



# Initiation à la programmation

## Les fonctions

# Objectif

- Qu'est-ce qu'une fonction ?
- Savoir quand et comment utiliser une fonction



# Les fonctions

## Préambule : découper un problème



## Préambule : découper un problème

Un des principe fondamentaux de l'algorithme consiste à scinder une tâches trop grande et incompréhensible, en un ensemble de sous-problèmes plus simples à appréhender.

« Diviser pour mieux régner »



# Les fonctions

## L'exemple de Paul

Paul est un enfant timide de 10 ans. Il a envie de faire un cadeau à ses 5 amis, Marc, Clément, Jean, Jacques, et Goldman. Paul a 20 billes à sa disposition qu'il veut offrir à ses amis. Il veut en donner 6 à son meilleur ami, Marc, 20% à son deuxième meilleur ami, Goldman, et répartir le reste équitablement. Si il ne peut pas répartir équitablement, il répartira au mieux, puis offrira ce qu'il reste à son meilleur ami. Aidez Paul à savoir combien de billes donner à ses amis.



# Les fonctions

## L'exemple de Paul

Paul est un enfant timide de 10 ans. Il a envie de faire un cadeau à ses 5 amis, Marc, Clément, Jean, Jacques, et Goldman. Paul a 20 billes à sa disposition qu'il veut offrir à ses amis. Il veut en donner 6 à son meilleur ami, Marc, 20% à son deuxième meilleur ami, Goldman, et répartir le reste équitablement. Si il ne peut pas répartir équitablement, il répartira au mieux, puis offrira ce qu'il reste à son meilleur ami. Aidez Paul à savoir combien de billes donner à ses amis.



# Les fonctions

## L'exemple de Paul

Paul veut répartir 20 billes de la manière suivante :

1. 6 à son meilleur ami
2. 20% à son deuxième meilleur ami
3.  $\frac{1}{3}$  de ce qu'il reste à chacun des 3 amis restants
4. En cas de reste après cette distribution, le reste revient à son meilleur ami



# Les fonctions

## L'exemple de Paul

Paul veut répartir 20 billes de la manière suivante :

1. 6 à son meilleur ami
2. 20% à son deuxième meilleur ami
3.  $\frac{1}{3}$  de ce qu'il reste à chacun des 3 amis restants
4. En cas de reste après cette distribution, le reste revient à son meilleur ami





# Les fonctions

## L'exemple de Paul

```
int billesRestantes, billesPA, billesDA, billesR;  
billesRestantes = 20;  
billesPA = 6;  
billesDA = billesRestantes * 20 / 100  
billesRestantes = billesRestantes - billesPA - billesDA  
billesR = billesRestantes / 3;  
billesPA = billesPA + billesRestantes % 3;
```



## Initiation aux fonctions

Les fonctions sont un outil qui nous permettra de découper notre algorithme en ensemble de sous-algo.

Elles permettent :

- de clarifier l'algorithme
- d'éviter la redondance



# Les fonctions

## Initiation aux fonctions

```
String nom;  
Scanner scan = new Scanner(System.in);  
nom = scan.nextLine();  
saluer(nom);
```

```
public void saluer(String x) {  
    System.out.println(x);  
}
```



# Les fonctions

## Initiation aux fonctions

```
String nom;  
Scanner scan = new Scanner(System.in);  
nom = scan.nextLine();  
saluer(nom);
```

APPEL de la fonction

NOM de la fonction

PARAMETRES de la fonction

CORPS de la fonction

```
public void saluer(String x) {  
    System.out.println(x);  
}
```

Le **NOM** et les **PARAMETRES** de la fonction constituent la SIGNATURE de la fonction



# Les fonctions

## Initiation aux fonctions

*fonction avec paramètre, sans retour*

```
for(int i = 0; i < 10; i++) {  
    afficherParite(i);  
}
```

```
public void afficherParite(int x) {  
    if (x % 2 == 0) {  
        System.out.println("x est pair");  
    } else {  
        System.out.println("x est impair");  
    }  
}
```



# Les fonctions

## Initiation aux fonctions

*fonction avec paramètre, avec retour*

```
int x;  
for(int i = 0; i < 4; i++) {  
    x = carre(i);  
    System.out.println(i);  
    System.out.println(x);  
}
```

```
public void carre(int x) {  
    x = x * x;  
    return x;  
}
```



# Les fonctions

## Initiation aux fonctions

*passage par valeur & passage par référence*



# Les fonctions

## Initiation aux fonctions

*passage par valeur & passage par référence*

```
int monNombre = 4;
int mesNombres = new int[]{5, 6, 7};
System.out.println(monNombre);
System.out.println(mesNombres[0]);
changerValeur(monNombre);
changerValeurTableau(mesNombres);
System.out.println(monNombre);
System.out.println(mesNombres[0]);
```

```
public void changerValeur(int x) {
    x = 99;
}
```

```
public void changerValeurTableau(int[] arr) {
    arr[0] = 22;
}
```





# Les fonctions

## Initiation aux fonctions

*passage par valeur & passage par référence*

```
int monNombre = 4;  
int mesNombres = new int[]{5, 6, 7};  
System.out.println(monNombre);  
System.out.println(mesNombres[0]);  
changerValeur(monNombre);  
changerValeurTableau(mesNombres);  
System.out.println(monNombre);  
System.out.println(mesNombres[0]);
```

```
> 4  
> 5  
> 4  
> 22
```

```
public void changerValeur(int x) {  
    x = 99;  
}
```

```
public void changerValeurTableau(int[] arr) {  
    arr[0] = 22;  
}
```



# Atelier 6

## Manipuler des fonctions





# Initiation à la programmation

## Les fonctions