



# 5IPRO

## Principes algorithmiques et programmation

Benjamin Delbar



# Cours 13

Feedback découverte des différents langages

Les Fonctions

- Pourquoi
- Comment

Normes de syntaxes

Exercices



# Feedback découverte

Quels langages avez-vous essayé?

Qu'est ce qui vous a marqué lors de la réalisation des exercices?



# Les Fonctions

Qu'est ce que c'est?

Ce sont des “blocs” de code que l'on va pouvoir réutiliser

On va découper notre code en petit morceau, réutilisable, et utilisant une entrée et une sortie (input et output)





# Les Fonctions

```
function somme (a, b) {  
    let maSomme = a + b;  
    return maSomme;  
}
```



# Les Fonctions

```
function somme (a, b) {  
  let maSomme = a + b;  
  return maSomme;  
}
```

Mot clef pour définir notre fonction



# Les Fonctions

```
function somme (a, b) {  
    let maSomme = a + b;  
    return maSomme;  
}
```

Nom de notre fonction





# Les Fonctions

```
function somme (a, b) {  
    let maSomme = a + b;  
    return maSomme;  
}
```

Paramètres de notre fonction

Les INPUTS



# Les Fonctions

```
function somme (a, b) {  
    let maSomme = a + b;  
    return maSomme;  
}
```

Contenu de notre fonction



# Les Fonctions

```
function somme (a, b) {  
  let maSomme = a + b;  
  return maSomme;  
}
```

Instruction(s)



# Les Fonctions

```
function somme (a, b) {  
  let maSomme = a + b;  
  return maSomme;  
}
```

Sortie de notre fonction

L'OUTPUT



# Les Fonctions

On appelle la fonction que l'on veut exécuter comme ceci

```
function somme(a, b) {  
    let maSomme = a + b;  
    return maSomme;  
}
```

```
let total = somme(10, 15);  
console.log(total) //vaudra 25
```



# Les Fonctions

Les arguments en entrées :

Peuvent être au nombre de 0, 1, 2, 3, 4, ....

Peuvent être de tout type (Number, String, Boolean, ...)

Porte un nom que l'on choisit



# Les Fonctions

Les arguments en entrées :

Peuvent recevoir une valeur par défaut

```
function somme (a=10, b=5) {  
    return a+b;  
}  
somme() ; //fera 10+5 car pas de valeurs passées pour a et b  
somme(8) ; //fera 8+5 car pas de valeur pour b  
somme(3, 4) ; //fera 3+4 car les 2 entrées sont bien définies
```



# Les Fonctions

Les instructions :

Peuvent faire autant de lignes que nécessaire

On respecte le principe de responsabilité simple

1 fonction = 1 tâche





# Les Fonctions

La sortie :

Ne retourne qu'une variable (de n'importe quel type, tableau, nombre, ...)

le code de la fonction s'arrête si il rencontre un `return`

```
function test () {  
    if (5+5 == 10) {  
        return true;  
    }  
    return "bonjour"; //ne sera jamais appelé  
}
```



# Les Fonctions

Fonction-ception

Une fonction peut en appeler d'autres, qui peuvent elles-mêmes en appeler d'autres etc,

Une fonction peut s'appeler elle-même



# Norme de syntaxe



# Normes de syntaxes

Noms des variables

Commence par une minuscule,

Peuvent être en camelCase => ou avec des under\_score, (juste rester cohérent)

```
let myAge = 5;
```

```
let my_age = 5;
```



# Normes de syntaxes

Noms des variables

Commence par une minuscule,

Peuvent être en camelCase => ou avec des under\_score, (rester cohérent)

```
let myAge = 5;
```

```
let my_age = 5;
```



# Normes de syntaxes

Noms des variables

Porte un nom cohérent,

On évite les a, b, c, d etc si l'on sait ce que représente la variable

```
let name = "Benjamin";  
let persons = ["Benjamin", "Philippe", "Jean", "Anne", "Marie"];
```



# Normes de syntaxes

## L'indentation

=> Le fait de laisser un certain nombre d'espaces (ou de tabulation) entre ses lignes

On ajoute toujours 1 indentation pour chaque block de code compris entre des { }

```
for(let i=0; i<5; i++) {  
    //je commence mon code ici  
    //et surement pas ici  
}
```



# Normes de syntaxes

Le placement des brackets { }

On les place directement après la condition, ou suite à un retour à la ligne

La bracket de fermeture est d'office sur une nouvelle ligne et alignée dans l'indentation

```
if (name == "John") {  
    //code ici  
}
```





# Normes de syntaxes

```
if (name == "John")  
{  
    //code ici  
    if (age == 50)  
    {  
        //code ici  
    }  
    //code ici  
}
```



# Normes de syntaxes

Pour les fonctions

```
function nomDeLaFonction (argument, argument) {  
  
}
```

Je vous conseille de laisser un espace entre le nom et les arguments  
En recherchant dans votre code =>

```
nomDeLaFonction( => retrouve les appels à la fonction  
nomDeLaFonction ( => retrouve directement la définition de la fonction
```



# Exercices

3 Groupes différents

Groupe 1 => Récap => On revoit ce sur quoi vous avez des lacunes (tableaux, boucles, ...)

Groupe 2 => Exercices => Vous réalisez une liste d'exercice sur les fonctions

Groupe 3 => Exercices avancés => Vous réalisez des exercices plus poussés sur les fonctions