



# Jour 1



Créer un fichier `jour01.js` : pour chaque exercice listé ci dessous, vous écrirez un commentaire le nom de l'exercice dans votre fichier pour vous y retrouver. Vous pouvez commenter le code de chaque exercice une fois terminé pour ne pas saturer votre console !

```
// 01 - Hello World  
  
> your code here  
  
// 02 - etc...  
  
> etc...
```

**Créez un repo GitHub nommé "js-digitous" spécialement pour ce module.  
N'oubliez pas de push régulièrement.**

# 01 - Hello World

Écrivez le code suivant dans votre fichier :

```
console.log("Hello World !");
```

Lancez maintenant la commande suivante dans le terminal :

```
$ node jour01.js
```

Vous devriez voir apparaître "Hello World !" : bravo ! Vous venez de créer et exécuter votre premier programme !



Vous aurez souvent besoin de lancer la même commande (par exemple, pour exécuter le code du fichier `jour01.js` pour chaque exercice). Au lieu de taper la commande à chaque fois, vous pouvez appeler l'historique de commande avec la flèche du haut !

# 02 - String

- Dans votre fichier, créez une variable `test`
- Donnez lui la valeur `"My name is xxx"` avec votre nom à la place de "xxx"
- Utilisez ensuite la fonction `console.log()` pour afficher la variable `test` dans la console

- Lancez ensuite la commande pour exécuter le code

## 03 - Concatenation

- Dans votre fichier, créez une variable `name`
- Donnez lui la valeur `"xxx"` avec votre nom à la place de "xxx"
- Utilisez ensuite la fonction `console.log()` pour afficher le texte "Nice to meet you" + la variable `name` dans la console
- Lancez ensuite la commande pour exécuter le code

## 04 - String Length

- Dans votre fichier, créez une variable `testLength`
- Donnez lui la valeur `"I'm very long !"`
- Utilisez ensuite la fonction `console.log()` pour afficher la longueur de `testLength`
- Lancez ensuite la commande pour exécuter le code

## 05 - Replace

- Dans votre fichier, créez une variable `food`
- Donnez lui la valeur `"croissant is meh"`
- Utilisez la méthode `.replace()` pour remplacer "meh" par "so good"
- Utilisez ensuite la fonction `console.log()` pour afficher le résultat dans la console
- Lancez ensuite la commande pour exécuter le code



À partir de maintenant, on vous donnera un peu moins d'indices dans les exercices. C'est à vous de trouver toutes les étapes nécessaires, on ne vous indiquera que les consignes de bases !

## 06 - Up and Down

- Créez une variable `basic` contenant la valeur `"This is Cool"`
- Créez une variable `basicUp` contenant la version en majuscules de `basic`
- Créez une variable `basicDown` contenant la version en minuscules de `basic`
- Faites apparaître les trois variables dans la console

## 07 - Split

- Créez une variable `word` contenant la valeur `"banana"`
- Créez une variable `letters` contenant un tableau des lettres de `word`
- Faites apparaître `letters` dans la console

```
// Le résultat doit être un tableau où chaque élément est une lettre:  
[ 'b', 'a', 'n', 'a', 'n', 'a' ]
```

## 08 - Template

- Créez une variable `age` contenant la valeur `"xx"` en remplaçant "xx" par votre âge
- Créez une variable `template` contenant la phrase "I'm \_\_\_\_ years old" en utilisant une template literal et la variable `age`
- Faites apparaître `template` dans la console

---

## ★ Bonus

C'est le premier jour, alors on y va doucement. En guise de bonus aujourd'hui, on recommande de lire la documentation MDN (on vous en a parlé, c'est le wikipedia du code !) sur les strings. C'est une bonne idée pour deux raisons :

- Il faudra vous habituer à lire de la documentation alors autant commencer avec des concepts simples
- Regardez bien la section "Méthodes" qui contient des choses qu'on a pas vues dans la leçon !
  - N'hésitez pas à reproduire les exemples et à afficher leur résultat (par exemple, substring avec 1 ou 2 paramètres)
  - Jouez avec l'échappement, essayez d'afficher un string avec deux lignes, un string mélangeant des guillemets simple et des guillemets doubles
  - Créez une variable `changed` ayant pour valeur "Bonjour", utilisez la doc pour afficher `Banjaur`

Pour la lire c'est par ici ⇒ [MDN - String](#)