

Rappels JavaScript ES6

Steeve Lefort

2025

Exercice 1 : Variables et constantes

Énoncé

Créez un fichier "01.js" et remplissez-le avec le code suivant:

```
const result = 0;  
const value = 5;  
const multi = 5;
```

```
result = value * multi;
```

```
console.log(result);
```

Lancez le programme avec Node.JS (node 01.js). Que constatez-vous ? Essayez d'améliorer la situation.

Exercice 2 : Tableaux et constantes

Énoncé

Créez un fichier “02.js” et remplissez-le avec le code suivant:

```
const fruits = ['pomme', 'fraise', 'endive'];
```

```
fruits[2] = 'cerise';
```

```
console.log(fruits);
```

Lancez le programme avec Node.JS. Que constatez-vous ?

Exercice 3 : Portée des variables

Énoncé

Créez un fichier "03.js" et remplissez-le avec le code suivant:

```
for (var i = 0; i < 10; i++){  
    console.log(i);  
}  
console.log(i);
```

Instructions

- 1 Lancez le programme avec Node.JS et constatez son effet
- 2 Remplacez "var" par "let"
- 3 Que constatez-vous ?
- 4 Essayez de corriger le programme (sans utiliser 'var')

Exercice 4 : Opérateur ternaire

Énoncé

Créez un fichier “04.js” et remplissez-le avec le code suivant:

```
const points = Math.floor(Math.random()*5);  
  
console.log(`Vous avez ${points} point`);
```

Instructions

- 1 Lancez plusieurs fois le programme
- 2 Le mot “point” est toujours au singulier
- 3 Utilisez l'opérateur ternaire pour régler le problème
- 4 Documentation : Opérateur conditionnel

Exercice 5 : Opérateur REST

Énoncé

Créez un fichier 05.js puis utilisez l'opérateur de REST pour réaliser une fonction "add" acceptant un nombre illimité d'arguments, et retournant leur somme.

Exemple

```
console.log(add(5,5,5)) // Doit retourner: 15 (5+5+5)
```

Exercice 6 : Déstructuration de tableau

Énoncé

Créez un fichier “06.js” et remplissez-le avec le code suivant:

```
const fruits = ['pomme', 'orange', 'banane'];
```

Instructions

- 1 Utilisez la déstructuration pour placer les 3 valeurs dans les variables:
 - first
 - second
 - third
- 2 Affichez ces variables avec console.log

Exercice 7 : Déstructuration d'objet

Énoncé

Créez un fichier "07.js" et remplissez-le avec le code suivant:

```
const jedi = { firstname: 'Obi-Wan', lastname: 'Kenobi' }
```

Instructions

- 1 Utilisez la déstructuration pour extraire les propriétés dans les variables:
 - firstname
 - lastname
- 2 Affichez ces variables avec console.log

Exercice 8 : Méthode map

Énoncé

Créez un fichier "08.js" avec le code suivant:

```
const cats = [ 'Cooky', 'Pixel', 'Pixie', 'Ruby', 'Figaro' ];  
console.log(cats);
```

Objectif

Utilisez la méthode "map" pour obtenir:

```
[  
  '<li>Cooky</li>',  
  '<li>Pixel</li>',  
  '<li>Pixie</li>',  
  '<li>Ruby</li>',  
  '<li>Figaro</li>'  
]
```

Documentation : [Array.prototype.map\(\)](#)

Exercice 9 : Promesses & AJAX

Énoncé

Utilisez l'API suivante pour afficher des citations aléatoires de Kaamelott:

<https://kaamelott.chaudie.re/api/random> ou

<https://kaamelott.formacitron.com/api/random>

Instructions

- 1 Explorez l'API avec votre navigateur
- 2 Utilisez l'API fetch pour interroger cette API
- 3 Affichez les citations reçues

Documentation

- Documentation API Fetch
- Documentation Promesses