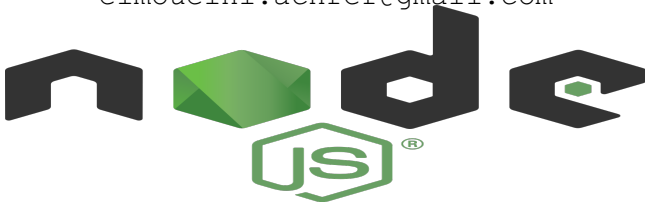


Node.js : REPL

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en programmation par contrainte (IA)
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



Plan

- 1 Introduction
- 2 node
- 3 .editor
- 4 .exit
- 5 .help
- 6 underscore
- 7 Quelques raccourcis

Node.js

REPL : Read-Eval-Print-Loop

- Terminal interactif de **Node.js** (et **JavaScript**)
- Accessible depuis n'importe quelle console
 - **Invite de commandes,**
 - **PowerShell**
 - ...

Node.js

REPL : Read-Eval-Print-Loop

- Terminal interactif de **Node.js** (et **JavaScript**)
- Accessible depuis n'importe quelle console
 - **Invite de commandes**,
 - **PowerShell**
 - ...

Pour lancer PERL

Saisissez `node` dans n'importe quel terminal puis cliquez sur .

Node.js

Avec REPL

- Une instruction commence par >
- Le résultat de l'instruction est affiché à la ligne sans >

© Achref EL M...

Node.js

Avec REPL

- Une instruction commence par >
- Le résultat de l'instruction est affiché à la ligne sans >

Afficher Hello world

```
> console.log("Hello world")  
Hello world
```

Node.js

On peut effectuer des opérations arithmétiques

```
> 3+2  
5
```

© Achref EL MOU

Node.js

On peut effectuer des opérations arithmétiques

```
> 3+2  
5
```

Ou un test logique

```
> 3+2>1+7  
false
```

Node.js

Pour déclarer une variable

```
> var a = 2
```

© Achref EL MOUELHI ©

Node.js

Pour déclarer une variable

```
> var a = 2
```

Ou

```
> x = 2
```

© Achref EL MOU

Node.js

Pour déclarer une variable

```
> var a = 2
```

Ou

```
> x = 2
```

Pour afficher une variable

```
> x  
2
```

Node.js

Pour déclarer une variable

```
> var a = 2
```

Ou

```
> x = 2
```

Pour afficher une variable

```
> x  
2
```

Pour utiliser une variable

```
> x + 3  
5
```

Node.js

Pour une structure conditionnelle

```
> if (x > 0)  
... console.log('positif')
```

© Achref EL MOUELHI

Node.js

Pour une structure conditionnelle

```
> if (x > 0)
... console.log('positif')
```

Pour un bloc de code plus long ou plus complexe, on peut démarrer le mode

.editor

```
> .editor
// Entering editor mode (Ctrl+D to finish, Ctrl+C to cancel)
if (x > 0)
console.log('positif')
else
console.log('négatif')
```

Node.js

Pour quitter REPL

```
> .exit
```

Node.js

Pour afficher la liste de commandes, on utilise la commande `.help`

```
> .help
.break      Sometimes you get stuck, this gets you out
.clear      Alias for .break
.editor     Enter editor mode
.exit       Exit the REPL
.help       Print this help message
.load       Load JS from a file into the REPL session
.save       Save all evaluated commands in this REPL session to a file
```

Node.js

Pour récupérer le résultat de l'opération précédente, on utilise `_`

```
> 2+3  
5  
> _+4  
9  
>
```

Node.js

Quelques raccourcis

- `CTRL` + `C` (une fois) $\equiv$ `.break`
- `CTRL` + `C` (deux fois) $\equiv$ `.exit`
- `CTRL` + `D` (une fois) $\equiv$ `.exit`
- `TAB` (deux fois) $\equiv$ pour l'auto-complétion
- ...