

## 1 - Fonctions fléchées

Une expression de fonction fléchée (arrow function en anglais) permet d'avoir une syntaxe plus courte que les expressions de fonction et ne possède pas ses propres valeurs pour this, arguments, super, ou new.target. Les fonctions fléchées sont souvent anonymes et ne sont pas destinées à être utilisées pour déclarer des méthodes.

Reference :

[https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Functions/Arrow\\_functions](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Functions/Arrow_functions)

---

## 2 - Comparaison des fonctions traditionnelles aux fonctions fléchées

```
// Traditional Anonymous Function
(function (a, b) {
  return a + b + 100;
});

// Arrow Function
(a, b) => a + b + 100;

const a = 4;
const b = 2;

// Traditional Anonymous Function (no arguments)
(function() {
  return a + b + 100;
});

// Arrow Function (no arguments)
() => a + b + 100;
```

Pour les fonctions nommées, nous traitons les expressions fléchées comme des variables :

```
// Traditional Function
function bob(a) {
  return a + 100;
}

// Arrow Function
const bob2 = (a) => a + 100;
```

---

## 3 - Syntaxe

### Syntaxe de base

Un param. Avec une expression simple, return n'est pas nécessaire :

```
param => expression  
(param) => expression
```

Plusieurs paramètres nécessitent des parenthèses. Avec une expression simple, return n'est pas nécessaire :

```
(param1, paramN) => expression
```

Les instructions multilignes nécessitent des accolades et un retour : les parenthèses sont facultatives avec un seul paramètre.

```
param => {  
  const a = 1;  
  return a + param;  
}
```

Plusieurs paramètres nécessitent des parenthèses. Les instructions multilignes nécessitent des accolades et renvoient :

```
(param1, paramN) => {  
  const a = 1;  
  return a + param1 + paramN;  
}
```

## Syntaxe avancée

Pour renvoyer une expression littérale d'objet, il faut des parenthèses autour de l'expression :

```
(params) => ({ foo: "a" })  
  
// returning the object { foo: "a" }
```

Les paramètres de repos sont pris en charge et nécessitent toujours des parenthèses :

```
(a, b, ...r) => expression
```

Les paramètres par défaut sont pris en charge et nécessitent toujours des parenthèses :

```
(a=400, b=20, c) => expression
```

La déstructuration dans les paramètres est prise en charge et nécessite toujours des parenthèses :

`([a, b] = [10, 20]) => a + b; // le resultat est 30`  
`({ a, b } = { a: 10, b: 20 }) => a + b; // le resultat est 30`

---

## 4 - Exemples de fonctions fléchées

Avant Flèche :

```
hello = function() {  
    return "Hello World!";  
}
```

Avec la fonction Flèche :

```
hello = () => {  
    return "Hello World!";  
}
```

Les fonctions fléchées renvoient la valeur par défaut :

```
hello = () => "Hello World!";
```

Fonction flèche avec paramètres :

```
hello = (val) => "Hello " + val;
```

Fonction fléchée sans parenthèses :

```
hello = val => "Hello " + val;
```

Reference: [https://www.w3schools.com/js/js\\_arrow\\_function.asp](https://www.w3schools.com/js/js_arrow_function.asp)

---