

# Cours n°1- Algorithmes de base

julien.brajard@upmc.fr, maria.potop-butucaru@lip6.fr

Polytech'Sorbonne

2018-2019

## 1 Les algorithmes

- Introduction
- Construction d'un algorithme
- Structures de base d'un algorithme
- Tester un algorithme
- Exemples

# Pourquoi faire appel à des algorithmes ?

Pour automatiser des tâches

Exemples :

- Métier à tisser
- Méthode de calcul à la main d'une division
- Recette de cuisine
- ...

# Qu'est-ce qu'un algorithme ?

## Définition

Un algorithme est un ensemble ordonné d'instructions simples permettant de résoudre un problème.

# Remarques

## Un algorithme nécessite :

- Des objets sur lesquels travailler,
- Un langage non ambigu,
- Des spécifications (description de l'algorithme).

Il n'existe généralement pas un unique algorithme pour traiter un problème.

# Historique

3ème siècle avant JC *Livre VII des Eléments d'Euclide*

Détermination du plus grand diviseur commun  
entre deux nombres :  $\text{PGCD}(12,8)=4$



8ème siècle après JC *Al-Khwarizmi* : Méthodes de résolution  
d'équations. Son nom est à l'origine du mot "algorithme".

## 1 Les algorithmes

- Introduction
- Construction d'un algorithme
- Structures de base d'un algorithme
- Tester un algorithme
- Exemples



# Construction d'un algorithme

Pour chaque problème, il vous est demandé de définir clairement :

- Les éventuelles **données d'entrée du problème** en précisant leurs types et leur rôle,
- les éventuelles **données de sortie du problème** en précisant leurs types,
- les différentes **instructions** permettant d'obtenir les données de sorties à partir des données d'entrée.

## Exemple

Algorithme qui détermine le prix d'entrée dans un musée (les mineurs payent moitié prix)

## Exemple

Algorithme qui détermine le prix d'entrée dans un musée (les mineurs payent moitié prix)

### Données

**Données d'entrée** : age (entier)

{age du client}

**Données de sortie** : tarif (décimal)

{prix de l'entrée}

## Exemple

Algorithme qui détermine le prix d'entrée dans un musée (les mineurs payent moitié prix)

### Données

**Données d'entrée :** age (entier)

{age du client}

**Données de sortie :** tarif (décimal)

{prix de l'entrée}

### Instructions

**Si** age < 18 **Alors**

tarif  $\leftarrow$  4

**Sinon**

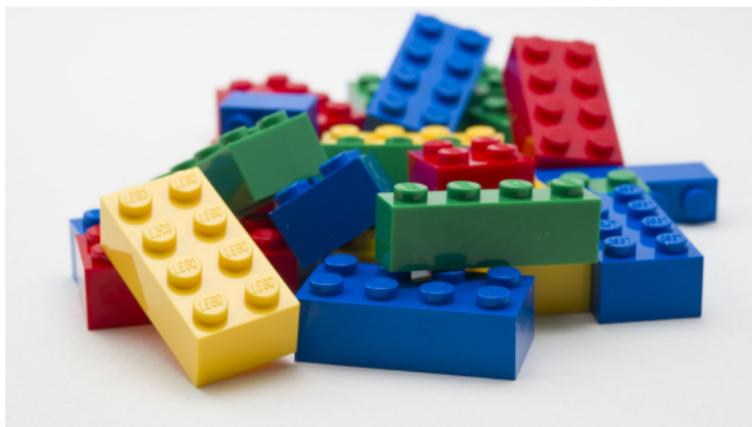
tarif  $\leftarrow$  8

**Fin Si**

**Renvoyer** tarif

## 1 Les algorithmes

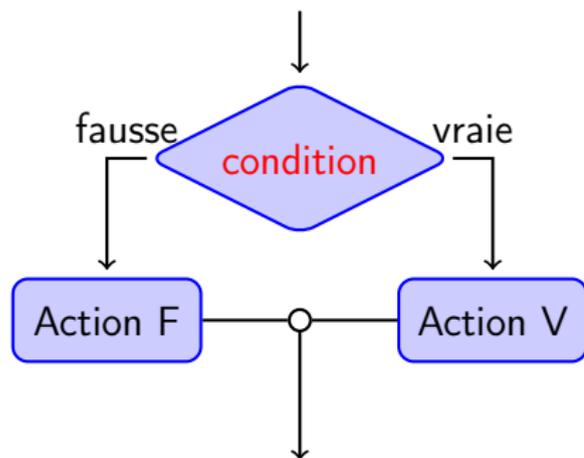
- Introduction
- Construction d'un algorithme
- Structures de base d'un algorithme
- Tester un algorithme
- Exemples



# Instructions élémentaires

- **Opérations arithmétiques de base** (+, −, ×, /, mod, ...)
- **Affectation de valeurs** (ex :  $x \leftarrow 2$ )
- **Afficher, lire**
- Structures conditionnelles
- Structures répétitives

# Structures conditionnelles 1/2

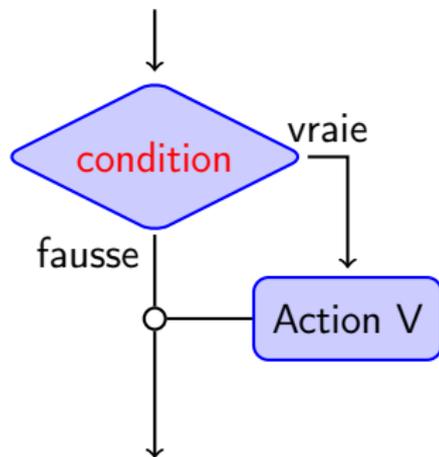


**Si condition Alors**  
Action V  
**Sinon**  
Action F  
**Fin Si**

## Un exemple

**Si  $\text{age} < 18$  Alors**  
tarif  $\leftarrow$  4  
**Sinon**  
tarif  $\leftarrow$  8  
**Fin Si**

## Structures conditionnelles 2/2

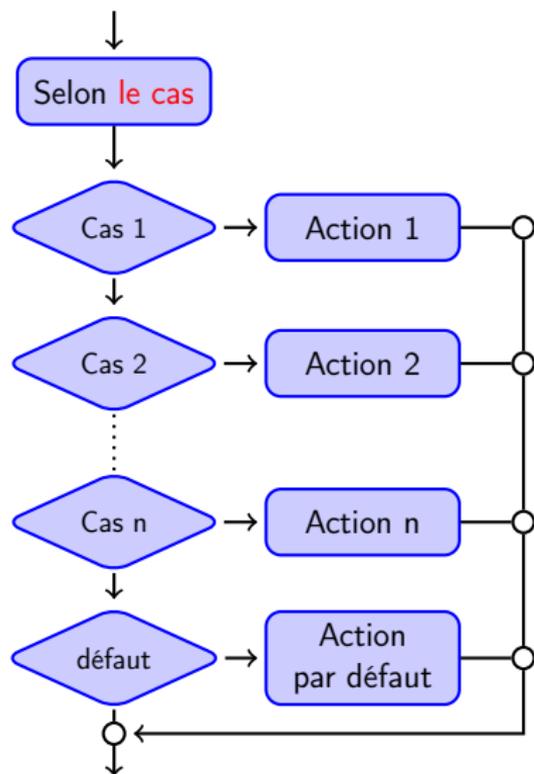


**Si condition Alors**  
Action V  
**Fin Si**

### Un exemple

**Si age < 18 Alors**  
**Afficher** "Réduction de  
50%"  
**Fin Si**

# Structures à choix multiple



## Selon le cas

Cas 1 : Action 1

Cas 2 : Action 2

...

Cas n : Action n

Défaut : Action par défaut

**Fin Selon**

## Un exemple

### Selon valeur de age

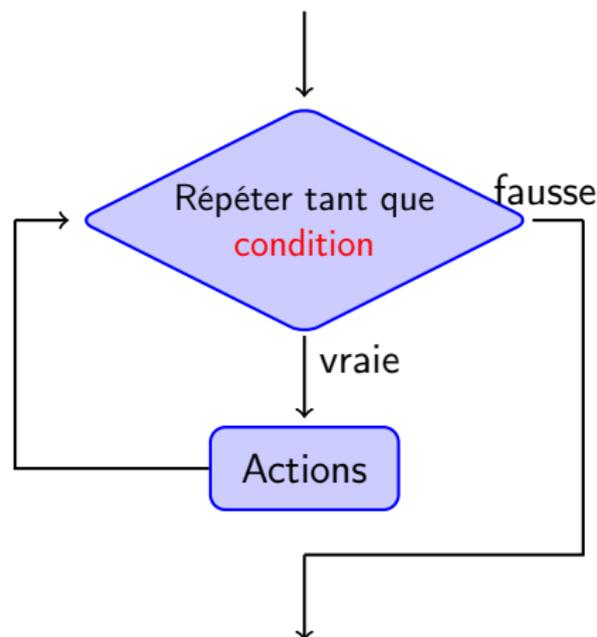
0..6 : tarif ← 0

7..18 : tarif ← 4

Défaut : tarif ← 8

**Fin Selon**

# Structures répétitives 1/2



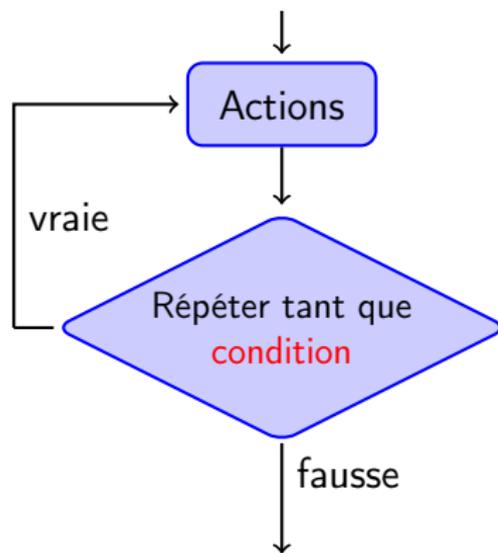
**Tant que** **condition** **Faire**  
Actions  
**Fin Tant que**

Les actions peuvent ne **pas** être exécutées si la condition est fausse au départ.

**Un exemple**

**Tant que** **tickets > 0** **Faire**  
vendre un ticket  
**Fin Tant que**

## Structures répétitives 2/2



**Répéter**

Actions

**Tant que** condition

Les actions sont exécutées au moins une fois.

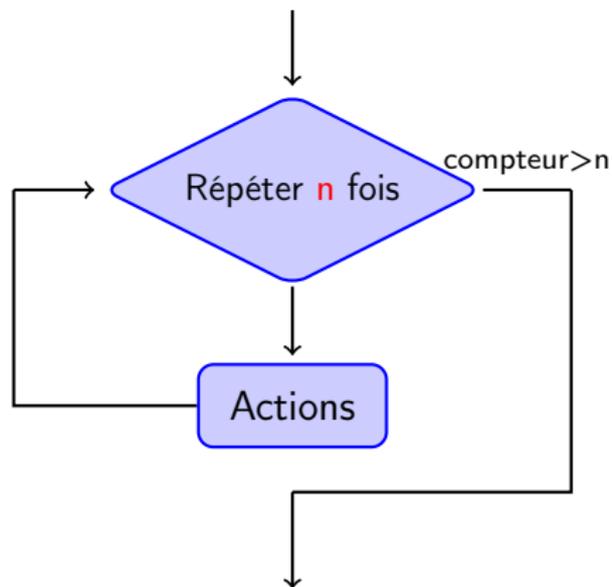
**Un exemple**

**Répéter**

vendre un ticket

**Tant que** ticket > 0

# Structures itératives



```
Pour i = 1 à n Faire  
    Actions  
Fin Pour
```

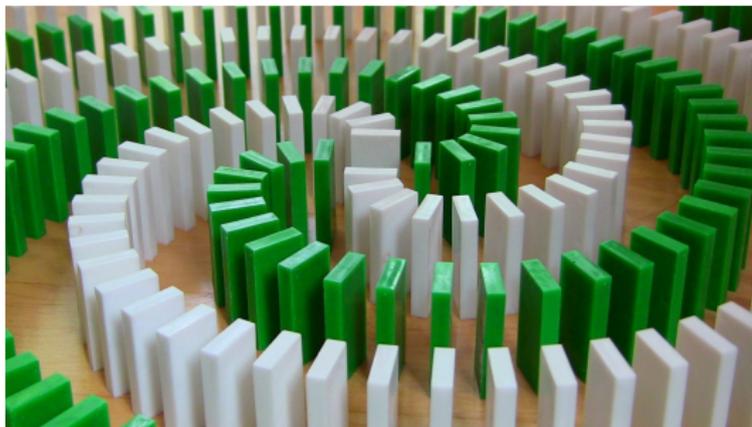
On connaît le nombre d'itérations à l'avance.

## Un exemple

```
Pour i = 1 à nbre_guide Faire  
    Si place pour guide i > 0 Alors  
        Afficher "Il reste des places  
        pour le guide i"  
    Fin Si  
Fin Pour
```

## 1 Les algorithmes

- Introduction
- Construction d'un algorithme
- Structures de base d'un algorithme
- **Tester un algorithme**
- Exemples



# Jeu d'essais

- Faire fonctionner l'algorithme avec des valeurs.
- Traiter tous les cas possibles.
- En particulier les cas pouvant nécessiter à traitement spécifique.

# Un exemple :

Calcul de la valeur absolue d'un nombre

## Données

**Données d'entrée :**  $X$  (entier)

{nombre dont on veut calculer la valeur absolue}

**Données de sortie :** Val\_abs (entier)

{valeur absolue de  $X$ }

## Instructions

**Si**  $X \geq 0$  **Alors**

Val\_abs  $\leftarrow X$

**Sinon**

Val\_abs  $\leftarrow -X$

**Fin Si**

**Renvoyer** Val\_abs

# Un exemple :

Calcul de la valeur absolue d'un nombre

## Données

**Données d'entrée :**  $X$  (entier)

{nombre dont on veut calculer la valeur absolue}

**Données de sortie :**  $Val\_abs$  (entier)

{valeur absolue de  $X$ }

## Instructions

**Si**  $X \geq 0$  **Alors**

$Val\_abs \leftarrow X$

**Sinon**

$Val\_abs \leftarrow -X$

**Fin Si**

**Renvoyer**  $Val\_abs$

## Jeu d'essais

- $X \leftarrow$  un entier négatif
- $X \leftarrow$  un entier positif
- $X \leftarrow 0$

## 1 Les algorithmes

- Introduction
- Construction d'un algorithme
- Structures de base d'un algorithme
- Tester un algorithme
- Exemples

## Exemple 1

Un colis est conforme si ses trois dimensions ( $L \times l \times h$  avec  $h \leq l \leq L$ ) vérifient les conditions suivantes :

- Un rectangle de  $10\text{cm} \times 7\text{cm}$  doit pouvoir s'inscrire dans une face du colis,
- $L + l + h \leq 100 \text{ cm}$ ,
- $L \leq 60\text{cm}$ .

### Exercice

Ecrire l'algorithme permettant de savoir si un colis est conforme ou non.

## Exemple 2

Que fait cet algo ?

**Données d'entrée :**

$X$ , tableau à 2 dimension de  $n \times n$  réels,  
 $v$ , vecteur de  $n$  réels

**Données de sortie :**  $s$ , vecteur de  $n$  réels

**Pour**  $i$  variant de 1 à  $n$  **Faire**

$s(i) \leftarrow 0$

**Pour**  $j$  variant de 1 à  $n$  **Faire**

$s(i) \leftarrow s(i) + X(i,j).v(j)$

**Fin Pour**

**Fin Pour**

**Renvoyer**  $s$