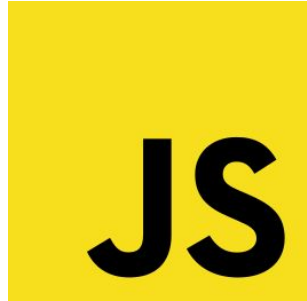


Javascript



## Objectifs

1. Stocker des informations dans la mémoire de l'ordinateur en utilisant **des variables**
2. Comprendre les **différents types de variables**

## Créer une variable

Les variables **permettent d'allouer un espace dans la mémoire vive**, où on peut y stocker des données.

## Créer une variable

Les variables **permettent d'allouer un espace dans la mémoire vive**, où on peut y stocker des données.

Pour créer une variable :

## Créer une variable

Les variables **permettent d'allouer un espace dans la mémoire vive**, où on peut y stocker des données.

Pour créer une variable :

- on utilise le **mot-clé** **const**

## Créer une variable

Les variables **permettent d'allouer un espace dans la mémoire vive**, où on peut y stocker des données.

Pour créer une variable :

- on utilise le **mot-clé const**
- suivi du nom de la variable

## Créer une variable

Les variables **permettent d'allouer un espace dans la mémoire vive**, où on peut y stocker des données.

Pour créer une variable :

- on utilise le **mot-clé const**
- suivi du nom de la variable
- puis du symbole **=** suivi de la valeur que l'on souhaite lui assigner

## Créer une variable

Les variables **permettent d'allouer un espace dans la mémoire vive**, où on peut y stocker des données.

Pour créer une variable :

- on utilise le **mot-clé const**
- suivi du nom de la variable
- puis du symbole = suivi de la valeur que l'on souhaite lui assigner

```
const country = "France"  
console.log(country);  
// m'affichera "France"
```



## Créer une variable

Si notre variable **peut voir sa valeur être ré-attribué**, alors le mot clé pour créer notre variable n'est plus `const` mais `let`

## Créer une variable

Si notre variable peut voir sa valeur être ré-assignée, alors le mot clé pour créer notre variable n'est plus `const` mais `let`

```
let country = "France"  
country = "Gaule"  
console.log(country);
```

## Créer une variable

Si notre variable peut voir sa valeur être ré-attribué, alors le mot clé pour créer notre variable n'est plus `const` mais `let`

```
let country = "France"  
country = "Gaule"  
console.log(country);
```

Si jamais j'essaie de ré-attribuer une valeur à une variable déclarée avec le mot clé **`const`**, j'obtiens une erreur :

## Créer une variable

Si notre variable peut voir sa valeur être ré-attribué, alors le mot clé pour créer notre variable n'est plus `const` mais `let`

```
let country = "France"  
country = "Gaule"  
console.log(country);
```

Si jamais j'essaie de ré-attribuer une valeur à une variable déclarée avec le mot clé **`const`**, j'obtiens une erreur :

```
✖ Uncaught TypeError: Assignment to constant variable.      script.js:2  
  at script.js:2
```

## Nommer une variable

En Javascript (et dans n'importe quel langage de programmation), on va chercher à **nommer nos variables de la façon la plus claire possible.**

## Nommer une variable

En Javascript (et dans n'importe quel langage de programmation), on va chercher à **nommer nos variables de la façon la plus claire possible**.

On cherche à rendre notre code le **plus lisible et explicite possible** !

## Nommer une variable

En Javascript (et dans n'importe quel langage de programmation), on va chercher à **nommer nos variables de la façon la plus claire possible**.

On cherche à rendre notre code le **plus lisible et explicite possible** !

```
// On peut écrire :  
const a = "France"  
// tout autant qu'on peut écrire :  
const country = "France"
```

## Nommer une variable

En Javascript (et dans n'importe quel langage de programmation), on va chercher à **nommer nos variables de la façon la plus claire possible**.

On cherche à rendre notre code le **plus lisible et explicite possible** !

```
// On peut écrire :  
const a = "France"  
// tout autant qu'on peut écrire :  
const country = "France"
```

On préférera naturellement le second car il est nettement plus lisible et explicite.



## Nommer une variable

Pour nommer une variable, on ne peut pas tout faire n'importe comment.. !. Il y a des règles à respecter.

## Nommer une variable

Pour nommer une variable, on ne peut pas tout faire n'importe comment.. !. Il y a des règles à respecter.

### Conventions de nommage pour une variable en JS :

- On peut utiliser : Lettres, chiffres, \_ , \$

## Nommer une variable

Pour nommer une variable, on ne peut pas tout faire n'importe comment.. !. Il y a des règles à respecter.

### Conventions de nommage pour une variable en JS :

- On peut utiliser : Lettres, chiffres, \_ , \$
- Le premier caractère ne peut pas être un chiffre

## Nommer une variable

Pour nommer une variable, on ne peut pas tout faire n'importe comment.. !. Il y a des règles à respecter.

### Conventions de nommage pour une variable en JS :

- On peut utiliser : Lettres, chiffres, \_ , \$
- Le premier caractère ne peut pas être un chiffre
- Le nom de la variable ne peut pas être un mot clé JS : const, let, new ou class

## Nommer une variable

Pour nommer une variable, on ne peut pas tout faire n'importe comment.. !. Il y a des règles à respecter.

### Conventions de nommage pour une variable en JS :

- On peut utiliser : Lettres, chiffres, \_ , \$
- Le premier caractère ne peut pas être un chiffre
- Le nom de la variable ne peut pas être un mot clé JS : const, let, new ou class
- Langue anglaise obligatoire

## Nommer une variable

Pour nommer une variable, on ne peut pas tout faire n'importe comment.. !. Il y a des règles à respecter.

### Conventions de nommage pour une variable en JS :

- On peut utiliser : Lettres, chiffres, \_ , \$
- Le premier caractère ne peut pas être un chiffre
- Le nom de la variable ne peut pas être un mot clé JS : const, let, new ou class
- Langue anglaise obligatoire
- Style : **camelCase**

## Nommer une variable

Pour nommer une variable, on ne peut pas tout faire n'importe comment.. !. Il y a des règles à respecter.

### Conventions de nommage pour une variable en JS :

- On peut utiliser : Lettres, chiffres, \_ , \$
- Le premier caractère ne peut pas être un chiffre
- Le nom de la variable ne peut pas être un mot clé JS : const, let, new ou class
- Langue anglaise obligatoire
- Style : **camelCase**
- Nommage le plus clair et lisible possible

Pratiquons !

