

# 0x04. Typescript

JavaScript Plain JavaScript TypeScript Planning/me



Weight: 1

Projects(/projects/current)



Project over - took place from Jul 3, 2024 6:00 AM to Jul 4, 2024 6:00 AM



Manual QA Reviews I can make corrections to review



Evaluation quizzes(/dashboards/my\_current\_evaluation\_quizzes)

In a nutshell...

- **Manual QA review:** 43.55/67 mandatory

- **Altogether:** 65.0%



Curriculums(/dashboards/my\_curriculums)

◦ Mandatory: 65.0%

◦ Optional: no optional tasks



Overall Comment(/concepts)

Great Work! Just try finish up the last remaining tasks :)



Conference rooms(/dashboards/video\_rooms)



Servers(/servers)



Sandboxes(/user\_containers/current)



Tools(/dashboards/my\_tools)



Video on demand(/dashboards/videos)



Peers(/users/peers)

Discord(<https://discord.com/app>)

TypeScript

It's just so much better.

My Profile(/users/my\_profile)



# Resources

Read or watch:

- TypeScript in 5 minutes (/rltoken/waTSa9Mguj912pel9On57w)
- TypeScript documentation (/rltoken/iPO8DIHCGzc1jnojLoP9HA)

## Learning Objectives

My Planning(/planning/me)

At the end of this project, you are expected to be able to explain to anyone (/rltoken/PM-5MDItTTOM8Aaa2QIEyQ), **without the help of Google:**

Projects(/projects/current)

- Basic types in Typescript
- Interfaces, Classes, and functions
- How to work with the DOM and Typescript
- Generic types

How to use namespaces

- Evaluating quizzes(/dashboards/my\_current\_evaluation\_quizzes)
- How to use an ambient Namespace to import an external library
- Basic nominal typing with Typescript

## Requirements

Curriculum(/dashboards/my\_curriculums)

- Allowed editors: `vi`, `vim`, `emacs`, Visual Studio Code
- All your files should end with a new line
- All your files will be transpiled on Ubuntu 18.04
- Your TS scripts will be checked with `jest` (version 24.9.\*)
- A `README.md` file, at the root of the folder of the project, is mandatory
- Your code should use the `ts` extension when possible
- The Typescript compiler should not show any warning or error when compiling your code

## Configuration Files

Sandboxes(/user\_containers/current)

Please use these files for the following tasks

Tools(/dashboards/my\_tools)  
**package.json**

Click to show/hide file contents

Video on demand(/dashboards/videos)

**.eslintrc.js**

Click to show/hide file contents

Peers(/users/peers)

**tsconfig.json**

Discord(/discord)  
Click to show/hide file contents

**webpack.config.js**

Click to show/hide file contents

My Profile(/users/my\_profile)



# Tasks

## 0. Creating an interface for a student

mandatory



Home(/)

Score: 65.0% (Checks completed: 100.0%)



Copy the following configuration files (provided above) into the `task_0` directory: `package.json`, `.eslintrc.js`, `tsconfig.json`, `webpack.config.js`



Write your code in the `main.ts` file:

Projects(/projects/current)

- Write an interface named `Student` that accepts the following elements: `firstName(string)`, `lastName(string)`, `age(number)`, and `location(string)`
- Create two students, and create an array named `studentsList` containing the two variables
- Using Vanilla Javascript, render a table and for each elements in the array, append a new row to the table
- Each row should contain the first name of the student and the location

Requirements:



- When running, Webpack should return `No type errors found`.
- Every variable should use TypeScript when possible.



Concepts(/concepts)

Repo:



- GitHub repository: `alx-backend-javascript`
- Directory: `0x04-TypeScript`
- File: `task_0/js/main.ts`, `task_0/package.json`, `task_0/.eslintrc.js`, `task_0/tsconfig.json`, `task_0/webpack.config.js`



Servers(/servers)



> Get a sandbox

View results

Sandboxes(/user\_containers/current)



Tools(/dashboards/my\_tools)

## 1. Let's build a Teacher interface

mandatory



Video on demand(/dashboards/videos)

Score: 65.0% (Checks completed: 100.0%)



Create a directory `task_1` and copy these configuration files into this folder: `package.json`,

Peers(/users/peers)

`tsconfig.json`, `webpack.config.js`



• `firstName(string)` and `lastName(string)`. These two attributes should only be modifiable when a Teacher is first initialized

Discord(<https://discord.com/app>)

- `fullTimeEmployee(boolean)` this attribute should always be defined
- `yearsOfExperience(number)` this attribute is optional
- `location(string)` this attribute should always be defined
- Add the possibility to add any attribute to the Object like `contract(boolean)` without specifying the type

My Profile(/users/my\_profile)



Example:



```
const teacher3: Teacher = {
  firstName: 'John',
  fullTimeEmployee: false,
  lastName: 'Doe',
  location: 'London',
  contract: false,
};

// My Planning(/planning/me)

console.log(teacher3);

// Projects(/projects/current)
// should print
// Object
// contract: false
// firstName: "John"
// fullTimeEmployee: false
// lastName: "Doe"
// location: "London"
// QA Reviews I can make(/corrections/to_review)
// Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)
```



**Repo:** [Curriculums\(/dashboards/my\\_curriculums\)](#)



- GitHub repository: alx-backend-javascript
- Directory: 0x04-TypeScript
- File: task\_1/js/main.ts, task\_1/webpack.config.js, task\_1/tsconfig.json, task\_1/package.json



[Conference rooms\(/dashboards/video\\_rooms\)](#)

[View results](#)



[Servers\(/servers\)](#)

## 2. Extending the Teacher class

mandatory



[Sandboxes\(/user\\_containers/current\)](#)

Score: 65.0% (Checks completed: 100.0%)



Write an interface named `Director` that extends `Teacher`. It requires an attribute named `numberOfReports(number)`



[Example video on demand\(/dashboards/videos\)](#)



[Peers\(/users/peers\)](#)



[Discord\(https://discord.com/app\)](https://discord.com/app)



[My Profile\(/users/my\\_profile\)](#)



```
const director1: Directors = {
  firstName: 'John',
  lastName: 'Doe',
  location: 'London',
  fullTimeEmployee: true,
  numberOfReports: 17,
};
console.log(director1);
// should print
// Object
// {firstName: 'John', lastName: 'Doe', location: 'London', fullTimeEmployee: true, numberOfReports: 17}
// fullTimeEmployee: true
// lastName: "Doe"
// location: "London"
// numberOfReports: 17
```



Evaluation quizzes(/dashboards/my\_current\_evaluation\_quizzes)

### Repo:



- GitHub repository: alx-backend-javascript
- Directory: 0x04-TypeScript
- File: task\_1/js/main.ts



Concepts(/concepts)

[View results](#)



Conference rooms(/dashboards/video\_rooms)

mandatory



Score: 100.0% (Checks completed: 100.0%)



Write a function `printTeacher` :

Sandboxes(/user\_containers/current)

- It accepts two arguments `firstName` and `lastName`
- It returns the first letter of the `firstName` and the full `lastName`
- Example: `printTeacher("John", "Doe") -> J. Doe`



Tools(/dashboards/my\_tools)

Write an interface for the function named `printTeacherFunction` .



Video on demand(/dashboards/videos)

### Repo:



- GitHub repository: alx-backend-javascript
- Directory: 0x04-TypeScript
- File: task\_1/js/main.ts



Discord(<https://discord.com/app>)

[View results](#)



My Profile(/users/my\_profile)

## 4. Writing a class

mandatory

(/)

Score: 65.0% (Checks completed: 100.0%)



Home(/)

Write a Class named StudentClass :



My Planning(/planning/me)

- The constructor accepts firstName(string) and lastName(string) arguments
- The class has a method named workOnHomework that return the string Currently working
- The class has a method named displayName . It returns the firstName of the student
- The constructor of the class should be described through an Interface
- The class should be described through an Interface



Projects(/projects/current)

Requirements:



QA Reviews I can make(/corrections/to\_review)

- You can reuse the Webpack configuration from the previous exercise to compile the code.
- When running npm run build ,no TypeScript error should be displayed.



• Evaluation quizzes I can make(/evaluation/quizzes)

Repo:



Public repos(/dashboards/my\_repos)

• Directory: 0x04-TypeScript



• File: task\_1/js/main.ts  
Concepts(/concepts)

View results



Conference rooms(/dashboards/video\_rooms)

## 5. Advanced types Part 1

mandatory



Servers(/servers)

Score: 65.0% (Checks completed: 100.0%)



Sandboxes(/user\_containers/current)

Create the DirectorInterface interface with the 3 expected methods:



- workFromHome() returning a string
- Tools(/dashboards/my\_tools)
- getCoffeeBreak() returning a string
- workDirectorTasks() returning a string



Video on demand(/dashboards/videos)

Create the TeacherInterface interface with the 3 expected methods:

- workFromHome() returning a string
- getCoffeeBreak() returning a string
- workDirectorTasks() returning a string



Users(/users/peers)

Create a class Director that will implement DirectorInterface



Discord(https://discord.com/app)

- workFromHome should return the string Working from home
- getToWork should return the string Getting a coffee break
- workDirectorTasks should return the string Getting to director tasks

Create a class Teacher that will implement TeacherInterface

- My Profile(/users/my\_profile)
- workFromHome should return the string Cannot work from home



- getCoffeeBreak should return the string Cannot have a break
- workTeacherTasks should return the string Getting to work

Create a function createEmployee with the following requirements:

- It can return either a Director or a Teacher instance
- It accepts 1 arguments:

- salary (either number or string)

- if salary is a number and less than 500 - It should return a new Teacher . Otherwise it should return a Director

Expected result:

```
console.log(createEmployee(200));
Teacher
QA Reviews I can make(/corrections/to_review)
console.log(createEmployee(1000));
Director
console.log(createEmployee('$500'));
Evaluation quizzes(/dashboards/my_current_evaluation_quizzes)
Director
```

**Repo:** Curriculums(/dashboards/my\_curriculums)

- GitHub repository: alx-backend-javascript
- Directory: 0x04-TypeScript
- File: task\_2/js/main.ts, task\_2/webpack.config.js, task\_2/tsconfig.json, task\_2/package.json

Conference rooms(/dashboards/video\_rooms)

View results

Servers(/servers)

## 6. Creating functions specific to employees

mandatory

Sandboxes(/user\_containers/current)

Score: 65.0% (Checks completed: 100.0%)

Tools(/dashboards/my\_tools)

- it accepts employee as an argument
- it will be used as a predicate and if the employee is a director

Write a function executeWork :

- it accepts employee as an argument
- if the employee is a Director, it will call workDirectorTasks
- if the employee is a Teacher, it will call workTeacherTasks

Expected result:

```
executeWork(createEmployee(200));
Getting to work
executeWork(createEmployee(1000));
Getting to director tasks
```

My Profile(/users/my\_profile)

## Repo:



- GitHub repository: alx-backend-javascript
- Directory: 0x04-TypeScript
- File: task\_2/js/main.ts



Home(/)



View results

My Planning(/planning/me)

## 7. String literal types

mandatory



Projects(/projects/current)

Score: 65.0% (Checks completed: 100.0%)



QA Reviews I can make(/corrections/to\_review)

Write a String literal type named `Subjects` allowing a variable to have the value `Math` or `History` only.

Write a function named `teachClass` :



Evaluation quizzes(/dashboards/my\_current\_evaluation\_quizzes)

- it takes `todayClass` as an argument
- it will return the string `Teaching Math` if `todayClass` is `Math`
- it will return the string `Teaching History` if `todayClass` is `History`



Expected result: Curriculums(/dashboards/my\_curriculums)



```
teachClass('Math');  
Teaching Math
```

```
teachClass('History');
```

```
Teaching History
```



Conference rooms(/dashboards/video\_rooms)



Servers(/servers)

- GitHub repository: alx-backend-javascript
- Directory: 0x04-TypeScript
- Sandboxes(/user\_containers/current)
- File: task\_2/js/main.ts



Sandboxes(/user\_containers/current)



Tools(/dashboards/my\_tools)

Get a sandbox

View results



Video on demand(/dashboards/videos)

## 8. Ambient Namespaces

mandatory



Peers(/users/peers)

Score: 65.0% (Checks completed: 100.0%)



Discord(https://discord.com/app)

Create a directory called `task_3` and copy these configuration files into it: `package.json`, `webpack.config.js`, `tsconfig.json`.

The first part of will require that you build an `interface` and a `type`. Inside a file named `interface.ts` :

- Create a type `(user: User) => number` equal to `number`





- Create an interface `RowElement` that contains these 3 fields:

```

    ◦ firstName: string
    ◦ lastName: string
    ◦ age?: number

```



You are building the next part of the application architecture. The goal is to save the entities to a database. Because of time constraints, you can't write a connector to the database, and you decided to download a library called `crud.js`. Copy this file into the `task_3/js` directory.



My Planning(/planning/me)

Here it is



Projects(/projects/current)



QA Reviews I can make(/corrections/to\_review)



Evaluation quizzes(/dashboards/my\_current\_evaluation\_quizzes)



Curriculums(/dashboards/my\_curriculums)



Concepts(/concepts)



Write an ambient file within `task_3/js`, named `crud.d.ts` containing the type declarations for each `crud` function. At the top of the file import `RowID` and `RowElement` from `interface.ts`.



In `main.ts`

- Servers(/servers)
- At the top of the file create a triple slash directive (`///<div data-bbox="69 600 104 621" data-label="Image"><img alt="Right arrow icon" data-bbox="69 600 104 621" data-kind="parent" data-rs="2" style="vertical-align: middle;"/></div><div data-bbox="147 600 629 617" data-label="Text"><code>@ts-ignore</code> // Use type <code>RowElement</code> from <code>interface.ts</code></div><div data-bbox="110 620 472 635" data-label="List-Group"><ul><li>• Import everything from <code>crud.js</code> as <code>CRUD</code></li></ul></div></div><div data-bbox="69 648 104 669" data-label="Image"><img alt="Wrench icon" data-bbox="69 648 104 669" data-kind="parent" data-rs="2" style="vertical-align: middle;"/></div><div data-bbox="147 648 803 666" data-label="Text"><p>Create an object called <code>row</code> with the type <code>RowElement</code> with the fields set to these values:</p></div><div data-bbox="110 677 463 717" data-label="List-Group"><ul><li>• <code>firstName</code>: Guillaume</li><li>• <code>lastName</code>: Salva</li></ul></div><div data-bbox="69 700 104 721" data-label="Image"><img alt="Video icon" data-bbox="69 700 104 721" data-kind="parent" data-rs="2" style="vertical-align: middle;"/></div><div data-bbox="147 700 463 717" data-label="Text"><p>Video on demand(/dashboards/videos)</p></div><div data-bbox="82 725 900 758" data-label="Text"><p>Create a <code>const</code> variable named <code>newRowID</code> with the type <code>RowID</code> and assign the value the <code>insertRow</code> command.</p></div><div data-bbox="69 772 104 793" data-label="Image"><img alt="Group of people icon" data-bbox="69 772 104 793" data-kind="parent" data-rs="2" style="vertical-align: middle;"/></div><div data-bbox="147 772 915 805" data-label="Text"><p>Next, create a <code>const</code> variable named <code>updatedRow</code> with the type <code>RowElement</code> and update <code>row</code> with an <code>age</code> field set to 23</p></div><div data-bbox="69 825 104 846" data-label="Image"><img alt="Discord icon" data-bbox="69 825 104 846" data-kind="parent" data-rs="2" style="vertical-align: middle;"/></div><div data-bbox="82 818 532 843" data-label="Text"><p>Finally, call the <code>updateRow</code> and <code>deleteRow</code> commands.</p></div><div data-bbox="82 847 216 862" data-label="Text"><p>Expected result:</p></div><div data-bbox="882 849 928 879" data-label="Image"><img alt="Search icon" data-bbox="882 849 928 879" data-kind="parent" data-rs="2" style="vertical-align: middle;"/></div><div data-bbox="147 926 381 943" data-label="Text"><p>My Profile(/users/my_profile)</p></div>`


My Profile(/users/my\_profile)

```
const obj = {firstName: "Guillaume", lastName: "Salva"};
CRUD.insertRow(obj)
//Insert row {firstName: "Guillaume", lastName: "Salva"}
```

```
const updatedRow: RowElement = { firstName: "Guillaume", lastName: "Salva", age: 23 }; Home()
CRUD.updateRow(newRowID, updatedRow);
// Update row 125 {firstName: "Guillaume", lastName: "Salva", age: 23}
My Planning(/planning/me)
CRUD.deleteRow(125);
// Delete row id 125
Projects(/projects/current)
```

Requirements:

- ✓ When running I can run without TypeScript error should be displayed.
- Every variable should use TypeScript when possible.
- The main file and the ambient file should both import the types defined in the interface file.
- ? You can easily test your ambient file by looking at the intelligence of your IDE when using the 3rd party functions.

 **Repo:** [Curriculums\(/dashboards/my\\_curriculums\)](#)

- GitHub repository: [alx-backend-javascript](#)
- Directory: [0x04-TypeScript](#)
- File: [task\\_3/js/main.ts](#), [task\\_3/js/interface.ts](#), [task\\_3/js/crud.d.ts](#)


 [Conference rooms\(/dashboards/video\\_rooms\)](#)

[View results](#)


## **Workspace & Declaration merging**

mandatory

 **Score:** 65.08% (*Checks completed: 100.00%*)

 Create a new directory `task_4` and copy the above `tsconfig.json` and put this `package.json` in there [Tools\(/dashboards/my\\_tools\)](#)

 [Video on demand\(/dashboards/videos\)](#)

 [Peers\(/users/peers\)](#)

 [Discord\(https://discord.com/app\)](https://discord.com/app)



[My Profile\(/users/my\\_profile\)](#)

```

{
  "name": "task_4",
  ("version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "build": "webpack",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "MIT",
  "devDependencies": {
    "@typescript-eslint/eslint-plugin": "^2.4.0",
    "@types/node": "^12.0.0",
    "clean-webpack-plugin": "^3.0.0",
    "fork-ts-checker-webpack-plugin": "^1.5.1",
    "webpack-cli": "^3.3.0",
    "webpack-dev-server": "^3.8.2"
  }
}

```

In task\_4/js/subjects:

- Create a file `teacher.ts` and write a `Teacher` interface in a namespace named `Subjects`.
  - the interface requires `firstName` and `lastName` as string
- Create a file `subject.ts` and write a `Subject` class in the same namespace named `Subjects`.
  - the class has one attribute `teacher` that implements the `Teacher` interface
  - the class has one setter method `setTeacher` that accepts a `teacher` in argument (and as setter, set the instance attribute `teacher` with it)
- Create a file `Cpp.ts` and make the following modifications in the same namespace.
  - Using declaration merging, add a new optional attribute `experienceTeachingC` (number) to the `Teacher` interface
  - Create a class `Cpp` extending from `Subject`
    - Write a method named `getRequirements` that will return a string Here is the list of requirements for Cpp
    - Write a method named `getAvailableTeacher` that will return a string Available Teacher: <first name of teacher>
    - If the teacher doesn't have any experience in teaching C, then the method should return a string No available teacher
- Create a file `React.ts` and write a `React` Class in the same namespace.
  - Add a new attribute `experienceTeachingReact?` (number) to the `Teacher` interface
  - In the class, write a method named `getRequirements` that will return a string Here is the list of requirements for React
  - Write a method named `getAvailableTeacher` that will return a string Available Teacher: <first name of teacher>



- If the teacher doesn't have any experience in teaching React, then the method should return a string `No available teacher`

7) Create a file `Java.ts` and write a `Java` Class in the same namespace.



◦ Add a new attribute `experienceTeachingJava? (number)` to the `Teacher` interface  
Home(/) In the class, write a method named `getRequirements` that will return a string Here is the list of requirements for Java



◦ Write a method named `getAvailableTeacher` that will return a string `Available`  
My Planning(/planning/me) `Teacher: <first name of teacher>`



◦ If the teacher doesn't have any experience in teaching Java, then the method should return a string `No available teacher`  
Projects(/projects/current)



Repo: QA Reviews I can make(/corrections/to\_review)

- GitHub repository: `alx-backend-javascript`
- Directory: `0x04-TypeScript`
- File: `task_4/package.json`, `task_4/tsconfig.json`, `task_4/js/subjects/Cpp.ts`, `task_4/js/subjects/Java.ts`, `task_4/js/subjects/React.ts`, `task_4/js/subjects/Subject.ts`, `task_4/js/subjects/Teacher.ts`



Curriculums(/dashboards/my\_curriculums)

View results



Concepts(/concepts)

## 10. Update task\_4/js/main.ts

mandatory



Conference rooms(/dashboards/video\_rooms)

Score: 65.0% (Checks completed: 100.0%)



- create and export a constant `cpp` for Cpp Subjects
- create and export a constant `java` for Java Subjects
- create and export a constant `react` for React Subjects



Sandboxes(/user\_containers/current)

- create and export one teacher object `cTeacher` with `experienceTeachingC = 10`
- for Cpp subject, log to the console `C++`, set `cTeacher` as the teacher, call the two methods `getRequirements` and `getAvailableTeacher` and print the strings they return



Tools(/dashboards/my\_tools)

- for Java subject, log to the console `Java`, set `cTeacher` as the teacher, call the two methods `getRequirements` and `getAvailableTeacher`, and print the strings they return



Video on demand(/dashboards/videos)

- for React subject, log to the console `React`, set `cTeacher` as the teacher, call the two methods `getRequirements` and `getAvailableTeacher`, and print the strings they return



Repo: Peers(/users/peers)

- GitHub repository: `alx-backend-javascript`



- Directory: `0x04-TypeScript`
- Discord(<https://discord.com/app>)
- File: `task_4/js/main.ts`



View results

My Profile(/users/my\_profile)

## 11. Brand convention & Nominal typing

mandatory

(/)

Score: 65.0% (Checks completed: 100.0%)



Home(/)

Create a directory `task_5` and copy these configuration files into the root of `task_5`: `package.json`, `tsconfig.json`, `webpack.config.js`



My Planning(/planning/me)

Create two interfaces `MajorCredits` and `MinorCredits` in `task_5/js/main.ts`:

- Each interface defines a number named `credits`
- Add a brand property to each interface in order to uniquely identify each of them



Projects(/projects/current)

Create two functions named `sumMajorCredits` and `sumMinorCredits` in `task_5/js/main.ts`:



QA Reviews I can make(/corrections/to\_review)

- Each function takes two arguments `subject1` and `subject2`
- `sumMajorCredits` returns `MajorCredits` value and `sumMinorCredits` returns `MinorCredits` value



Evaluation quizzes(/dashboards/my\_current\_evaluation\_quizzes)

- Each function sums the credits of the two subjects

Repo:



Curriculums(/dashboards/my\_curriculums)

- GitHub repository: `alx-backend-javascript`
- Directory: `0x04-TypeScript`



Files(/dashboards/my\_files)  
• `task_5/js/main.ts`, `task_5/package.json`, `task_5/webpack.config.js`, `task_5/tsconfig.json`



Conference rooms(/dashboards/video\_rooms)

View results



Servers(/servers)



Sandboxes(/user\_containers/current)

Ready for a new review



Tools(/dashboards/my\_tools)



Video on demand(/dashboards/videos)

Copyright © 2024 ALX, All rights reserved.



Peers(/users/peers)



Discord(<https://discord.com/app>)



My Profile(/users/my\_profile)