

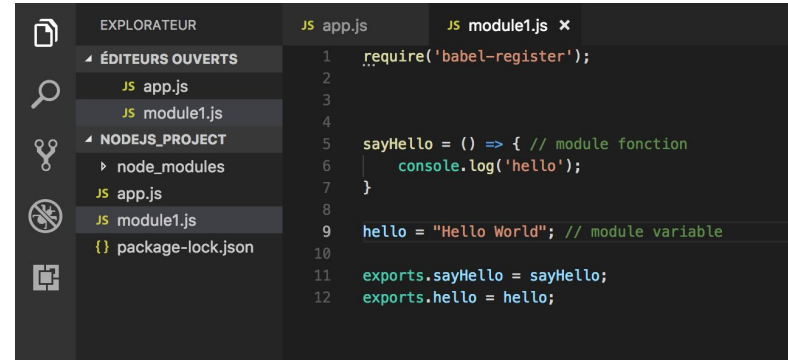
1er façon d'exporter un module



The image shows the VS Code interface. The Explorer on the left lists 'app.js' and 'module1.js' under 'NODEJS_PROJECT'. The Editor shows 'module1.js' with the following code:

```
1 .require('babel-register');
2
3
4
5 exports.sayHello = () => { // module fonction
6     console.log('hello');
7 }
8
9 exports.hello = "Hello World"; // module variable
10
```

2eme façon d'exporter un module "TOP"



The image shows the VS Code interface. The Explorer on the left lists 'app.js' and 'module1.js' under 'NODEJS_PROJECT'. The Editor shows 'module1.js' with the following code:

```
1 .require('babel-register');
2
3
4
5 sayHello = () => { // module fonction
6     console.log('hello');
7 }
8
9 hello = "Hello World"; // module variable
10
11 exports.sayHello = sayHello;
12 exports.hello = hello;
```



The image shows the VS Code interface. The Explorer on the left lists 'app.js' and 'module1.js' under 'NODEJS_PROJECT'. The Editor shows 'app.js' with the following code:

```
1 const mod1 = .require('./module1')
2
3
4 mod1.sayHello(); //appel du module fonction sayHello()
5
6 console.log(mod1.hello); // appel du module variable ()
```

Par convention en mettant notre fichier module.js dans le répertoire node_modules on est plus obligé de préciser le chemin relatif pour l'importer.



The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with the following structure:

- EXPLORATEUR
 - ÉDITEURS OUVERTS
 - JS app.js
 - JS module1.js node_...
 - NODEJS_PROJECT
 - node_modules
 - JS app.js
 - package-lock.json

The main editor area has two tabs: 'JS app.js' and 'JS module1.js'. The 'JS module1.js' tab is active, displaying the following code:

```
1  const mod1 = require('module1')
2
3
4  mod1.sayHello(); //appel du module fonction sayHello()
5
6  console.log(mod1.hello); // appel du module variable ()
```