

Bonnes pratiques

L'indentation

L'indentation est le décalage par rapport à la marge gauche à chaque entrée dans un bloc imbriqué. L'espacement choisi est généralement d'*un tab* bien qu'il soit à la convenance du développeur. L'indentation sert à avoir un code simple et rapide à lire. Par exemple avec 3 boucles « for » imbriquées :

Sans-indentation

```
for (let i = 0 ; i < 2 ; i++) {  
  
for (let i = 0 ; i < 5 ; i++) {  
  
for(let i = 0 ; i < 10 ; i++) {  
  
console.log('trois');  
  
}  
  
console.log('deux');  
  
}  
  
console.log('un');  
  
}
```

Avec indentation

```
for (let i = 0 ; i < 2 ; i++) {  
    for (let i = 0 ; i < 5 ; i++) {  
        for(let i = 0 ; i < 10 ; i++) {  
            console.log('trois');  
        }  
        console.log('deux');  
    }  
    console.log('un');  
}
```

L'attribution de noms explicites

Que ce soit pour les fonctions, les variables ou autre, un nom explicite vous permettra à vous et aux développeurs qui liront votre code de pouvoir comprendre le but d'une fonction ou ce que contient une variable sans avoir à décortiquer le code. De plus avoir des noms distincts simplifiera la tâche lors de sessions de débogage.

Nom non-explicite

```
function a(b) {  
    return b * b ;  
}
```

a(7) ; // On a aucune idée de ce que fait cette fonction sans la lire et cela complique la compréhension

Nom explicite

```
function carre(num) {  
    return num * num ;  
}
```

carre(7); // On peut deviner facilement ce que cette fonction va accomplir

On utilisera également une notation des noms qui permettra une lecture aisée. Par exemple le camel case où les mots sont collés mais où chaque mot suivant le premier à une majuscule.

Sans camel case

firstletterlowercase() // Dur à lire

Avec camel case

firstLetterLowerCase() // Plus simple à lire