

Notions

- ▶ **Fonctions fléchées** : Elles correspondent à une syntaxe plus rapide de l'écriture d'une fonction et permettent de résoudre les problématiques rencontrées par certains mots-clés tels que « this » ou « super »
- ▶ **Nombre d'arguments** : Plusieurs écritures sont possibles en fonction du nombre d'arguments :
 - ▶ 0 argument : `() => {}`
 - ▶ 1 argument : `arg => {}`
 - ▶ N arguments : `(arg1,arg2,...) => {}`
- ▶ **Nombre de ligne de retour** : Si la fonction ne retourne qu'une ligne de code, alors les accolades sont optionnelles

Exemples

```
const fct1 = () => console.log("coucou");  
fct1();
```

coucou

```
const fct2 = () => "coucou";  
console.log(fct2());
```

```
const fct3 = () => {  
  console.log("Salut");  
  console.log("Bonjour");  
}  
fct3();
```

Salut
Bonjour

```
const fct4 = () => {  
  let x = 5;  
  let y = 3;  
  return x * y;  
}  
console.log(`Resultat : ${fct4()}`);
```

```
const fct5 = y => 5*y;  
console.log(`Resultat : ${fct5(3)}`);
```

Resultat : 15

```
const fct6 = y => {  
  let x = 5;  
  return x * y;  
}  
console.log(`Resultat : ${fct6(3)}`);
```

```
const fct7 = (x,y) => x*y;  
console.log(`Resultat : ${fct7(3,5)}`);
```

```
const fct8 = (x,y) => {  
  let z = 10;  
  return x*y*z;  
}  
console.log(`Resultat : ${fct8(3,5)}`);
```

Resultat : 150