![Help](#)[Check your code](#)[> Get a sandbox](#)[QA Review](#)

5. A Building

mandatory

Score: 33.33% (Checks completed: 33.33%)

Implement a class named `Building` :

- Constructor attributes:
 - sqft (Number)
- Each attribute must be stored in an “underscore” attribute version (ex: name is stored in `_name`)
- Implement a getter for each attribute.



- Consider this class as an abstract class. And make sure that any class that extends from it should (/) implement a method named `evacuationWarningMessage` .
 - If a class that extends from it does not have a `evacuationWarningMessage` method, throw an error with the message `Class extending Building must override evacuationWarningMessage`

```
bob@dylan:~$ cat 5-main.js
import Building from './5-building.js';

const b = new Building(100);
console.log(b);

class TestBuilding extends Building {}

try {
  new TestBuilding(200)
}
catch(err) {
  console.log(err);
}

bob@dylan:~$
bob@dylan:~$ npm run dev 5-main.js
Building { _sqft: 100 }
Error: Class extending Building must override evacuationWarningMessage
...
bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x02-ES6_classes`
- File: `5-building.js`

☐ Done?

Help

Check your code

Ask for a new correction

> Get a sandbox

QA Review

6. Inheritance

mandatory

Score: 100.0% (Checks completed: 100.0%)

Import `Building` from `5-building.js` .Implement a class named `SkyHighBuilding` that extends from `Building` :

- Constructor attributes:
 - `sqft` (Number) (must be assigned to the parent class `Building`)
 - `floors` (Number)



- Each attribute must be stored in an “underscore” attribute version (ex: name is stored in `_name`)
- (/). Implement a getter for each attribute.
- Override the method named `evacuationWarningMessage` and return the following string `Evacuate slowly the NUMBER_OF_FLOORS floors.`

```
bob@dylan:~$ cat 6-main.js
import SkyHighBuilding from './6-sky_high.js';

const building = new SkyHighBuilding(140, 60);
console.log(building.sqft);
console.log(building.floors);
console.log(building.evacuationWarningMessage());

bob@dylan:~$
bob@dylan:~$ npm run dev 6-main.js
140
60
Evacuate slowly the 60 floors
bob@dylan:~$
```

Repo:

- GitHub repository: `alx-backend-javascript`
- Directory: `0x02-ES6_classes`
- File: `6-sky_high.js`

☒ Done!

Help

Check your code

>_ Get a sandbox

QA Review

7. Airport

mandatory

Score: 100.0% (Checks completed: 100.0%)

Implement a class named `Airport` :

- Constructor attributes:
 - `name` (String)
 - `code` (String)
- Each attribute must be stored in an “underscore” attribute version (ex: name is stored in `_name`)
- The default string description of the class should return the airport `code` (example below).



```
bob@dylan:~$ cat 7-main.js
import Airport from "../7-airport.js";

const airportSF = new Airport('San Francisco Airport', 'SF0');
console.log(airportSF);
console.log(airportSF.toString());

bob@dylan:~$
bob@dylan:~$ npm run dev 7-main.js
Airport [SF0] { _name: 'San Francisco Airport', _code: 'SF0' }
[object SF0]
bob@dylan:~$
```


Repo:

- GitHub repository: alx-backend-javascript
- Directory: 0x02-ES6_classes
- File: 7-airport.js

☒ Done!

Help

Check your code

 Get a sandbox

QA Review

8. Primitive - Holberton Class

mandatory

Score: 100.0% (Checks completed: 100.0%)

Implement a class named HolbertonClass :

- Constructor attributes:
 - size (Number)
 - location (String)
- Each attribute must be stored in an "underscore" attribute version (ex: name is stored in _name)
- When the class is cast into a Number , it should return the size.
- When the class is cast into a String , it should return the location.

```
bob@dylan:~$ cat 8-main.js
import HolbertonClass from "../8-hbtn_class.js";

const hc = new HolbertonClass(12, "Mezzanine")
console.log(Number(hc));
console.log(String(hc));

bob@dylan:~$
bob@dylan:~$ npm run dev 8-main.js
12
Mezzanine
bob@dylan:~$
```



(/)
Repo:

- GitHub repository: alx-backend-javascript
- Directory: 0x02-ES6_classes
- File: 8-hbtn_class.js

☒ Done![Help](#)[Check your code](#)[>_ Get a sandbox](#)[QA Review](#)

9. Hoisting

mandatory

Score: 25.0% (*Checks completed: 25.0%*)

Fix this code and make it work.



```
const class2019 = new HolbertonClass(2019, 'San Francisco');
const class2020 = new HolbertonClass(2020, 'San Francisco');

export class HolbertonClass {
  constructor(year, location) {
    this._year = year;
    this._location = location;
  }

  get year() {
    return this._year;
  }

  get location() {
    return this._location;
  }
}

const student1 = new StudentHolberton('Guillaume', 'Salva', class2020);
const student2 = new StudentHolberton('John', 'Doe', class2020);
const student3 = new StudentHolberton('Albert', 'Clinton', class2019);
const student4 = new StudentHolberton('Donald', 'Bush', class2019);
const student5 = new StudentHolberton('Jason', 'Sandler', class2019);

export class StudentHolberton {
  constructor(firstName, lastName) {
    this._firstName = firstName;
    this._lastName = lastName;
    this._holbertonClass = holbertonClass;
  }

  get fullName() {
    return `${this._firstName} ${this._lastName}`;
  }

  get holbertonClass() {
    return this.holbertonClass;
  }

  get fullStudentDescription() {
    return `${self._firstName} ${self._lastName} - ${self._holbertonClass.year} - ${self._holbertonClass.location}`;
  }
}

export const listOfStudents = [student1, student2, student3, student4, student5];
```

Result:

```
bob@dylan:~$ cat 9-main.js
import listOfStudents from './9-hoisting.js';

console.log(listOfStudents);

const listPrinted = listOfStudents.map(
  student => student.fullStudentDescription
);

console.log(listPrinted)

bob@dylan:~$
bob@dylan:~$ npm run dev 9-main.js
[
  StudentHolberton {
    _firstName: 'Guillaume',
    _lastName: 'Salva',
    _holbertonClass: HolbertonClass { _year: 2020, _location: 'San Francisco' }
  },
  StudentHolberton {
    _firstName: 'John',
    _lastName: 'Doe',
    _holbertonClass: HolbertonClass { _year: 2020, _location: 'San Francisco' }
  },
  StudentHolberton {
    _firstName: 'Albert',
    _lastName: 'Clinton',
    _holbertonClass: HolbertonClass { _year: 2019, _location: 'San Francisco' }
  },
  StudentHolberton {
    _firstName: 'Donald',
    _lastName: 'Bush',
    _holbertonClass: HolbertonClass { _year: 2019, _location: 'San Francisco' }
  },
  StudentHolberton {
    _firstName: 'Jason',
    _lastName: 'Sandler',
    _holbertonClass: HolbertonClass { _year: 2019, _location: 'San Francisco' }
  }
]
[
  'Guillaume Salva - 2020 - San Francisco',
  'John Doe - 2020 - San Francisco',
  'Albert Clinton - 2019 - San Francisco',
  'Donald Bush - 2019 - San Francisco',
  'Jason Sandler - 2019 - San Francisco'
]
bob@dylan:~$
```



- GitHub repository: alx-backend-javascript
- (/). Directory: 0x02-ES6_classes
- File: 9-hoisting.js

☐ Done?

Help

Check your code

Ask for a new correction

> Get a sandbox

QA Review

10. Vroom

mandatory

Score: 100.0% (Checks completed: 100.0%)

Implement a class named `Car` :

- Constructor attributes:
 - `brand` (String)
 - `motor` (String)
 - `color` (String)
- Each attribute must be stored in an "underscore" attribute version (ex: `name` is stored in `_name`)
- Add a method named `cloneCar` . This method should return a new object of the class.

Hint: Symbols in ES6

```
bob@dylan:~$ cat 10-main.js
import Car from "./10-car.js";

class TestCar extends Car {}

const tc1 = new TestCar("Nissan", "Turbo", "Pink");
const tc2 = tc1.cloneCar();

console.log(tc1);
console.log(tc1 instanceof TestCar);

console.log(tc2);
console.log(tc2 instanceof TestCar);

console.log(tc1 == tc2);

bob@dylan:~$
bob@dylan:~$ npm run dev 10-main.js
TestCar { _brand: 'Nissan', _motor: 'Turbo', _color: 'Pink' }
true
TestCar { _brand: undefined, _motor: undefined, _color: undefined }
true
false
bob@dylan:~$
```

**Repo:**

- GitHub repository: alx-backend-javascript
- (/). Directory: 0x02-ES6_classes
- File: 10-car.js

☒ Done!

Help

Check your code

>_ Get a sandbox

QA Review

11. EVCar

#advanced

Score: 100.0% (Checks completed: 100.0%)

Import `car` from `10-car.js`.

Implement a class named `EVCar` that extends the `car` class:

- Constructor attributes:
 - `brand` (String)
 - `motor` (String)
 - `color` (String)
 - `range` (String)
- Each attribute must be stored in an "underscore" attribute version (ex: `name` is stored in `_name`)
- For privacy reasons, when `cloneCar` is called on a `EVCar` object, the object returned should be an instance of `car` instead of `EVCar`.

```
bob@dylan:~$ cat 100-main.js
import EVCar from './100-evcar.js';

const ec1 = new EVCar("Tesla", "Turbo", "Red", "250");
console.log(ec1);

const ec2 = ec1.cloneCar();
console.log(ec2);

bob@dylan:~$
bob@dylan:~$ npm run dev 100-main.js
EVCar {
  _brand: 'Tesla',
  _motor: 'Turbo',
  _color: 'Red',
  _range: '250'
}
Car { _brand: undefined, _motor: undefined, _color: undefined }
bob@dylan:~$
```



Repo:

- GitHub repository: alx-backend-javascript
- Directory: 0x02-ES6_classes

• File: 100-evcar.js
(/)

☒ Done!

Help

Check your code

 Get a sandbox

QA Review

Copyright © 2024 ALX, All rights reserved.

