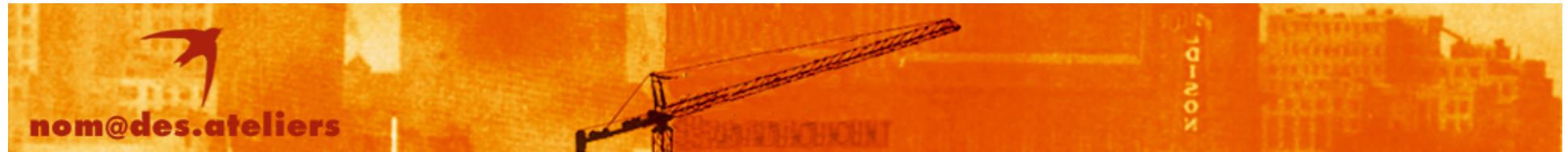


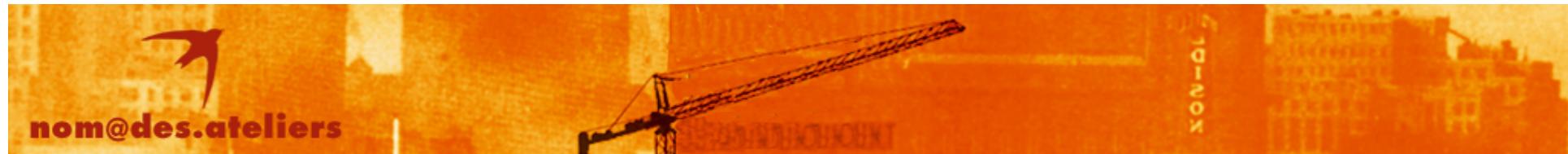
# Introduction à la programmation

Rolf Hauri

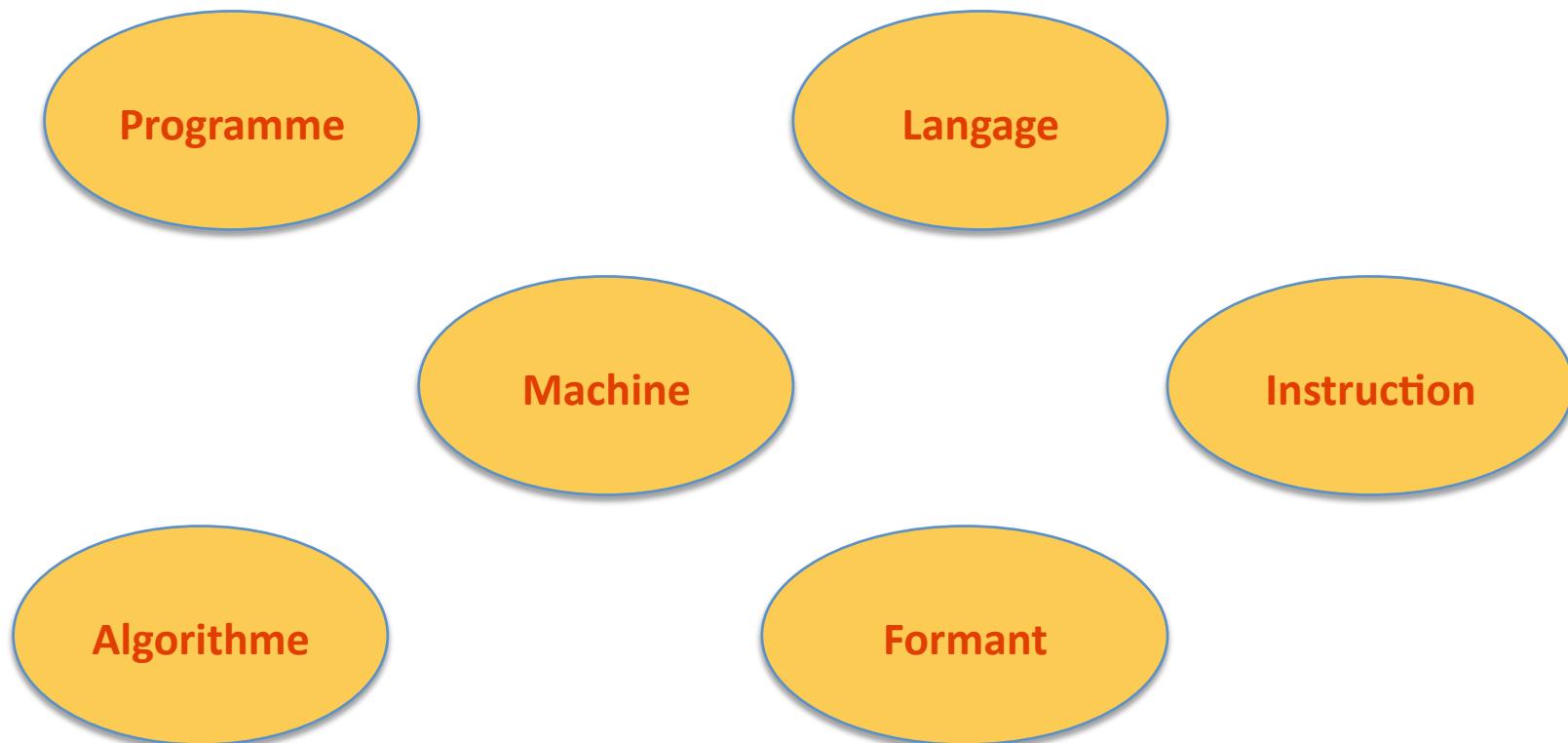


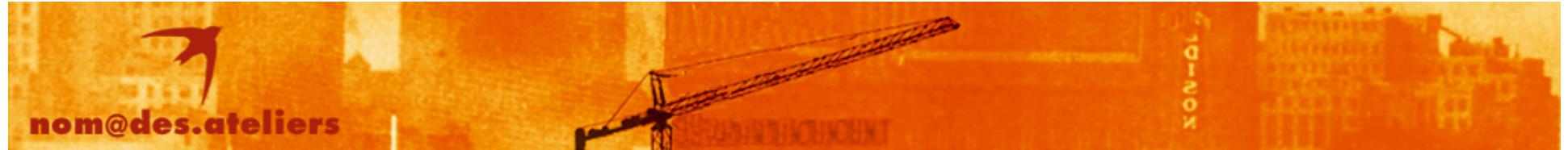
# Introduction à la programmation

- Programme
  - instruction, langage, machine, algorithme
- Formants de base
  - Organigrammes
- Fonctions, procédures
- Information, données
- Variables, constantes et paramètres
- Portée, visibilité, effets de bord
- Les classes, objets
  - Attributs, méthodes



# Introduction à la programmation





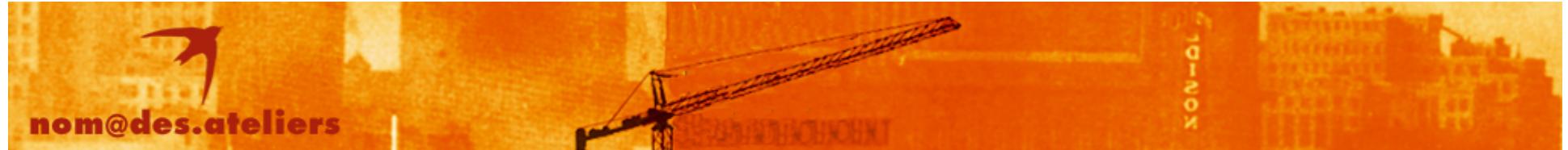
# Introduction à la programmation

- Programme
- *suite d'opérations pré-déterminées destinées à être exécutées de manière automatique par un appareil informatique en vue d'effectuer des travaux, des calculs arithmétiques ou logiques, ou simuler un déroulement*

*Wikipedia*

- *Instructions qu'utilise un ordinateur pour effectuer un traitement donné*

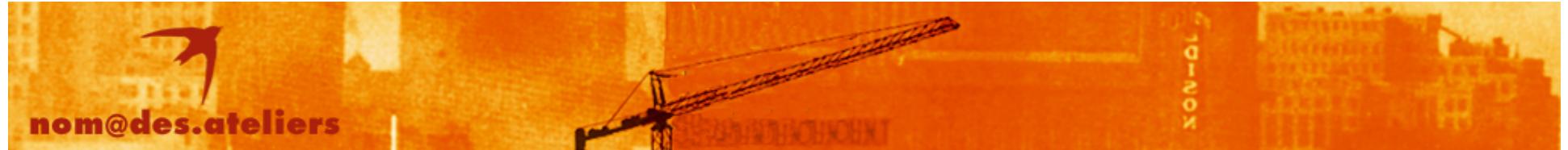
*Dictionnaire de l'internaute*



# Introduction à la programmation

- Un programme est une solution à un besoin
- La "suite des instructions" est définie pour atteindre un objectif de la meilleure\* façon possible

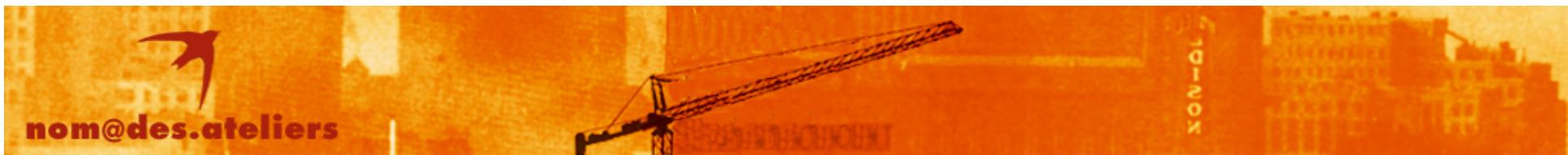
\*Meilleur est en termes de qualité (efficacité, efficience, maintenance, robustesse, ...)



# Introduction à la programmation

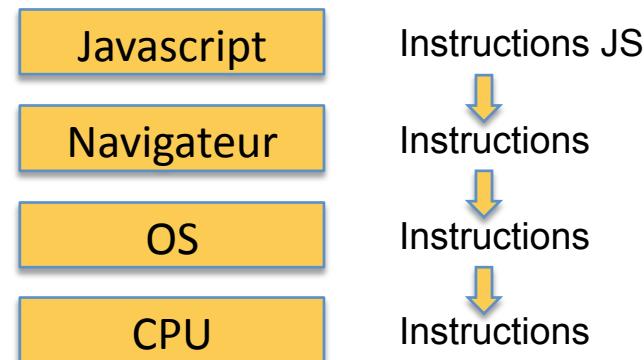
- Instruction
- *Une instruction est une forme d'information communiquée qui est à la fois une commande et une explication pour décrire l'action, le comportement, la méthode ou la tâche qui devra commencer, se terminer, être conduit, ou exécuté*

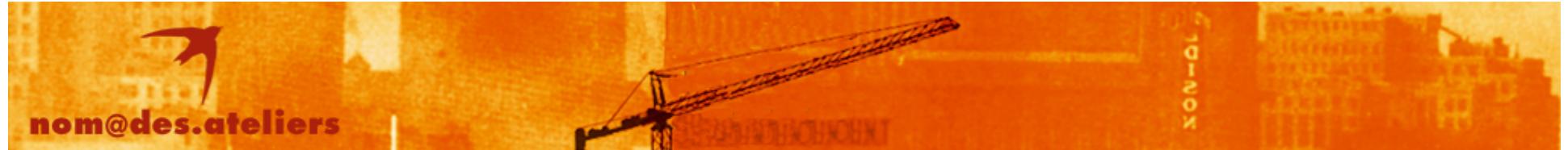
*Wikipedia*



# Introduction à la programmation

- Machine
- *Machine virtuelle qui exécute les instructions*
- Le navigateur interprète les instructions Javascript
- Niveau de machine virtuelle



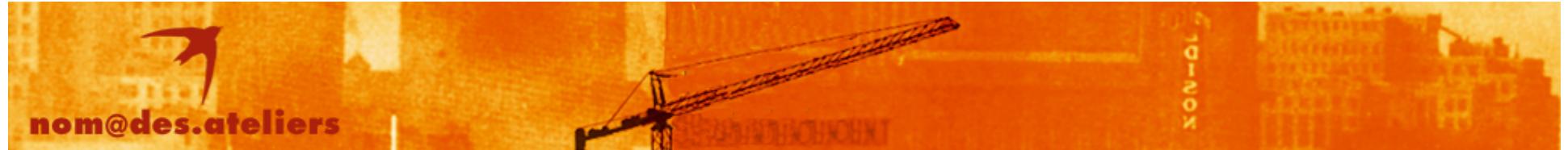


# Introduction à la programmation

- Langage
- *Le langage est un système de signes identifiés permettant une communication entre une ou plusieurs entités*

*Wikipedia*

- Langage formel vs langage naturel

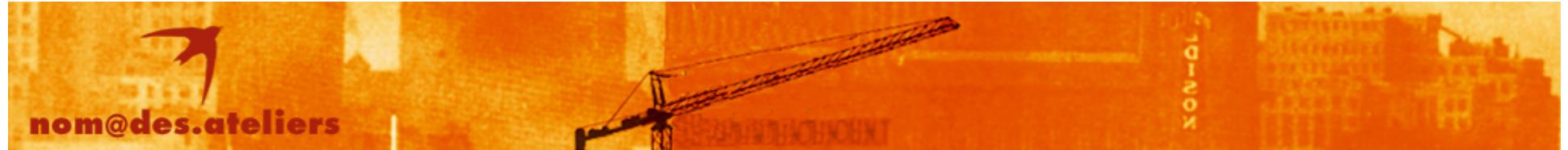


# Introduction à la programmation

- Langage formel
- *Langage qui utilise un ensemble de termes et de règles syntaxiques pour permettre de communiquer sans aucune ambiguïté*

*Wikipedia*

- Un langage formel fait abstraction de la sémantique

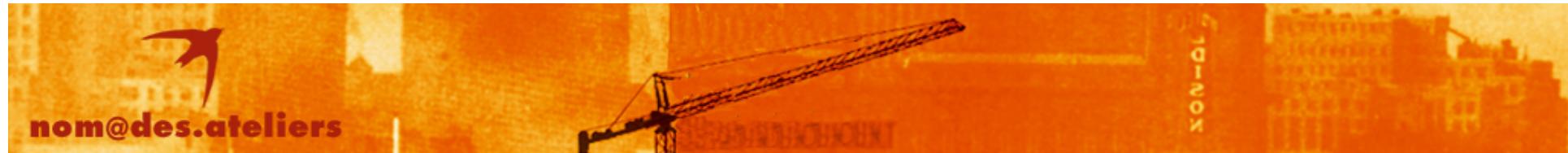


# Introduction à la programmation

- Langage formel
- *L'inconvénient est évident : ne pas connaître le sens de l'énoncé empêche de savoir quelles sont les transformations pertinentes et nuit à l'intuition du raisonnement. Ainsi, il est bon de savoir lire rapidement un énoncé en langage formel et de le traduire tout aussi rapidement en un ou plusieurs énoncés du langage naturel, plus significatif*

*Wikipedia*

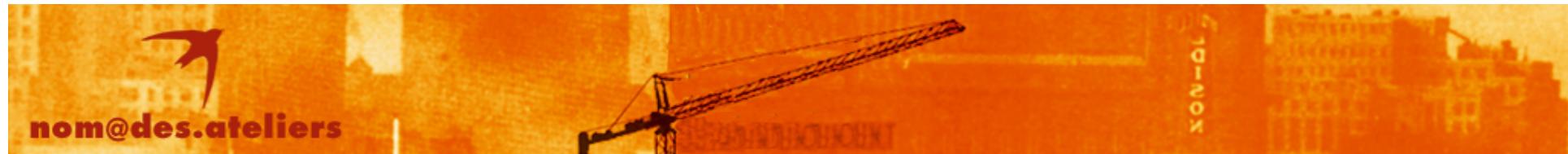
Note: La modélisation sera l'outil pour ces traductions, dans les deux sens



# Introduction à la programmation

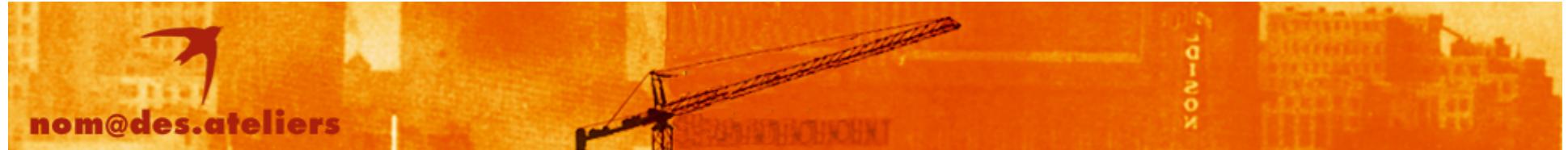
- Algorithmique
- *L'algorithmique est l'ensemble des règles et des techniques qui sont impliquées dans la définition et la conception d'algorithmes, c'est-à-dire de processus systématiques de résolution, par le calcul, d'un problème permettant de décrire les étapes vers le résultat. En d'autres termes, un algorithme est une suite finie et non ambiguë d'opérations permettant de donner la réponse à un problème.*

*Wikipedia*



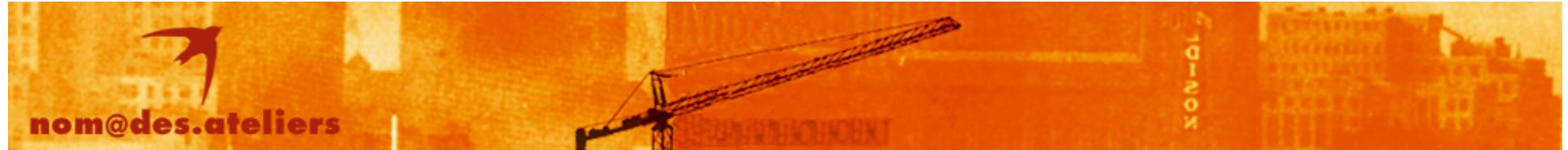
# Introduction à la programmation

- Programme
  - instruction, langage, machine, algorithme
- Formants de base
  - Organigrammes
- Fonctions, procédures
- Information, données
- Variables, constantes et paramètres
- Portée, visibilité, effets de bord
- Les classes, objets
  - Attributs, méthodes



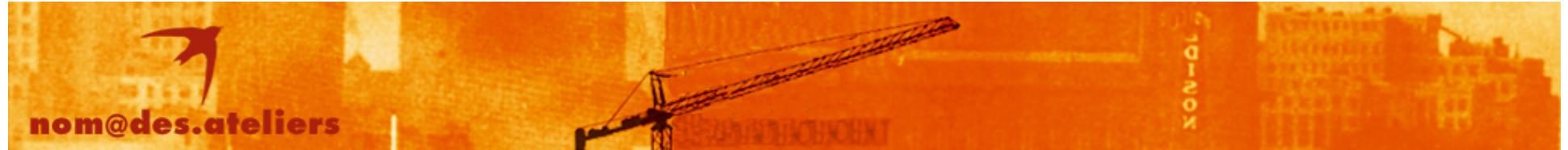
# Introduction à la programmation

- Les formants de base
  - Séquence
  - Alternative
  - Boucle
- Les trois sont suffisants pour tout programme
- Cas particuliers, la rupture



# Introduction à la programmation

- Séquence
  - Instruction 01
  - Instruction 02
  - ...
  - Instruction n



# Introduction à la programmation

- Alternative

- Si (test) alors

- Instruction  $a_1$

- ...

- Instruction  $a_n$

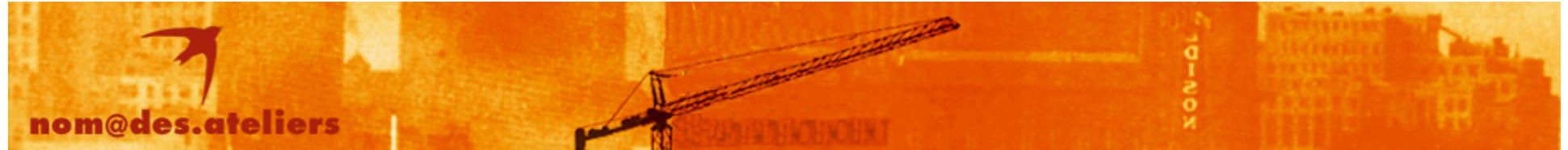
- Sinon

- Instruction  $b_1$

- ..

- Instruction  $b_m$

- Fin



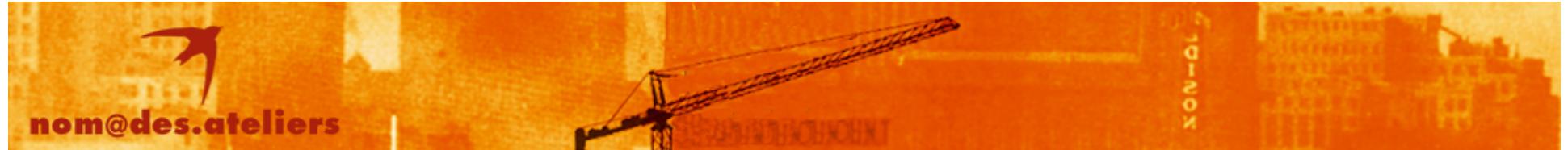
# Introduction à la programmation

- Alternative dans une version "mono-branche"

Si (test) alors

- Instruction  $a_1$
- ...

Fin



# Introduction à la programmation

- Alternative avec option de luxe

Si (test) alors

— ...

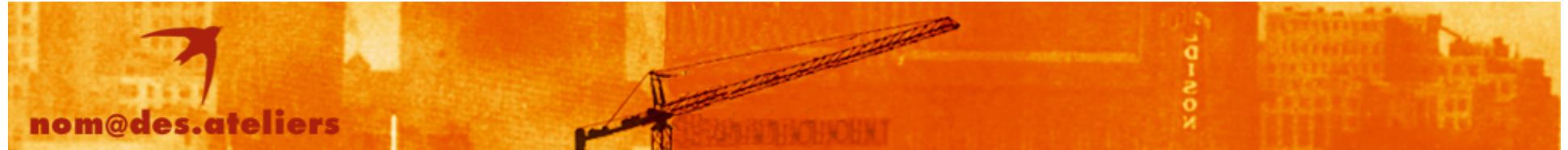
Sinon si (test) alors

— ..

Sinon

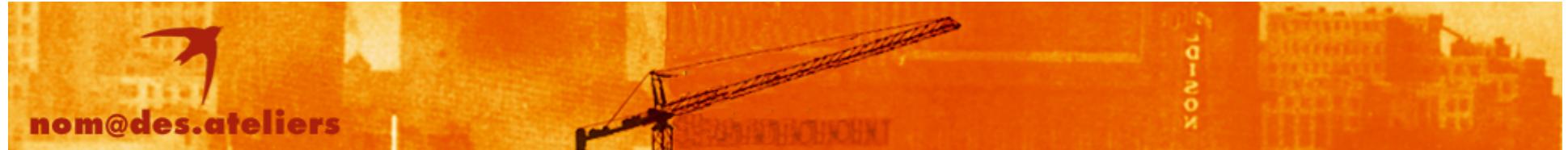
— ..

Fin



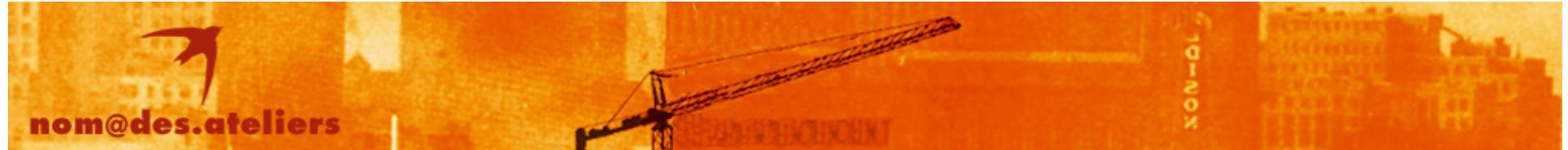
# Introduction à la programmation

- Test?
  - C'est une réponse à une question
- Valeurs possibles?
  - Vrai, faux
- Questions complexes?
  - Et, Ou, Pas, ()



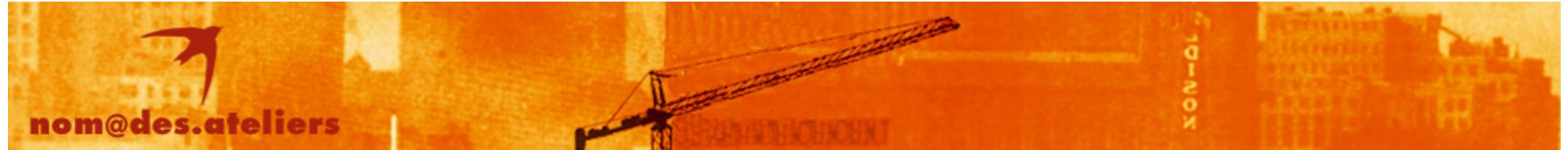
# Introduction à la programmation

- Boucles
- Le nombre d'itération est connu:
  - De 1 à n
    - Instructions
  - Fin



# Introduction à la programmation

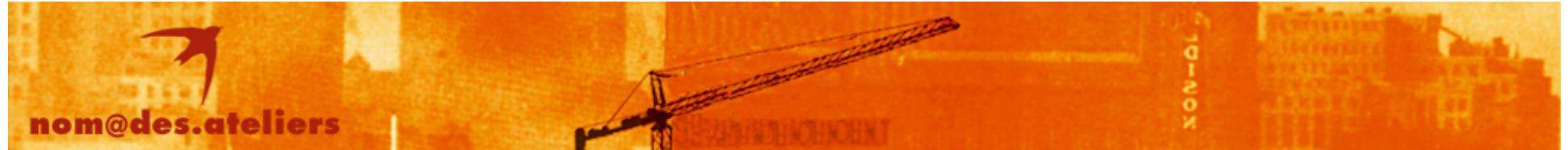
- Boucles
- Le nombre d'itération N'est PAS connu:
  - Tant que (test)
    - Instructions
  - Fin



# Introduction à la programmation

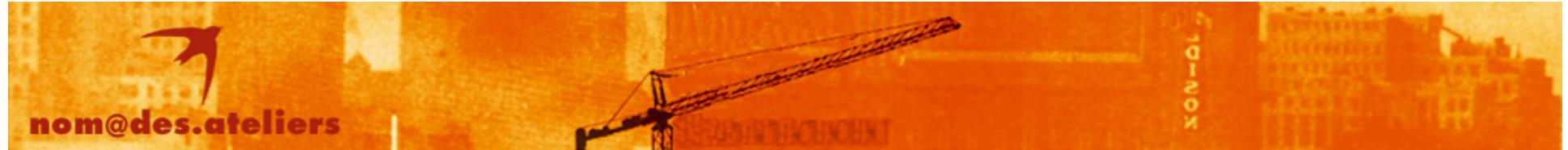
- Boucles options de luxe
  - Début
    - Instructions
  - Si (test) Refaire

Remarque: les instructions sont au moins faites une fois



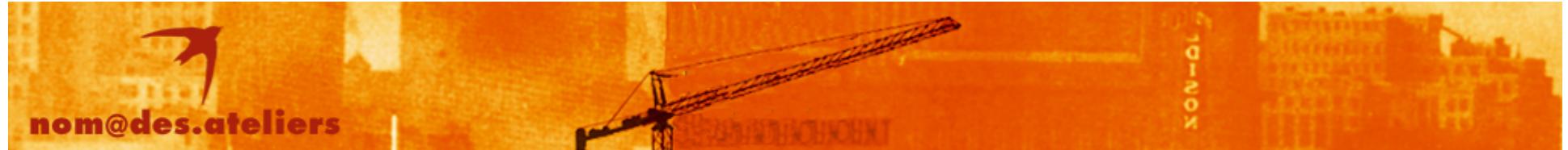
# Introduction à la programmation

- Cas particuliers (rupture)
- Arrêt du programme ou sous-programme en cours
- Option de luxe



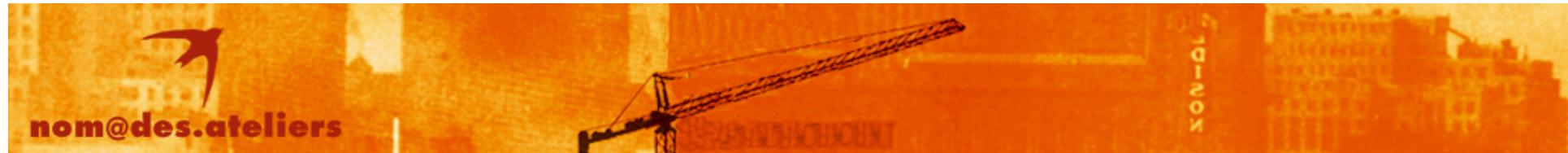
# Introduction à la programmation

- Programme
  - instruction, langage, machine, algorithme
- Formants de base
  - Organigrammes
- Fonctions, procédures
- Information, données
- Variables, constantes et paramètres
- Portée, visibilité, effets de bord
- Les classes, objets
  - Attributs, méthodes



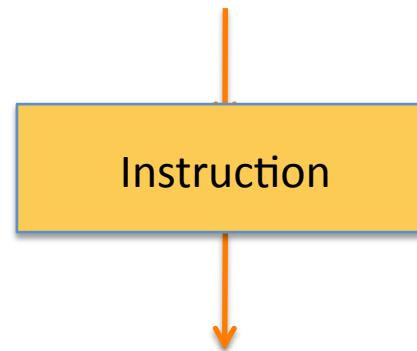
# Introduction à la programmation

- Organigrammes
  - Représentation visuelle des formants de base
  - Modélisation non textuelle



# Introduction à la programmation

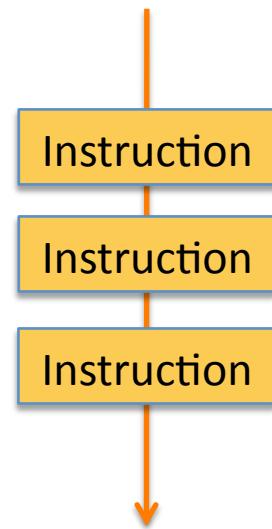
- Organigrammes – Eléments de base I
- Une instruction "mais" d'un certain niveau
- "Afficher résultats"
- Une entrée, une sortie





# Introduction à la programmation

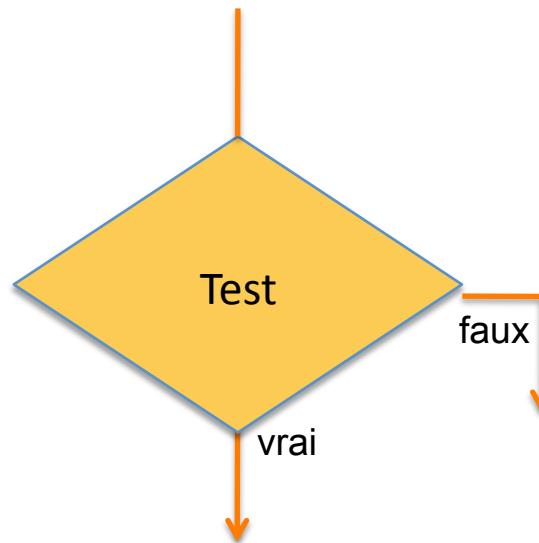
- Pour représenter le formant "séquence":





# Introduction à la programmation

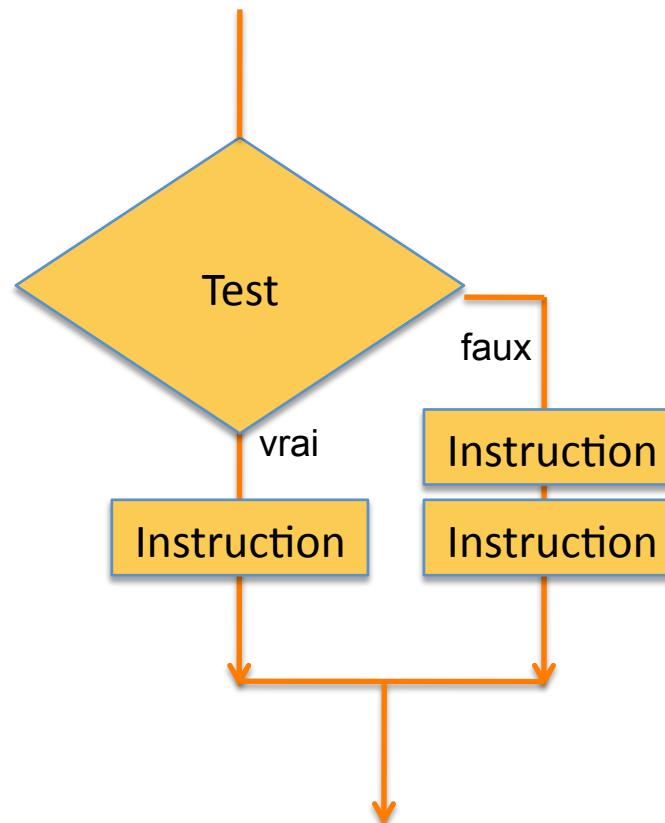
- Organigrammes – Eléments de base II
- L'alternative
- Une entrée
- Deux sorties





# Introduction à la programmation

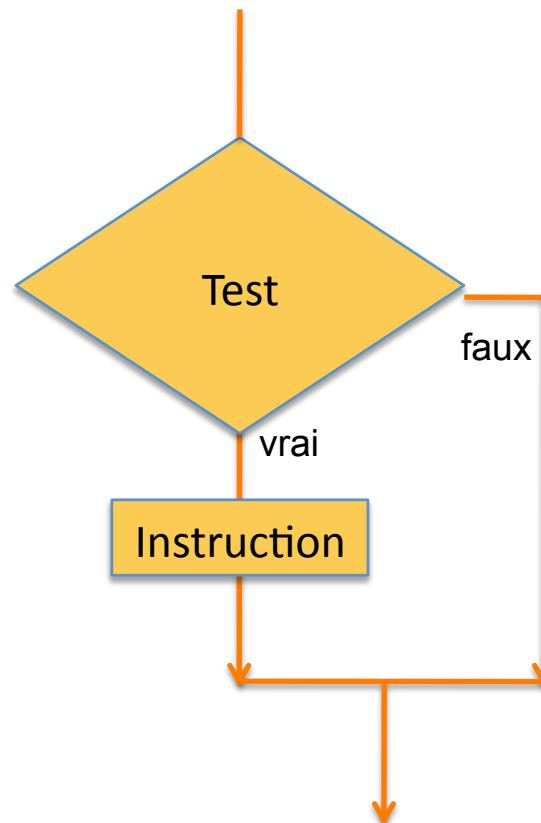
- Pour représenter le formant "alternative":





# Introduction à la programmation

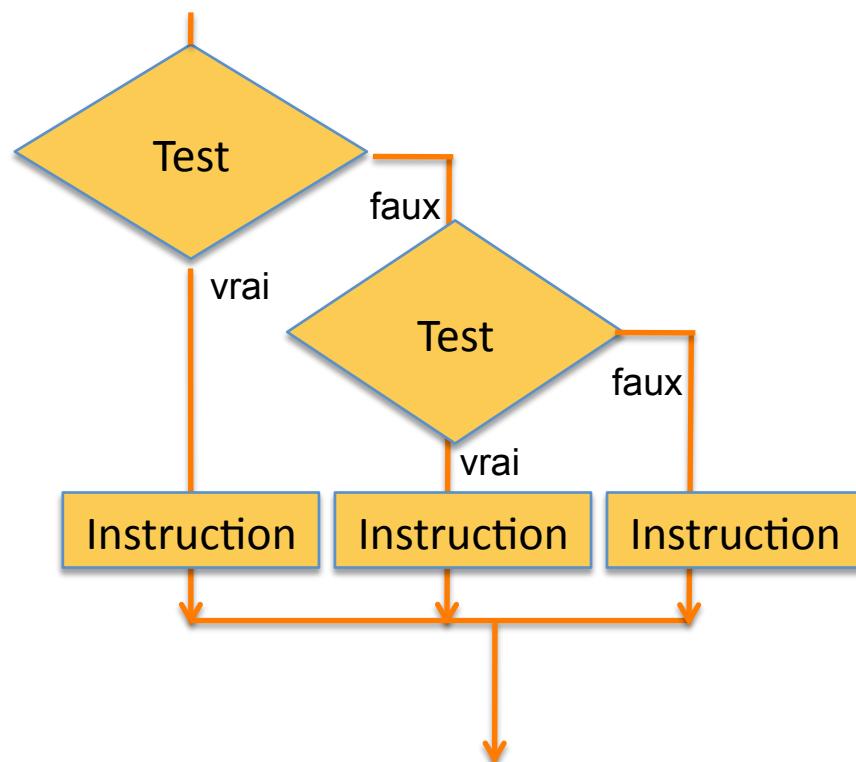
- Pour représenter le formant mono-branche:





# Introduction à la programmation

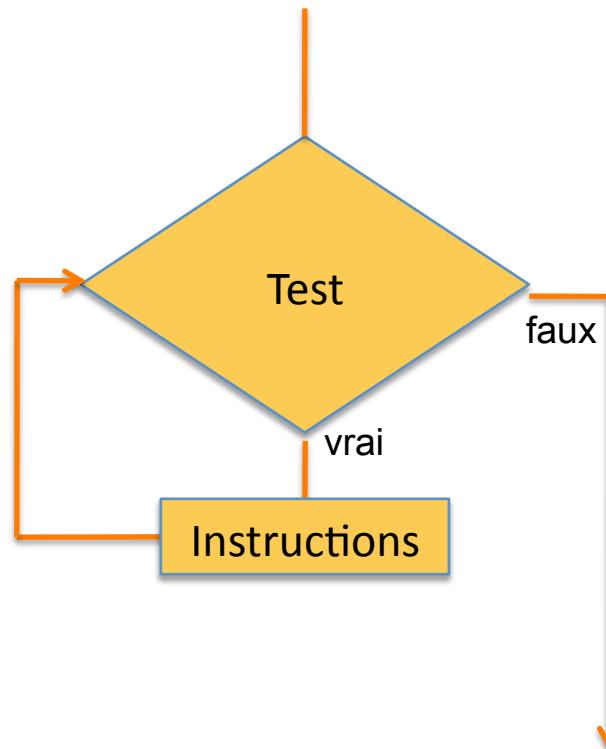
- Pour représenter le formant "alternative" avec plus de deux sorties (imbrication ~ sinon si):





# Introduction à la programmation

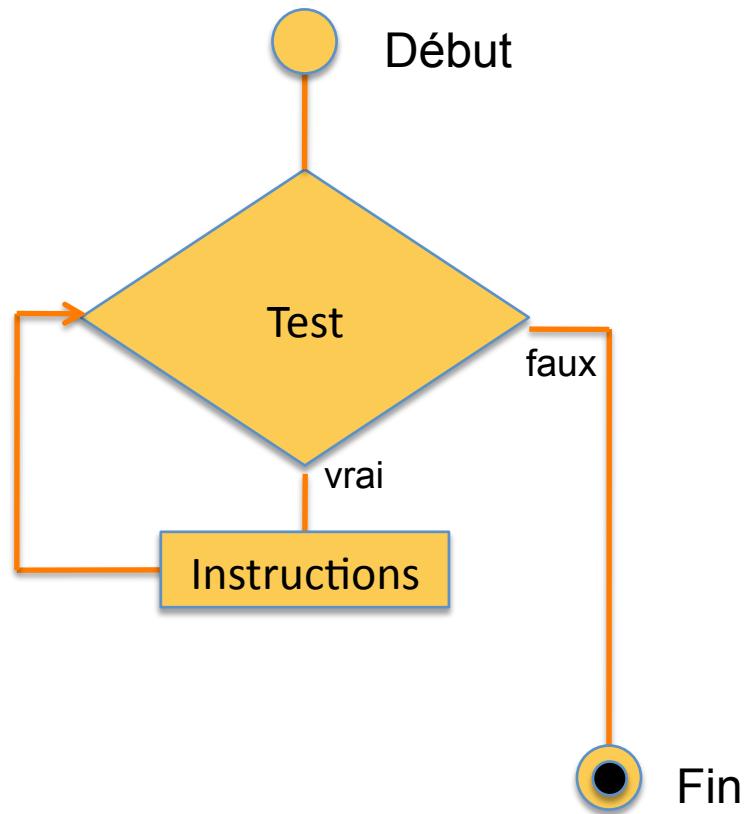
- Organigrammes – Eléments de base III
- La boucle
- Une entrée
- Une sortie

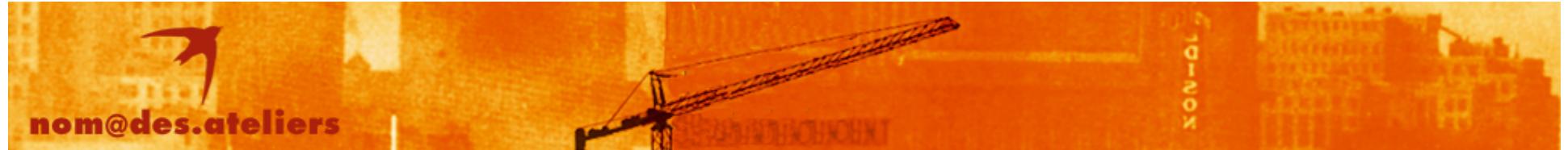




# Introduction à la programmation

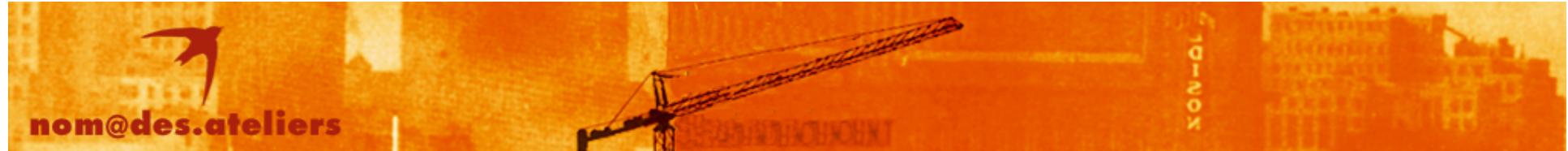
- Organigrammes – Début et fin





# Introduction à la programmation

- Programme
  - instruction, langage, machine, algorithme
- Formants de base
  - Organigrammes
- Fonctions, procédures
- Information, données
- Variables, constantes et paramètres
- Portée, visibilité, effets de bord
- Les classes, objets
  - Attributs, méthodes



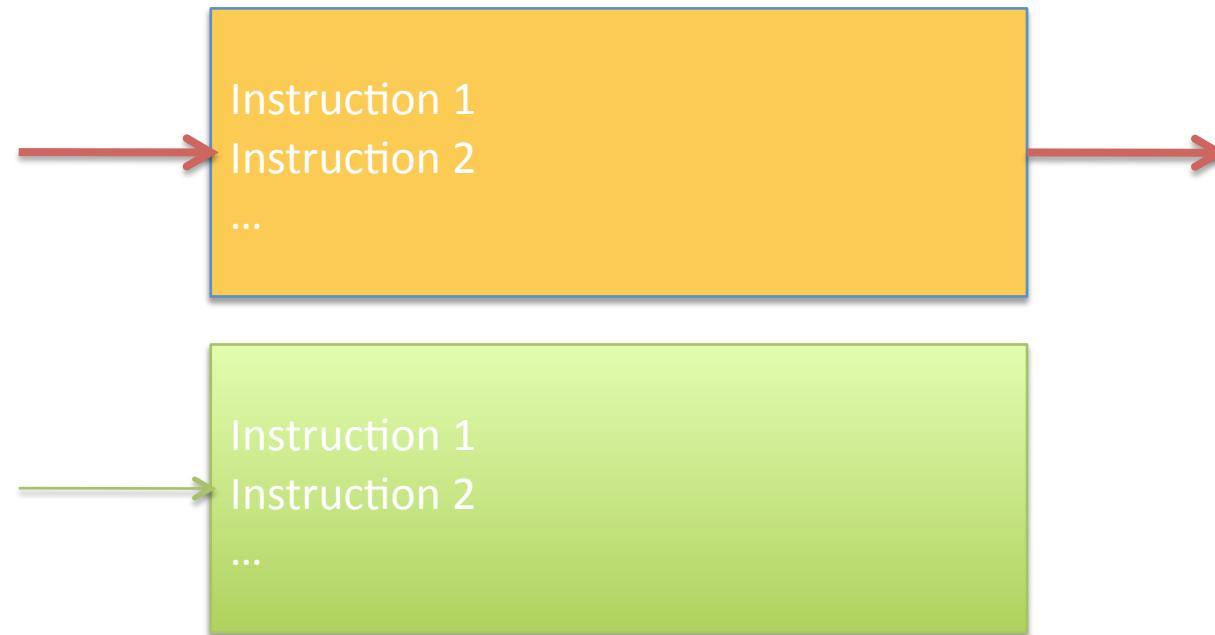
# Introduction à la programmation

- **Fonction et procédure:**
- Sous ensemble d'instructions réalisant un objectif
- Avantages
  - Réutilisation
  - Organisation du code
  - Décomposition par niveau
  - Librairies



# Introduction à la programmation

- Fonction et procédure:



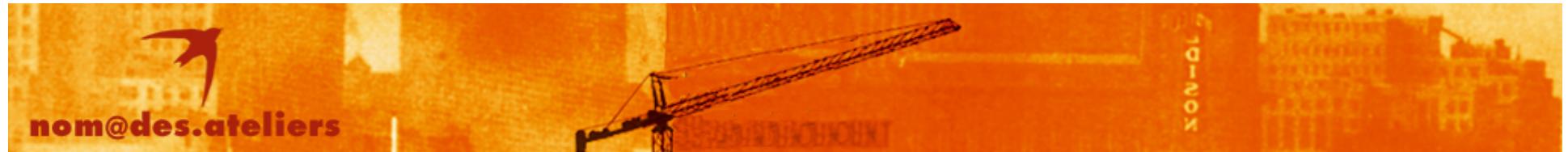


# Introduction à la programmation

- Fonction et procédure:

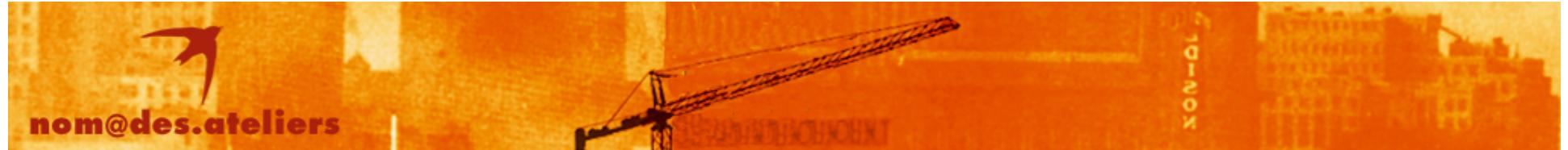


- afficher(arrondir(calculer(x)))



# Introduction à la programmation

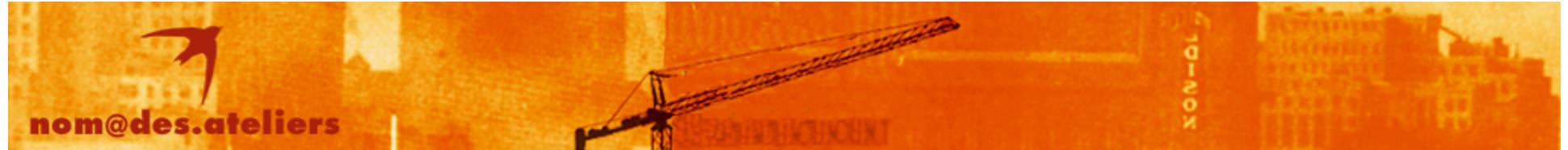
- Programme
  - instruction, langage, machine, algorithme
- Formants de base
  - Organigrammes
- Fonctions, procédures
- Information, données
- Variables, constantes et paramètres
- Portée, visibilité, effets de bord
- Les classes, objets
  - Attributs, méthodes



# Introduction à la programmation

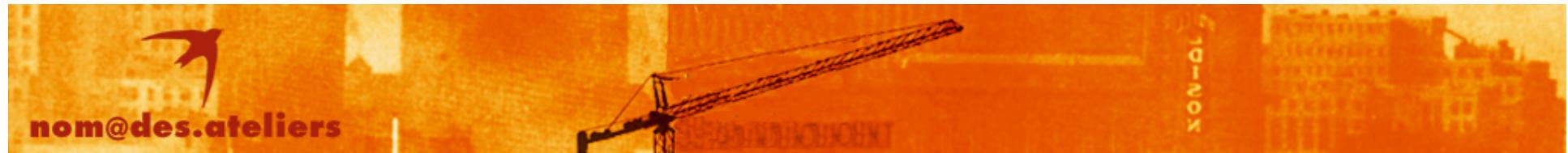
- Représentation de l'information
- Du monde analogique au numérique
- Du monde infini au monde limité





# Introduction à la programmation

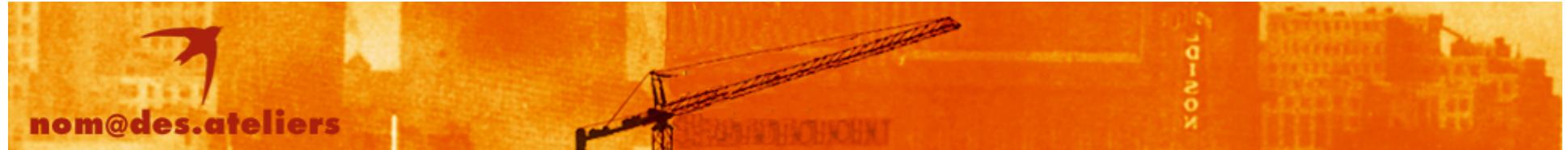
- Valeur numérique
  - Entière, flottante, signée, ...
- Chaine de caractère
  - Un caractère, des caractères
- Date
  - heure
- Booléen
- Ensemble d'octets (bytes)



# Introduction à la programmation

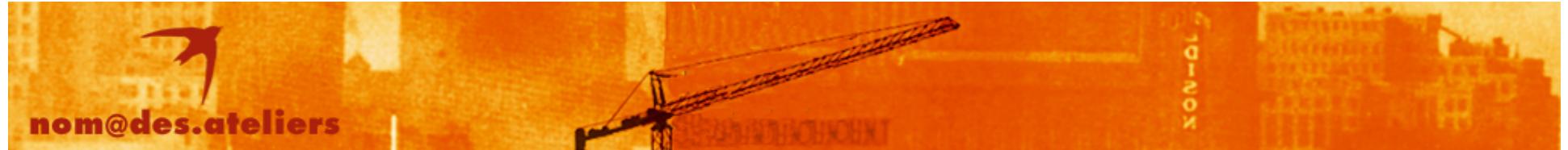
- Représentation de l'information

Information	Type ?	Exemple ?
Nom		
Rue		
Téléphone		
Numéro postal		
Année de naissance		
Date d'inscription		
Est membre?		
Remarque		
Photo		



# Introduction à la programmation

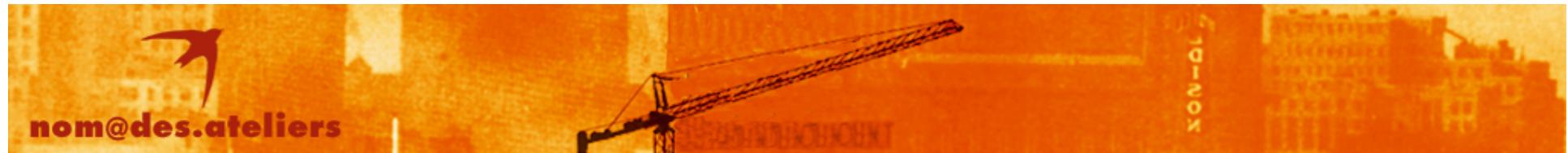
- Les données sont utilisées et traitées par les instructions
- En entrée, au départ : paramètre, constante
- Lors du traitement : variable
- En sortie : résultat, variable



# Introduction à la programmation

- **Constante:**
- Identificateur associé à une valeur fixe. Il n'est pas possible de la modifier
- Avantages
  - Manipulation d'un identificateur (PI, TVA) plutôt qu'une valeur
  - Lorsque la définition de la constante change, la modification est faite dans tout le programme

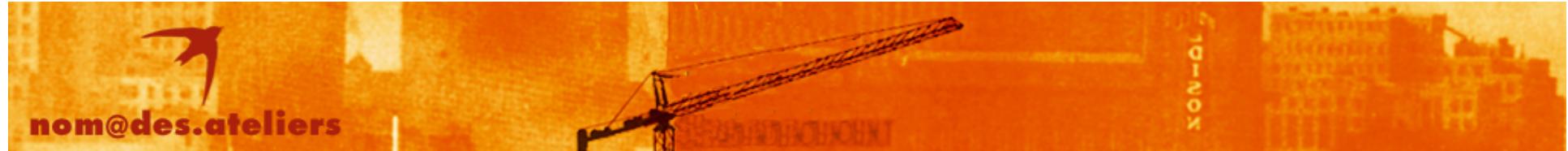
Note: une constante est remplacée par sa valeur lors de la compilation



# Introduction à la programmation

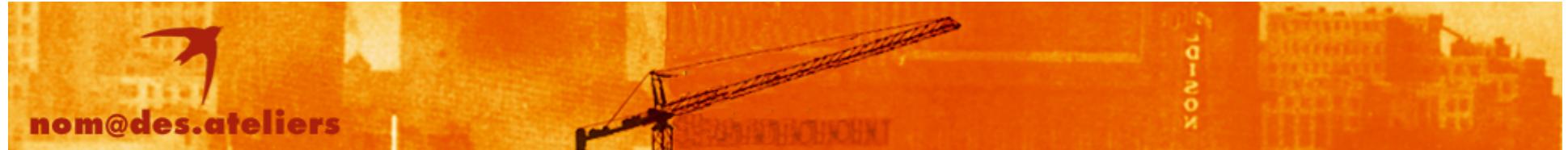
- **Variable:**
- Une variable est un identificateur associé à un espace mémoire, contenant une valeur\*
- Avantages
  - Manipulation d'un identificateur (taux, nom) plutôt qu'une valeur
  - Espace temporaire de stockage entre les instructions

\* Ou un ensemble de valeurs



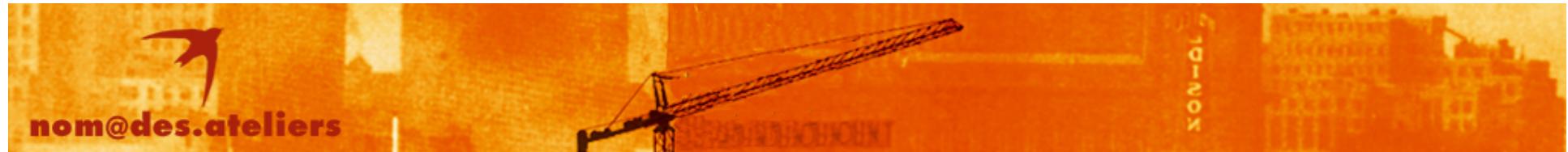
# Introduction à la programmation

- **Un ensemble de valeurs:**
- Les informations peuvent être stockées/  
manipulées dans des structures
  - Tableau (liste, 2d, 3d, ...)
  - Pile (LIFO, FIFO, ...)
  - Arbre
  - Type (Client, Véhicule, Compte)
  - ...
  - Objets (voir POO)



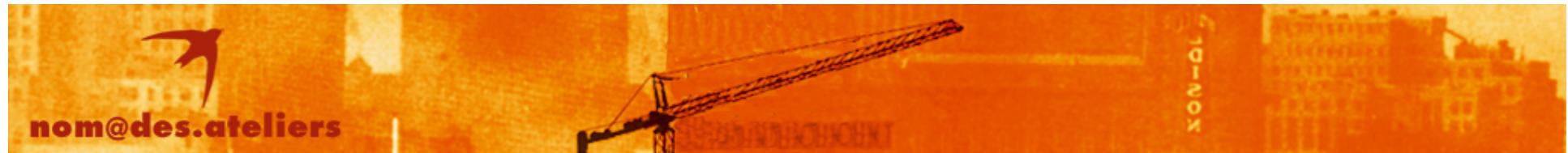
# Introduction à la programmation

- **Paramètre:**
- Valeur fournie à un programme/fonction/procédure
- En "entrée" en général, la valeur est une copie
  - L'original est préservé
- Possible en "sortie" possible
  - Le paramètre est "l'original" et peut être modifié



# Introduction à la programmation

- Programme
  - instruction, langage, machine, algorithme
- Formants de base
  - Organigrammes
- Fonctions, procédures
- Information, données
- Variables, constantes et paramètres
- Portée, visibilité, effets de bord
- Les classes, objets
  - Attributs, méthodes



# Introduction à la programmation

- Programme
  - instruction, langage, machine, algorithme
- Formants de base
  - Organigrammes
- Fonctions, procédures
- Information, données
- Variables, constantes et paramètres
- Portée, visibilité, effets de bord
- Les classes, objets
  - Attributs, méthodes