

coursNodeJs

01 - Fondamentaux

Rappels syntaxe :

<https://code-garage.fr/blog/le-guide-simplifie-sur-la-syntaxe-markdown>

'# Titre 1 '## Titre 2 ... '##### Titre 6

'*italique*

'**gras**

'***gras/italique***

'~~barré~~

Cours / Exercices

Dans le Terminal GitBash

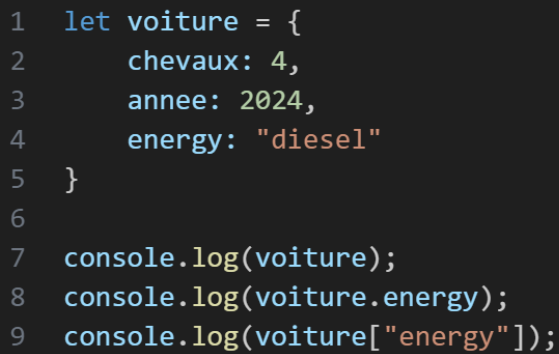
```
57522J@PWVD3139 MINGW64 /c/Workspaces/coursNodeJs $ touch intro.js
```

```
57522J@PWVD3139 MINGW64 /c/Workspaces/coursNodeJs $ node intro hello Bonjour je mappelle Sofiane  
Bonjour je mappelle Thibault
```

```
57522J@PWVD3139 MINGW64 /c/Workspaces/coursNodeJs $ node intro hello Bonjour je mappelle Sofiane  
Bonjour je mappelle Thibault Bonjour je mappelle Jeremy
```

nouveau fichier : vehicule.js

```
57522J@PWVD3139 MINGW64 /c/Workspaces/coursNodeJs $ touch vehicule.js
```

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains the following JavaScript code:

```
1 let voiture = {
2     chevaux: 4,
3     annee: 2024,
4     energy: "diesel"
5 }
6
7 console.log(voiture);
8 console.log(voiture.energy);
9 console.log(voiture["energy"]);
```

```
let voiture = { chevaux: 4, annee: 2024, energy: "diesel" }
```

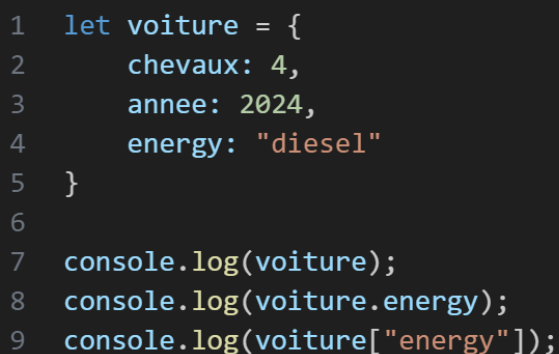
```
console.log(voiture);
```

```
console.log(voiture.energy);
```

```
console.log(voiture["energy"]);
```

Dans le terminal

```
57522J@PWVD3139 MINGW64 /c/Workspaces/coursNodeJs $ node vehicule.js { chevaux: 4, annee: 2024,
energy: 'diesel' } diesel diesel
```

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It contains the following JavaScript code:

```
1 let voiture = {
2     chevaux: 4,
3     annee: 2024,
4     energy: "diesel"
5 }
6
7 console.log(voiture);
8 console.log(voiture.energy);
9 console.log(voiture["energy"]);
```

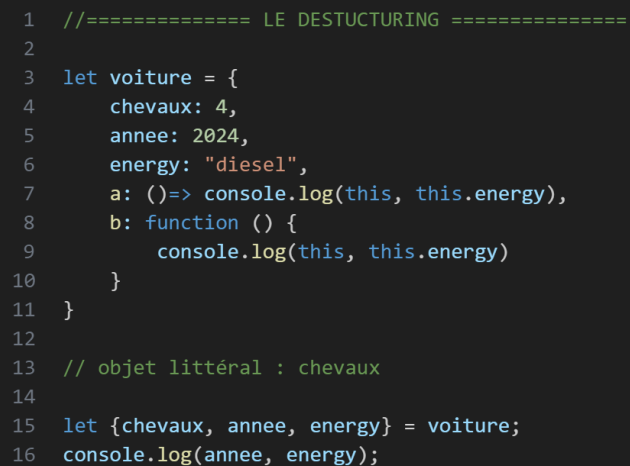
```
let voiture = { chevaux: 4, annee: 2024, energy: "diesel", a: ()=> console.log(this), b: function () {
console.log(this) } }
```

```
voiture.a() voiture.b()
```

Dans le terminal

57522J@PWVD3139 MINGW64 /c/Workspaces/coursNodeJs \$ node vehicule.js {} { chevaux: 4, annee: 2024, energy: 'diesel', a: [Function: a], b: [Function: b] }

==> L'objet global de NODE est accessible de partout !!!!!!!



```
1 //===== LE DESTUCTURING =====
2
3 let voiture = {
4   chevaux: 4,
5   annee: 2024,
6   energy: "diesel",
7   a: ()=> console.log(this, this.energy),
8   b: function () {
9     console.log(this, this.energy)
10  }
11 }
12
13 // objet littéral : chevaux
14
15 let {chevaux, annee, energy} = voiture;
16 console.log(annee, energy);
```

//===== LE DESTUCTURING =====

```
let voiture = { chevaux: 4, annee: 2024, energy: "diesel", a: ()=> console.log(this, this.energy), b: function () {
  console.log(this, this.energy) } }
```


Dans le terminal

PS C:\Workspaces\coursNodeJs> node vehicule 2024 diesel PS C:\Workspaces\coursNodeJs>



```
1  const eleves = [  
2    {  
3      nom : "Omar",  
4      niveau: 1  
5    },  
6    {  
7      nom : "Sandrine",  
8      niveau: 1  
9    },  
10   {  
11     nom : "Olivier",  
12     niveau: 1  
13   },  
14   {  
15     nom : "Marie",  
16     niveau: 1  
17   },  
18   {  
19     nom : "Jérémy",  
20     niveau: 1  
21   }  
22 ]  
23  
24 console.log(eleves);
```

```
//===== require('./formation');
```



```
1  //=====  
2  require('./formation');
```


Dans le terminal

57522J@PWVD3139 MINGW64 /c/Workspaces/coursNodeJs \$ node intro [{ nom: 'Omar', niveau: 1 }, { nom: 'Sandrine', niveau: 1 }, { nom: 'Olivier', niveau: 1 }, { nom: 'Marie', niveau: 1 }, { nom: 'Jérémy', niveau: 1 }] Avec les modules : dans formation.js



```
1 module.exports = eleves
```

dans intro.js

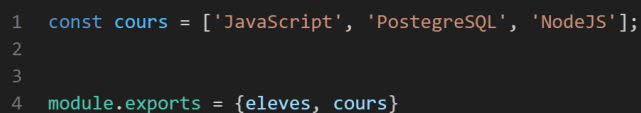


```
1 // ===== bis avec les modules
2 let eleves = require('./formation')
3
4 console.log(eleves);
```

Dans le terminal

57522J@PWVD3139 MINGW64 /c/Workspaces/coursNodeJs \$ node intro [{ nom: 'Omar', niveau: 1 }, { nom: 'Sandrine', niveau: 1 }, { nom: 'Olivier', niveau: 1 }, { nom: 'Marie', niveau: 1 }, { nom: 'Jérémy', niveau: 1 }]

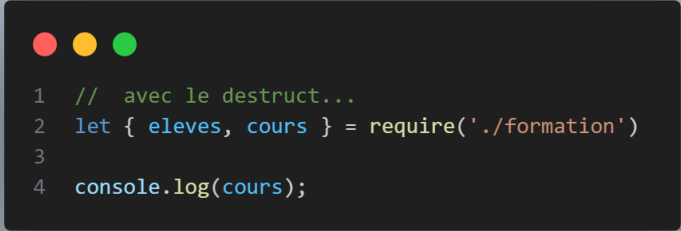
Avec le destructuring on peut exporter/ importer des élts d'un même module



```
1 const cours = ['JavaScript', 'PostgreSQL', 'NodeJS'];
2
3
4 module.exports = {eleves, cours}
```

Dans formation :

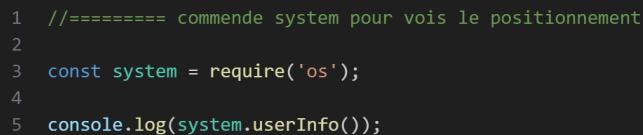
Dans intro :



```
1 // avec le destruct...
2 let { eleves, cours } = require('./formation')
3
4 console.log(cours);
```

Dans le terminal

57522J@PWVD3139 MINGW64 /c/Workspaces/coursNodeJs \$ node intro ['JavaScript', 'PostgreSQL', 'NodeJS']



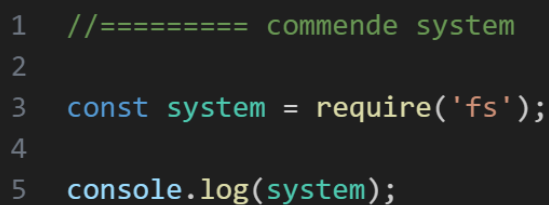
```
1 //===== commande system pour vois le positionnement
2
3 const system = require('os');
4
5 console.log(system.userInfo());
```

Intro : Info System

Dans le terminal

57522J@PWVD3139 MINGW64 /c/Workspaces/coursNodeJs \$ node intro { uid: -1, gid: -1, username: '57522J', homedir: 'C:\Users\57522J', shell: null }

Infos syst 'fs' 'os' --> operating system 'fs' --> file system



```
1 //===== commande system
2
3 const system = require('fs');
4
5 console.log(system);
```

//===== commande system

const system = require('fs'); console.log(system); throw new Error('il y a une nouvelle erreur')

--> gestion d'erreurs avec throw, catch...

Dans le terminal

Constitution d'un url

https://maif.fr/assurance

protocol : https://

domaine : 13.146.246.42

port : 3000

ressource : /assurance => end point (// avec API REST)

crypté (haché) ==> impossible à décrypter

PC --> requete --> Server

PC <-- response <-- Server (response et requete avec head, body,...)

! Rq : Couper l'execution sur le Terminal : ctrl + c



```
const http = require('http');
```

```
const server = http.createServer( (Request, Response) => { console.log("Server created successfully"); } )
```

```
server.listen(8080, "localhost", ()=>{ console.log("Server is listening on port 8080"); })
```

Dans le terminal

```
57522J@PWVD3139 MINGW64 /c/Workspaces/coursNodeJs $ node intro Server is listening on port 8080
```

=> Après avoir ouvert localhost:8080 dans le navigateur :

Dans le terminal

```
57522J@PWVD3139 MINGW64 /c/Workspaces/coursNodeJs $ node intro Server is listening on port 8080
Server created successfully
```

Pour obtenir la methode :



```
1  const http = require('http');
2
3  const server = http.createServer(
4    (Request, Response) => {
5      console.log("Server created successfully");
6      console.log(Request.method);
7    }
8  )
9
10 server.listen(8080, "localhost", ()=>{
11   console.log("Server is listening on port 8080");
12 })
```

```
const http = require('http');
```

```
const server = http.createServer( (Request, Response) => { console.log("Server created successfully");
console.log(Request.method); } )
```

```
server.listen(8080, "localhost", ()=>{ console.log("Server is listening on port 8080"); })
```



```
1  const http = require('http');
2
3  const server = http.createServer(
4    (Request, Response) => {
5      if (Request.url === '/formation') {
6        console.log('Vous êtes en formation chez Simplon');
7      } else if (Request.url === '/entreprise') {
8        console.log('Vous êtes en alternance chez la Maif');
9      }
10   }
11 )
12
13 server.listen(8080, "localhost", ()=>{
14   console.log("Server is listening on port 8080");
15 })
```

```
server.listen(8080, "localhost", ()=>{ console.log("Server is listening on port 8080"); })
```

après avoir saisi : `http://localhost:8080/entreprise` dans le anv

```
57522J@PWVD3139 MINGW64 /c/Workspaces/coursNodeJs $ node intro
Server is listening on port 8080
Vous êtes en alternance chez la Maif
```

Autre ex :


```

1  const http = require('http');
2
3  const server = http.createServer(
4    (Request, Response) => {
5      if ((Request.url === '/' || Request.url === '/home') && Request.method === 'GET') {
6        console.log('Vous êtes sur la page home');
7      } else if ((Request.url === '/formation') && Request.method === 'GET') {
8        console.log('Vous êtes sur la page Simplon');
9      } else if ((Request.url === '/entreprise') && Request.method === 'GET') {
10       console.log('Vous êtes en alternance chez la Maif');
11     }
12   }
13 )
14
15 server.listen(8080, "localhost", ()=>{
16   console.log("Server is listening on port 8080");
17 })

```

let rep = "";

```

const server = http.createServer( (Request, Response) => { if ((Request.url === '/profil') && Request.method === 'GET') { console.log('vous êtes sur la page profil'); rep = true; } else if ((Request.url === '/contact') && Request.method === 'GET') { console.log('vous êtes sur la page contact'); rep = true; } else if (rep !== true) { console.log('404 page non valide'); } // } else { // console.error('404 page non valide'); // } } )

```

```

server.listen(8080, "localhost", ()=>{ console.log("Server is listening on port 8080"); })

```

// ===== Autre version avec Switch

```

switch (req.url) { case '/': case '/home': console.log('Vous êtes sur la home page'); break; case '/formation': console.log('Vous êtes en formation chez Simplon'); break; case '/entreprise': console.log('Vous êtes à la Maif'); break; case '/maison': console.log('Vous êtes en pyjama'); break; default: console.log('404 dans les choux'); break; }

```

```

1  const server = http.createServer(
2    (Request, Response) => {
3
4
5      Response.setHeader("content-type", "text/html");
6      Response.write('<head><meta charset="UTF-8"></head>')
7
8      if ((Request.url === '/' || Request.url === '/home') && Request.method === 'GET') {
9        Response.write('<h1>Vous êtes sur la page home</h1>');
10     } else if ((Request.url === '/formation') && Request.method === 'GET') {
11       Response.write('<h1>Vous êtes sur la page Simplon</h1>');
12     } else if ((Request.url === '/profil') && Request.method === 'GET') {
13       Response.write('<h1>vous êtes sur la page profil</h1>');
14       // rep = true;
15     } else if ((Request.url === '/contact') && Request.method === 'GET') {
16       Response.write('<h1>vous êtes sur la page contact</h1>');
17       // rep = true;
18     } // } else if (rep !== true) {
19     //   Response.write('404 page non valide');
20     // }
21     } else {
22       Response.write('404 page non valide');
23     }
24     Response.end()
25   }
26 )

```

```

const server = http.createServer( (Request, Response) => {

```

```

  Response.setHeader("content-type", "text/html");
  Response.write('<head><meta charset="UTF-8"></head>')

```

```

  if ((Request.url === '/' || Request.url === '/home') && Request.method ===

```

```
'GET') {
    Response.write('<h1>Vous êtes sur la page home</h1>');
} else if ((Request.url === '/formation') && Request.method === 'GET') {
    Response.write('<h1>Vous êtes sur la page Simplon</h1>');
} else if ((Request.url === '/profil') && Request.method === 'GET') {
    Response.write('<h1>vous êtes sur la page profil</h1>');
    // rep = true;
} else if ((Request.url === '/contact') && Request.method === 'GET') {
    Response.write('<h1>vous êtes sur la page contact</h1>');
    // rep = true;
// } else if (rep !== true) {
//     Response.write('404 page non valide');
// }
} else {
    Response.write('404 page non valide');
}
Response.end()
```

)

Installation nodemon

permet de relance le code à chaque modif

PS C:\Workspaces\coursNodeJs> npm install --save-dev nodemon

PS C:\Workspaces\coursNodeJs> npm init -y

==> création d'un dossier node_modules (à ne pas modifier !!!!)

==> création de 2 fichiers json

dans script, indiquer les fichiers à relancer automatiquement à chaque modif : "dev": "nodemon ./server.js"



ex : "start": "nodemon index.js" <== lancement de index.js

Création d'un dossier

Création d'un fichier dans un dossier (à créer)

--> Voir le fichier gestionFichiers.js

```

1 const fs = require('fs');
2
3 // ===== CREATION ET SUPPRESSION DE DOSSIER
4
5 // if (fs.existsSync('./monDossier')) {
6 //   fs.mkdir('./monDossier', (error) => { // mkdir -> création du dossier
7 //     if (error) {
8 //       console.log(error);
9 //     } else {
10 //       console.log('Dossier correctement créé');
11 //     }
12 //   }
13 // } else {
14
15 //   fs.rmdir('./monDossier', (error) => { // rmdir -> suppression du dossier
16 //     if (error) {
17 //       console.log(error);
18 //     } else {
19 //       console.log('Dossier correctement supprimé');
20 //     }
21 //   }
22 // }
23
24 // ===== 2nd méthode
25
26 // fs.access('./monDossier', (err) => { // access -> Tester si le chemin existe // + performant avec le callback pour la gestion d'erreur
27 //   if (err) {
28 //     fs.mkdir('./monDossier', (error) => { // mkdir -> création du dossier
29 //       if (error) {
30 //         console.log(error);
31 //       } else {
32 //         console.log('Dossier correctement créé');
33 //       }
34 //     }
35 //   } else {
36 //     fs.rmdir('./monDossier', (error) => { // rmdir -> suppression du dossier
37 //       if (error) {
38 //         console.log(error);
39 //       } else {
40 //         console.log('Dossier correctement supprimé');
41 //       }
42 //     }
43 //   }
44 // })
45
46 // objet Error
47
48 // mkdir => création d'un dossier : monDossier
49 // rmdir => suppression d'un dossier
50 // touch gestionFichiers.js => création d'un fichier : gestionFichiers.js
51
52 // ctrl + MAJ + L ( ou ctrl + F2) => sélection de TOUTES les occurrences
53 // ctrl + d => sélection UNE à UNE des occurrences suivantes
54
55
56 // ===== CREATION d'un FICHIER dans un DOSSIER DIRECTEMENT =====
57
58 // fs.access('./monDossier', (err) => { // access -> Tester si le chemin existe // + performant avec le callback pour la gestion d'erreur
59 //   if (err) {
60 //     fs.mkdir('./monDossier', (error) => { // mkdir -> création du dossier
61 //       if (error) {
62 //         console.log(error);
63 //       } else {
64 //         console.log('Dossier correctement créé');
65 //         fs.writeFile(
66 //           './monDossier/monFichier.txt',
67 //           'Ce fichier est écrit avec node.js',
68 //           (err) => {
69 //             if (err) {
70 //               console.log(err);
71 //             } else {
72 //               console.log('le fichier a été créé correctement');
73 //             }
74 //           }
75 //         )
76 //       }
77 //     }
78 //   }
79 // })
80
81 // ===== 2nd version
82
83 const makeFile = () => {
84   fs.writeFile(
85     './monDossier/monFichier.txt',
86     'Ce fichier est écrit avec node.js',
87     (err) => {
88       if (err) {
89         console.log(err);
90       } else {
91         console.log('le fichier a été créé correctement');
92       }
93     }
94   )
95 }
96
97 // OU =====
98
99 // const makeFile = () => {
100 //   fs.writeFile(
101 //     './monDossier/monFichier.txt',
102 //     'Ce fichier est écrit avec node.js',
103 //     err => err ? console.log(err) : console.log('le fichier a été créé correctement')
104 //   );
105 // };
106
107
108 const createDir = () => {
109   fs.mkdir('./monDossier', (error) => { // mkdir -> création du dossier
110     if (error) {
111       console.log(error);
112     } else {
113       makeFile()
114     }
115   }
116 // sur une seule ligne
117   Error ? console.log(error) : makeFile(); // if -> ? ET else -> :
118 })
119
120 // ou ENCORE mieux !!!!!!!
121
122 const createDir = () => fs.mkdir('./monDossier', Error ? console.log(error) : makeFile())
123
124 fs.access('./monDossier', (err) => {
125   if (err) {
126     createDir()
127   }
128 })
129
130 // MIEUX =>
131
132 fs.access('./monDossier', err => err && createDir())

```

const fs = require('fs');

```

1  ===== CREATION et SUPPRESSION de DOSSIER
2
3  if (!fs.existsSync('./monDossier')) {
4    fs.mkdir('./monDossier', (Error)=>{ // mkdir -> création du dossier
5      if (Error) {
6        console.log(Error);
7      } else {
8        console.log('Dossier correctement créé');
9      }
10   })
11 } else {
12
13   fs.rmdir('./monDossier', (Error)=>{ // rmdir -> suppression du dossier
14     if (Error) {
15       console.log(Error);
16     } else {
17       console.log('Dossier correctement supprimé');
18     }
19   })
20 }
21
22 ===== 2nd méthode
23
24 fs.access('./monDossier', (err)=>{ // access -> Tester si le chemin existe // + performant avec le callback pour la gestion d'erreur
25   if (err){
26     fs.mkdir('./monDossier', (Error)=>{ // mkdir -> création du dossier
27       if (Error) {
28         console.log(Error);
29       } else {
30         console.log('Dossier correctement créé');
31       }
32     })
33   } else {
34     fs.rmdir('./monDossier', (Error)=>{ // rmdir -> suppression du dossier
35       if (Error) {
36         console.log(Error);
37       } else {
38         console.log('Dossier correctement supprimé');
39       }
40     })
41   }
42 })

```

===== CREATION et SUPPRESSION de DOSSIER

```

if (!fs.existsSync('./monDossier')) { fs.mkdir('./monDossier', (Error)=>{ // mkdir -> création du dossier if (Error) {
console.log(Error); } else { console.log('Dossier correctement créé'); } }) } else {

```

```

fs.rmdir('./monDossier', (Error)=>{ // rmdir -> suppression du dossier
  if (Error) {
    console.log(Error);
  } else {
    console.log('Dossier correctement supprimé');
  }
})

```

```

}

```

===== 2nd méthode

```

fs.access('./monDossier', (err)=>{ // access -> Tester si le chemin existe // + performant avec le callback pour la gestion d'erreur if (err){ fs.mkdir('./monDossier', (Error)=>{ // mkdir -> création du dossier if (Error) {
console.log(Error); } else { console.log('Dossier correctement créé'); } }) } else { fs.rmdir('./monDossier',
(Error)=>{ // rmdir -> suppression du dossier if (Error) { console.log(Error); } else { console.log('Dossier
correctement supprimé'); } }) })

```

```

1  objet Error
2
3  mkdir => création d'un dossier : monDossier
4  rmdir => suppression d'un dossier
5  touch gestionFichiers.js => création d'un fichier : gestionFichiers.js
6
7  ctrl + MAJ + L ( ou ctrl + F2 )=> sélection de TOUTES les occurrences
8  ctrl + d ==> sélection UNE à UNE des occurrences suivantes

```

objet Error

mkdir => création d'un dossier : monDossier rmdir => suppression d'un dossier touch gestionFichiers.js => création d'un fichier : gestionFichiers.js

ctrl + MAJ + L (ou ctrl + F2) ==> sélection de TOUTES les occurrences ctrl + d ==> sélection UNE à UNE des occurrences suivantes



```

1  ===== CREATION d'un FICHIER dans un DOSSIER DIRECTEMENT =====
2
3  fs.access('./monDossier', (err)=>{ // access -> Tester si le chemin existe // + performant avec le callback pour la gestion d'erreur
4    if (err){
5      fs.mkdir('./monDossier', (Error)=>{ // mkdir -> création du dossier
6        if (Error) {
7          console.log(Error);
8        } else {
9          console.log('Dossier correctement créé');
10         fs.writeFile(
11           './monDossier/monFichier.txt',
12           'Ce fichier est écrit avec node.js',
13           (err) => {
14             if (err) {
15               console.log(err);
16             } else {
17               console.log('le fichier a été créé correctement');
18             }
19           }
20         )
21       })
22     }
23   })
24 })
25
26 ===== 2nd version
27
28 const makeFile = () => {
29   fs.writeFile(
30     './monDossier/monFichier.txt',
31     'Ce fichier est écrit avec node.js',
32     (err) => {
33       if (err) {
34         console.log(err);
35       } else {
36         console.log('le fichier a été créé correctement');
37       }
38     }
39   )
40 }

```

===== CREATION d'un FICHIER dans un DOSSIER DIRECTEMENT

=====

fs.access('./monDossier', (err)=>{ // access -> Tester si le chemin existe // + performant avec le callback pour la gestion d'erreur if (err){ fs.mkdir('./monDossier', (Error)=>{ // mkdir -> création du dossier if (Error) { console.log(Error); } else { console.log('Dossier correctement créé'); fs.writeFile('./monDossier/monFichier.txt', 'Ce fichier est écrit avec node.js', (err) => { if (err) { console.log(err); } else { console.log('le fichier a été créé correctement'); } }) } } } }

===== 2nd version

const makeFile = () => { fs.writeFile('./monDossier/monFichier.txt', 'Ce fichier est écrit avec node.js', (err) => { if (err) { console.log(err); } else { console.log('le fichier a été créé correctement'); } }) }

```

1 // === SUPPRESSION - CREATION d'un FICHIER =====
2
3 if (!fs.existsSync('./monDossier/monFichier.txt')) {
4   fs.writeFile(
5     './monDossier/monFichier.txt',
6     'Ce fichier est écrit avec node.js',
7     (err) => {
8       if (err) {
9         console.log(err);
10      } else {
11        console.log('le fichier a été créé correctement');
12      }
13    }
14  )
15 } else {
16   fs.unlink ('./monDossier/monFichier.txt', (Error) => {
17     if (Error) {
18       console.log(Error);
19     } else {
20       console.log('Dossier correctement supprimé');
21     }
22   })
23 }
24

```

// === SUPPRESSION - CREATION d'un FICHIER =====

```

if (!fs.existsSync('./monDossier/monFichier.txt')) { fs.writeFile( './monDossier/monFichier.txt', 'Ce fichier est écrit
avec node.js', (err) => { if (err) { console.log(err); } else { console.log('le fichier a été créé correctement'); } } ) }
else { fs.unlink ('./monDossier/monFichier.txt', (Error) => { if (Error) { console.log(Error); } else {
console.log('Dossier correctement supprimé'); } } ) }

```

```

1 OU =====
2
3 const makeFile = () =>
4   fs.writeFile(
5     './monDossier/monFichier.txt',
6     'Ce fichier est écrit avec node.js',
7     err => err ? console.log(err) : console.log('le fichier a été correctement créé')
8   );
9
10
11 const createDir = () => {
12   fs.mkdir('./monDossier', (Error)=>{ // mkdir -> création du dossier
13     // if (Error) {
14     //   console.log(Error);
15     // } else {
16     //   makeFile()
17     // }
18   // sur une seule ligne
19   Error ? console.log(Error) : makeFile (); // if -> ? ET else -> :
20 })}
21
22 ou ENCORE mieux !!!!!!!!
23
24 const createDir = () => fs.mkdir('./monDossier', Error => Error ? console.log(Error) : makeFile ())
25
26 fs.access('./monDossier', (err) => {
27   if (err) {
28     createDir()
29   }
30 })
31
32 MIEUX =>
33
34 fs.access('./monDossier', err => err && createDir())
35
36 RECUPERER le contenu du fichier
37
38 fs.readFile(
39   './monDossier/monFichier.txt',
40   (err, data) => {
41     if (err) {
42       console.log(err);
43     } else {
44       console.log(data.toString());
45     }
46   }
47 )

```

OU =====

```
const makeFile = () => fs.writeFile( './monDossier/monFichier.txt', 'Ce fichier est écrit avec node.js', err => err ? console.log(err) : console.log('le fichier a été correctement créé') );
```

```
const createDir = () => { fs.mkdir('./monDossier', (Error)=>{ // mkdir -> création du dossier // if (Error) { // console.log(Error); // } else { // makeFile() // } // sur une seule ligne Error ? console.log(Error) : makeFile (); // if -> ? ET else -> : })}
```

ou ENCORE mieux !!!!!!!!!!!

```
const createDir = () => fs.mkdir('./monDossier', Error => Error ? console.log(Error) : makeFile ())
```

```
fs.access('./monDossier', (err) => { if (err) { createDir() } })
```

MIEUX =>

```
fs.access('./monDossier', err => err && createDir())
```

RECUPERER le contenu du fichier

```
fs.readFile( './monDossier/monFichier.txt', (err, data) => { if (err) { console.log(err); } else { console.log(data.toString()); } } )
```