

Introduction à la programmation

WebProgrammer - JS

Antonio Pisanello



Introduction à la programmation

Sommaire

- Programmer
 - Programme, instruction, modélisations, structure
 - Les phases générique d'un programme
 - Fonctions, procédures
 - Efficience d'un programme



Introduction à la programmation

Sommaire

- Informations manipulés
 - Variables, constante, paramètres
 - Portée, visibilité, effets de bords
- Les classes, objets
 - Attributs, méthodes



Introduction à la programmation

Programme / algorithme

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

Wikipedia



Introduction à la programmation

Programme / algorithme

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

Wikipedia

- Suite d'instructions qu'utilise un ordinateur pour effectuer un traitement donné



Introduction à la programmation

Programme / Algorithme

- Un programme : une solution à un besoin, pour atteindre un objectif, de la meilleure* façon possible



Introduction à la programmation

Programme / Algorithme

- Un programme : une solution à un besoin, pour atteindre un objectif, de la meilleure* façon possible
- *Meilleure en terme 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 exécutée

Wikipedia



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 exécutée

Wikipedia

- Un ordre, une commande qu'on l'on demande a l'ordinateur d'exécuter



Introduction à la programmation

Exemple de programme

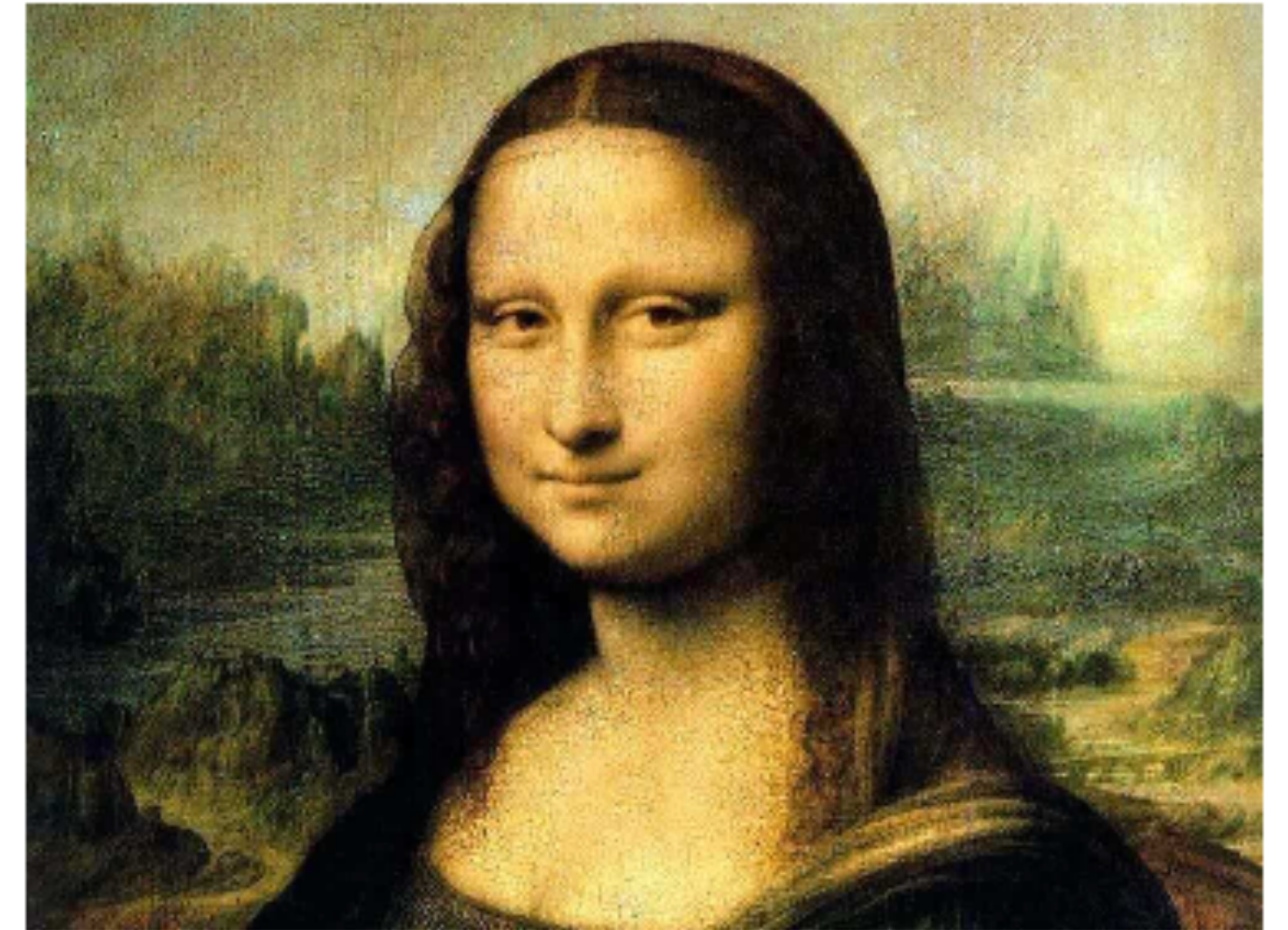
- Diviser la taille d'une image par deux



Introduction à la programmation

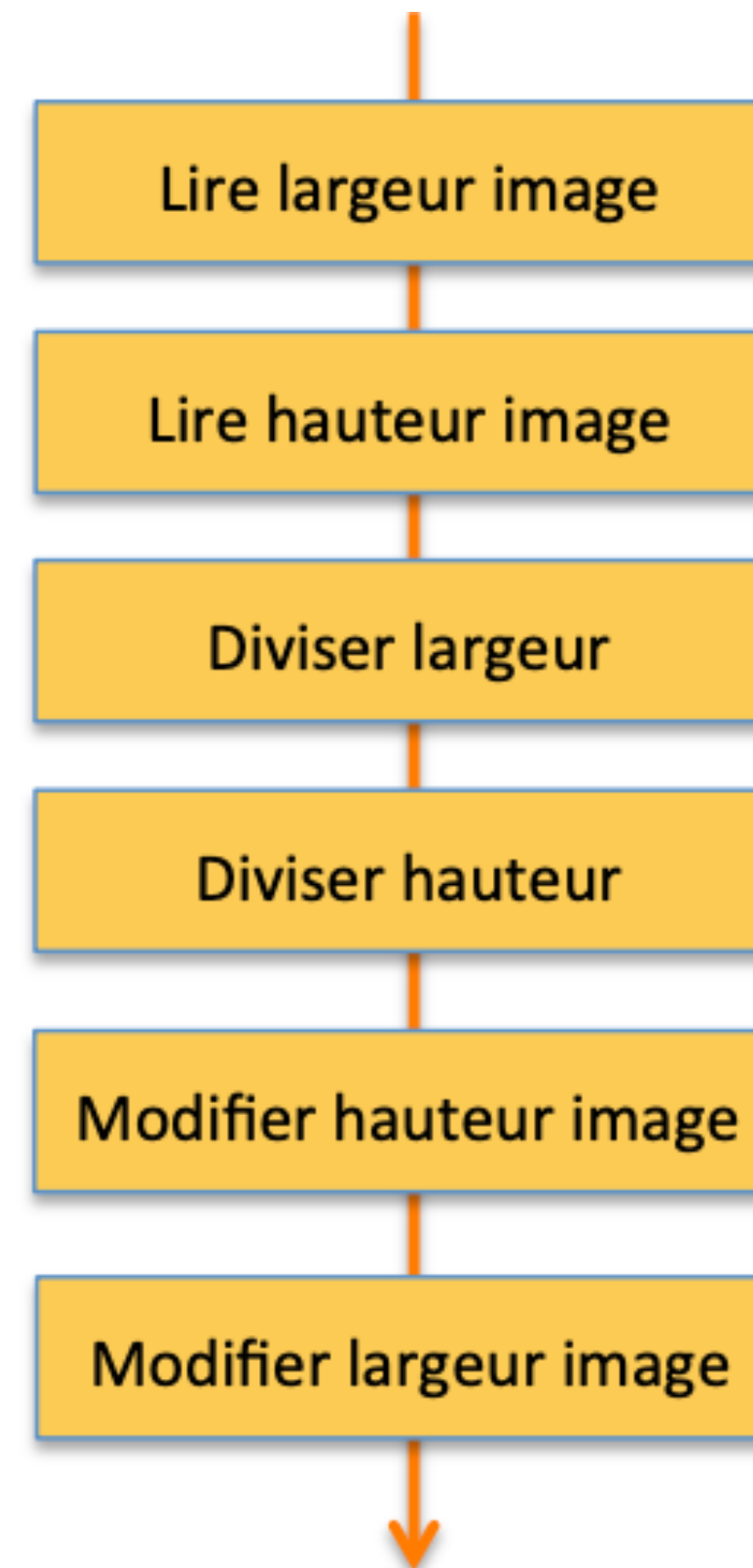
Exemple d'instruction

- Lire la largeur d'une image
- Lire la hauteur d'une image
- Diviser la largeur par 2
- Diviser la hauteur par 2
- Modifier la largeur de l'image
- Modifier la hauteur de l'image⁴



Introduction à la programmation

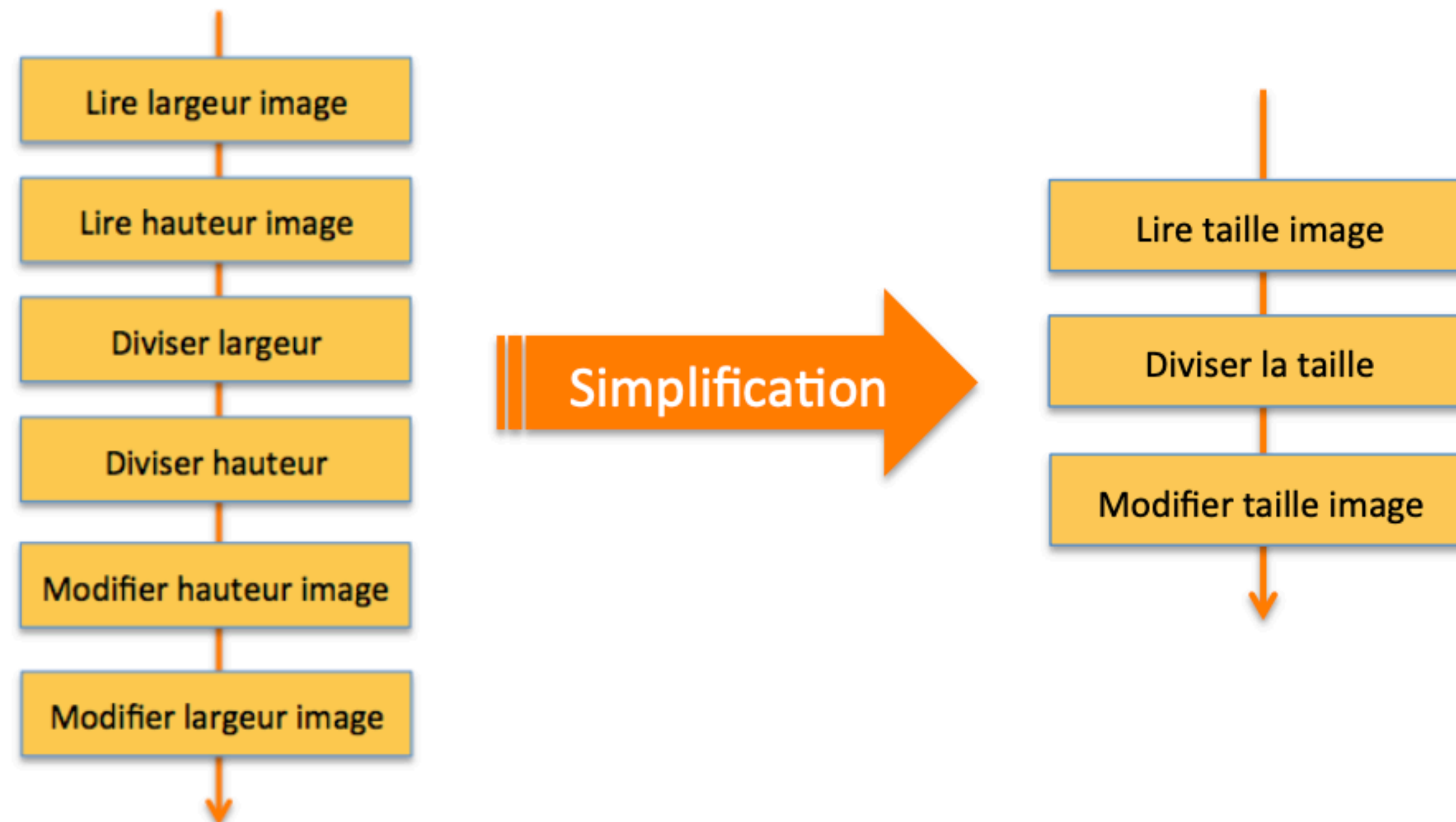
Représentation des instructions (modèle)



Introduction à la programmation

Représentation des instructions (fonctions)

- Les instructions sont souvent modélisées à un niveau “plus élevé” afin de simplifier le modèle:



Introduction à la programmation

Modélisation

- La réalisation d'un programme commence par la modélisation
 1. Objectif du programme
 2. Etapes principales
 - Niveau plus élevé -> pour le développeur
 3. Etapes détaillées de chaque étapes principales
 - Niveau détaillé (instruction) -> pour le programme



Introduction à la programmation

Modélisation

1. Objectif du programme: réduire la taille de l'image

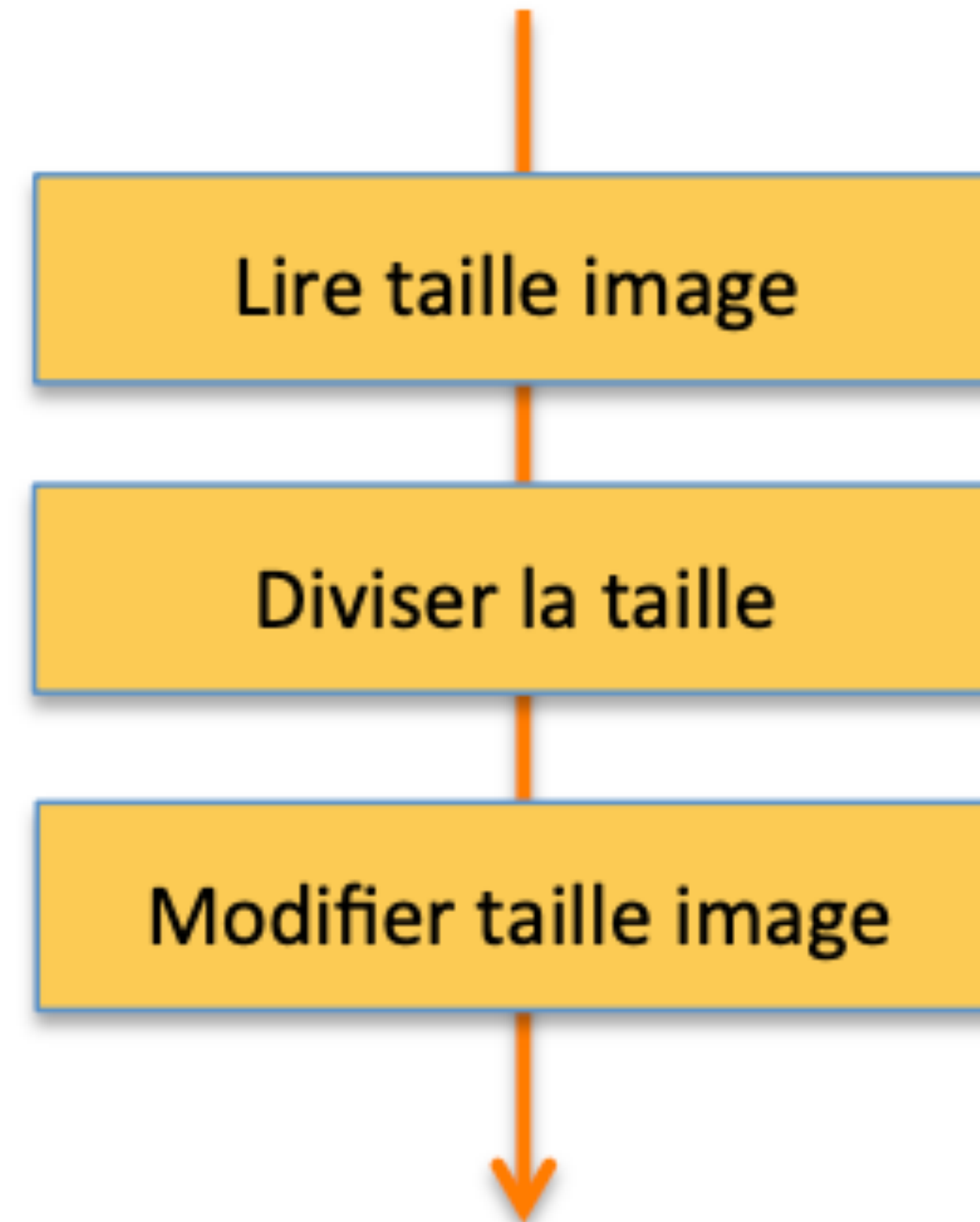


Introduction à la programmation

Modélisation

2. Étapes principales

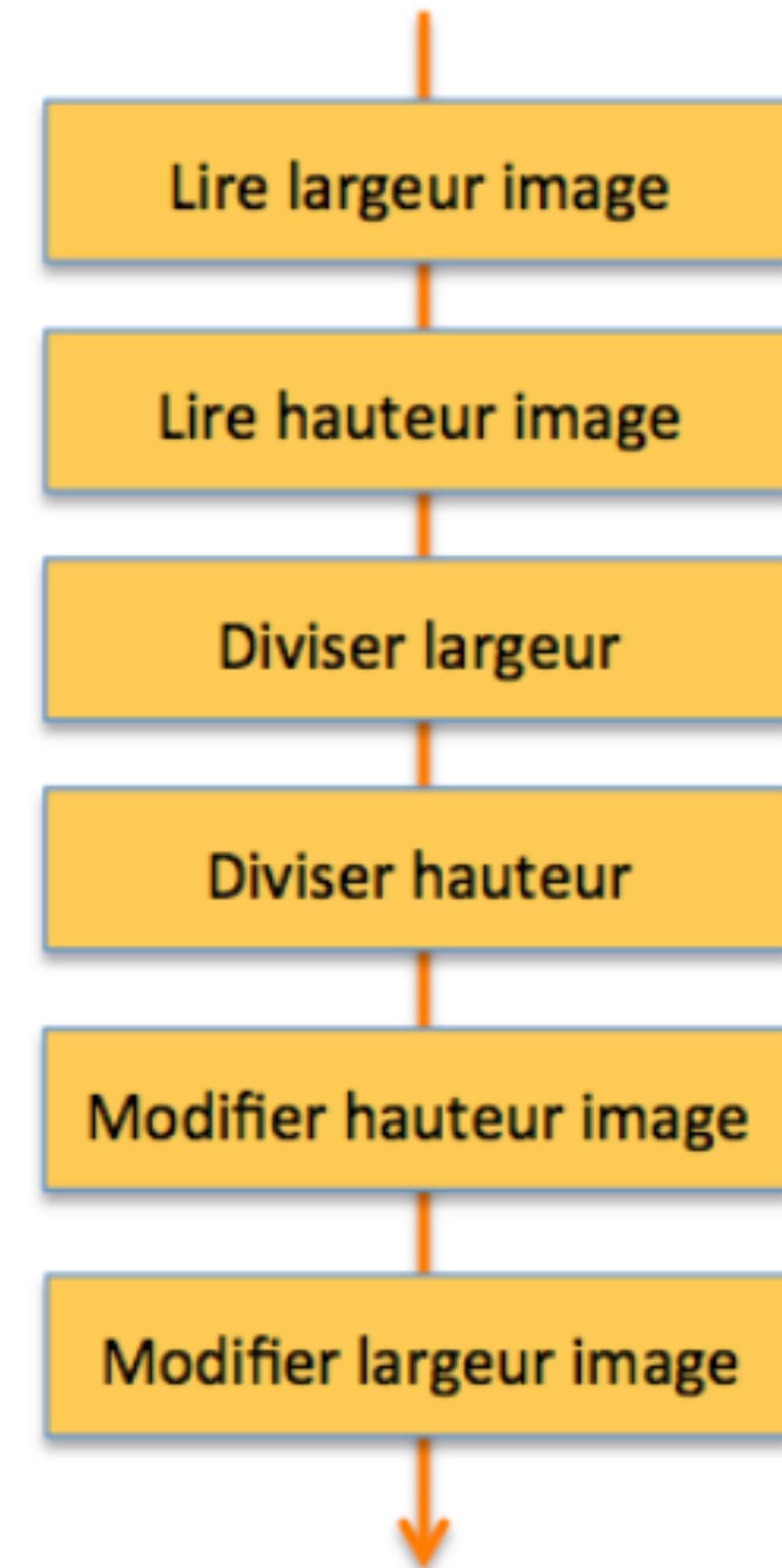
- Niveau plus élevé (pour le développeur)



Introduction à la programmation

Modélisation

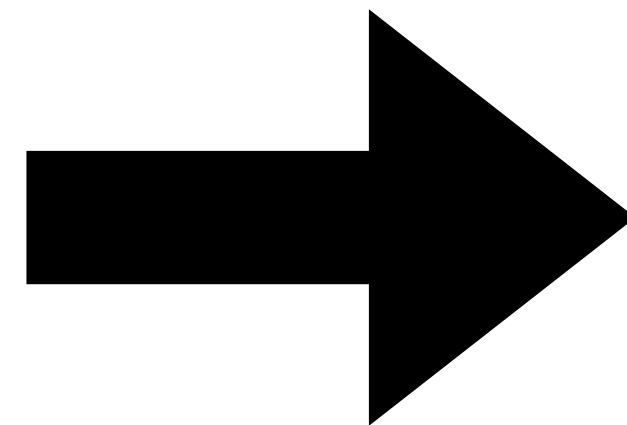
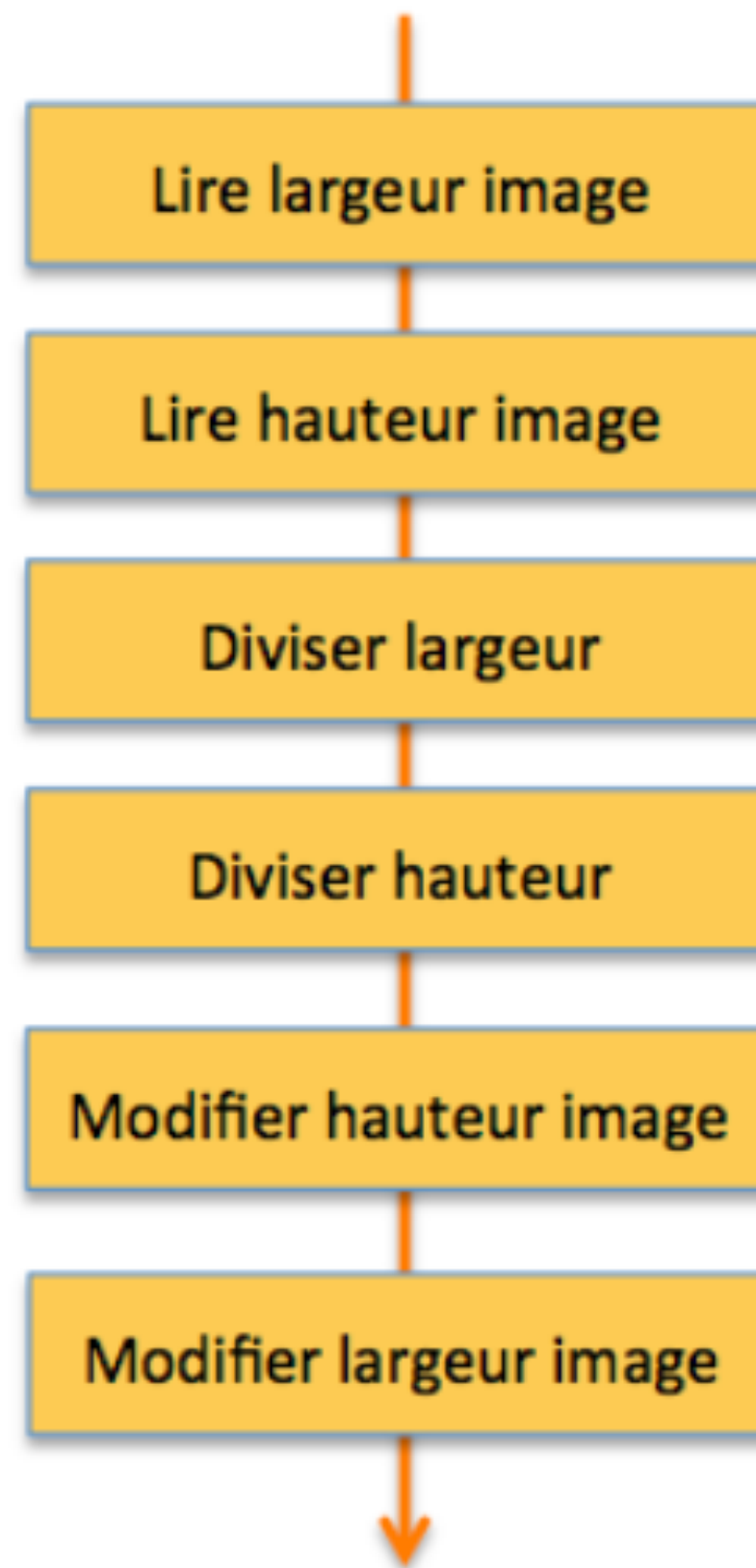
3. Étapes détaillées de chaque étape principale
- Niveau détaillé (instruction) -> pour le programme



Introduction à la programmation

Le code

- Après la modélisation, les étapes détaillées sont traduites en code



```
var largeur = myImage.style.width
var hauteur = myImage.style.height
largeur = largeur/2
hauteur = hauteur/2
myImage.style.width = largeur
myImage.style.height = hauteur
```



Introduction à la programmation

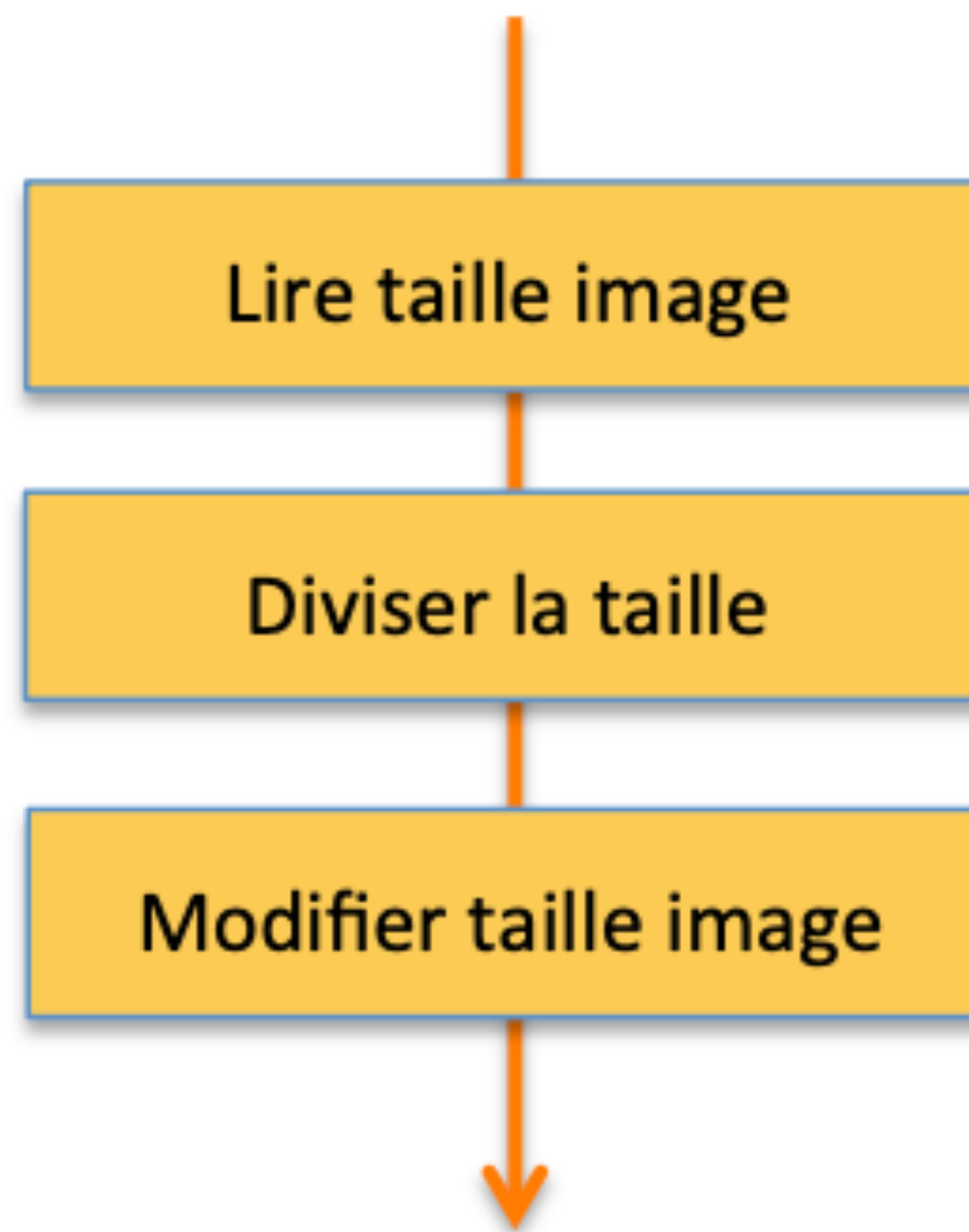
Strucutres de bases

1. La sequence : les instruction se suivent
2. La condition : les instruction sont faites selon un test (if; else)
3. La boucle : les instruction sont répétées (for; while)



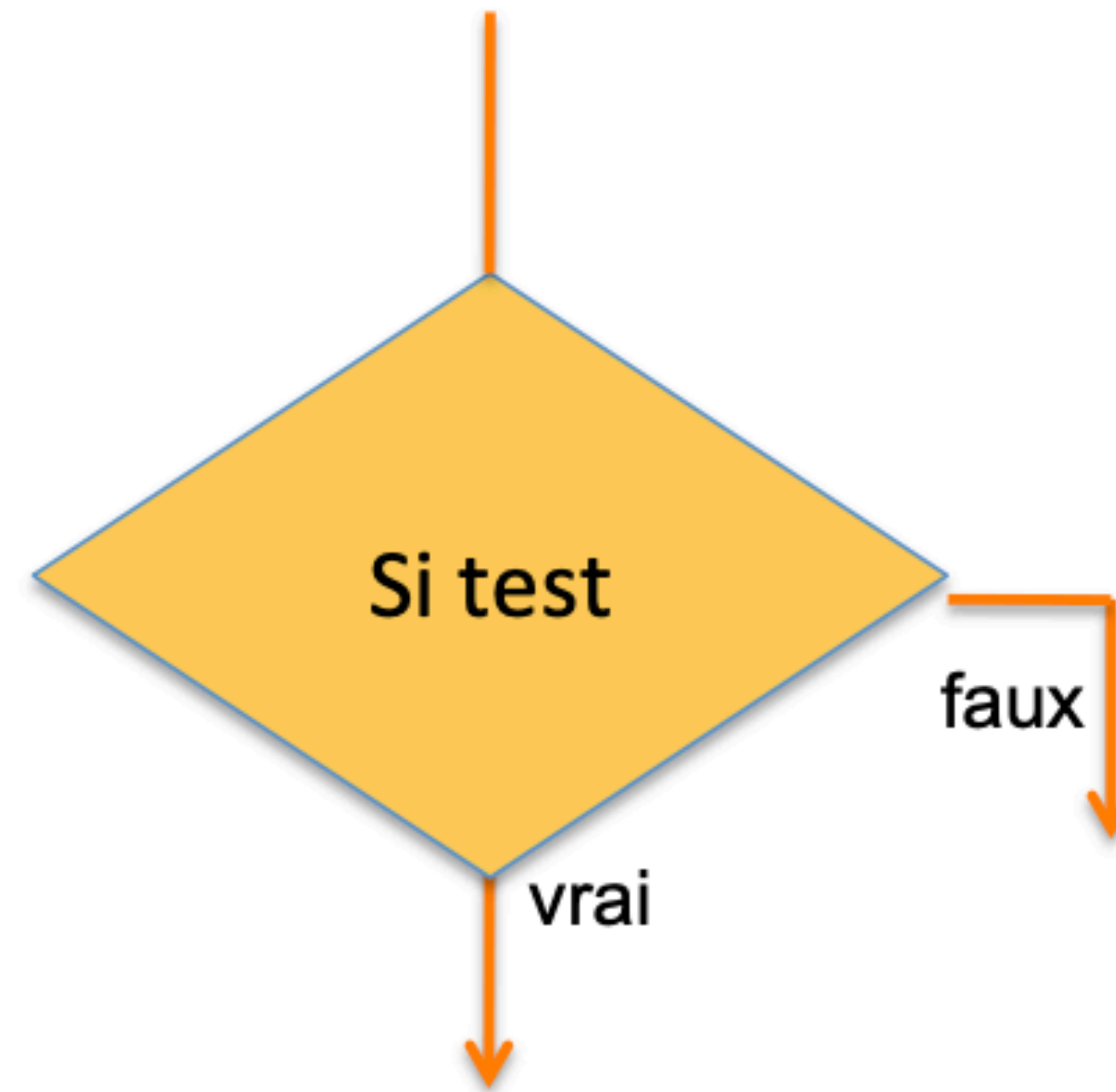
Introduction à la programmation

La sequence



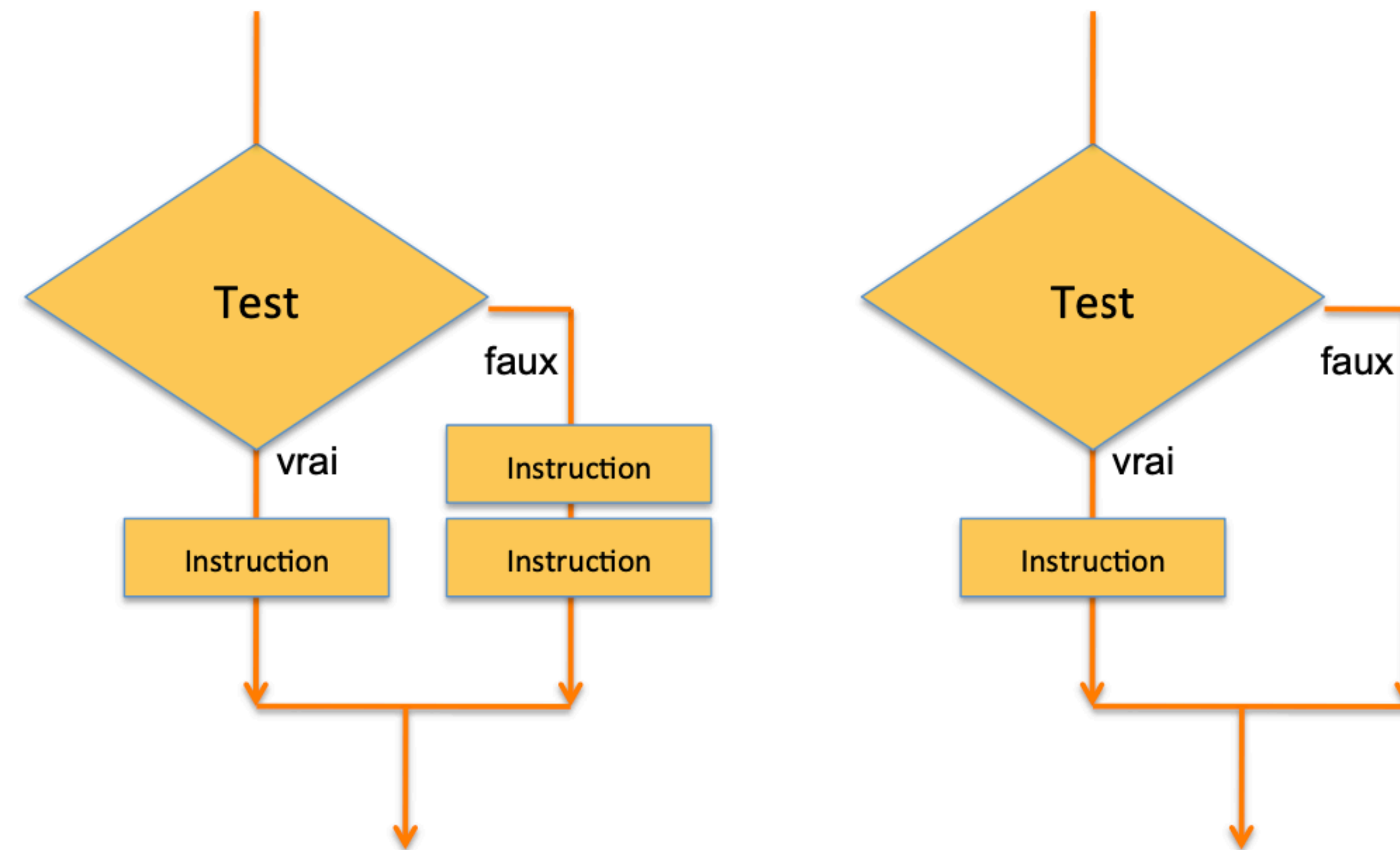
Introduction à la programmation

La condition



Introduction à la programmation

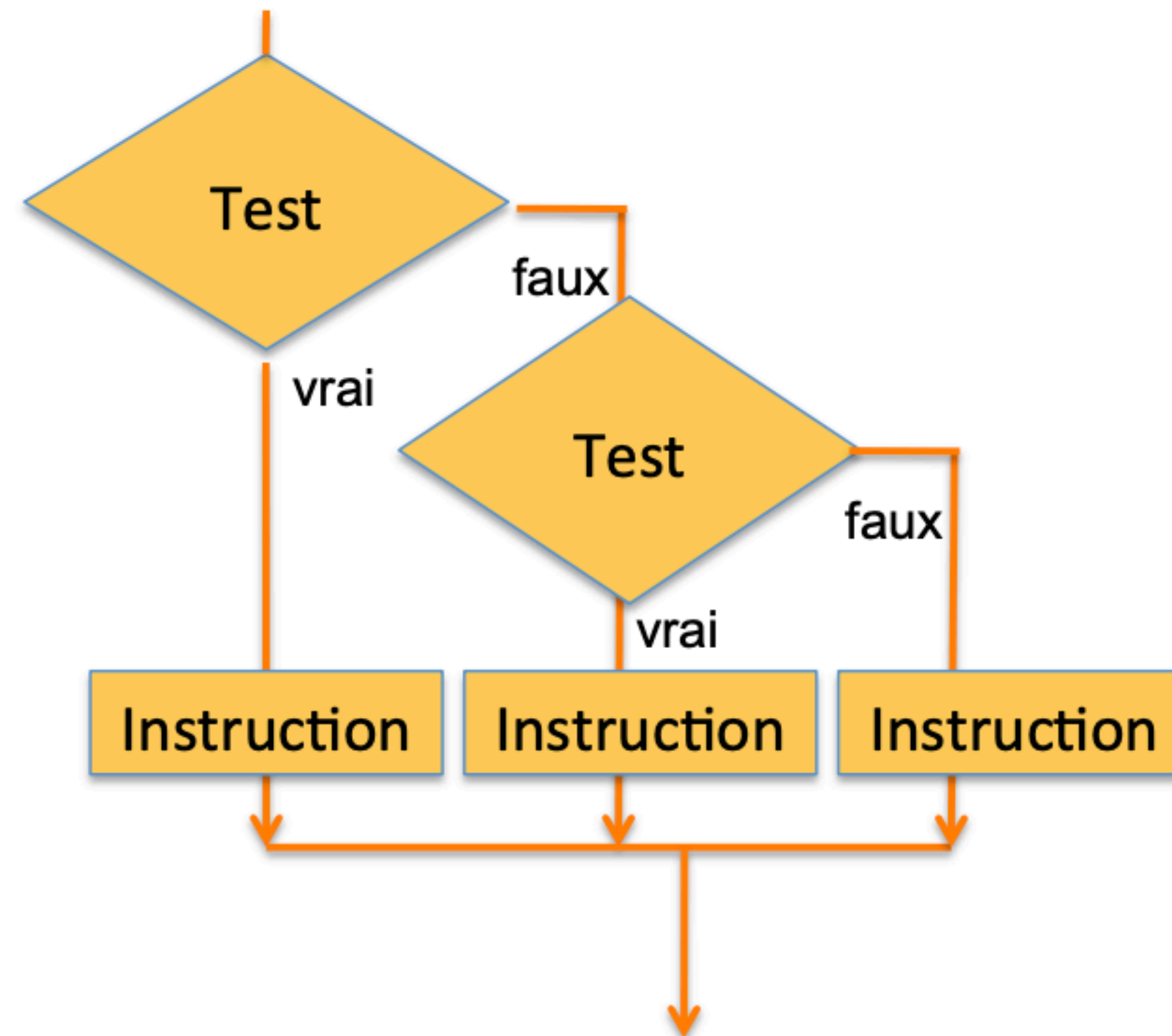
La condition exemple



Introduction à la programmation

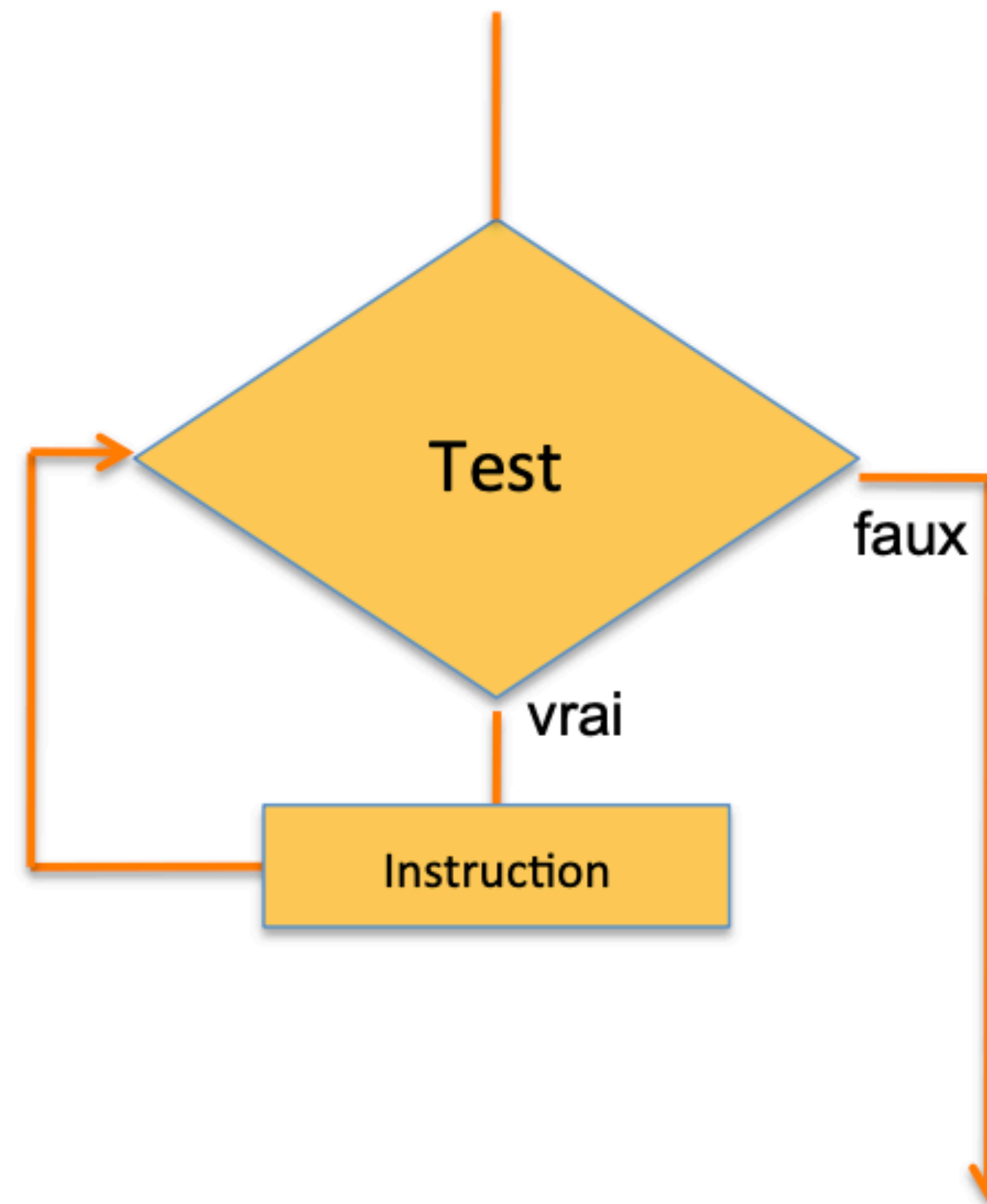
La condition exemple

La condition n'a que deux "sorties", si besoin de. Plus, enchainement de tes:



Introduction à la programmation

La boucle



Introduction à la programmation

Fonctions et procédures

- Sous ensemble d'instruction réalisant un sous-objectif
- Avantages
 - Réutilisation
 - Organisation du code
 - Décomposition par niveau
 - Bibliothèques



Introduction à la programmation

Fonctions et procédures



Introduction à la programmation

Fonctions et procédures

- Paramètres: variable d'entrée d'une fonction
- Résultat (return) : valeur de retour de la fonction
- Une fonction peut ne pas avoir de valeur de retour



Introduction à la programmation

Fonctions et procédures



Exemple: `afficher(arrondir(calculer(n)))`



Introduction à la programmation

Efficiency d'un programme / algorithme

- « Time + space complexity »
- Nous voulons prédire combien de ressources notre algorithme va utiliser, en général en fonction du temps d'exécution (running time)
- Notation $O(\dots)$



Introduction à la programmation

Efficiences d'un programme / algorithme

- Définit un modèle
 - Les instructions se suivent (séquentiel)
 - Les instructions de « base » coûtent $O(1)$
 - Arithmétiques : addition, soustraction, division, multiplication, arrondi, reste
 - Contrôle : branchement (conditionnels), retour, saut dans le code (call)
 - Manipulation de donnée : load, store, copy
- On omet la précision



Introduction à la programmation

Effcience d'un programme / algorithme

Programme1(n):

i=1

Total = 0

Tantque i <= n

Total = Total + i

i = i + 1

Fin Tantque

Return Total

Fin Programme1

Programme2(n)

Return $(n(n+1))/2$

Fin Programme2



Introduction à la programmation

Efficiences d'un programme / algorithme

Programme1(n):

i=1

Total = 0

Tantque i <= n

Total = Total + i

i = i + 1

Fin Tantque

Return Total

Fin Programme1

Programme2(n)

Return $(n(n+1))/2$

Fin Programme2

$O(1)$

$O(n)$



G

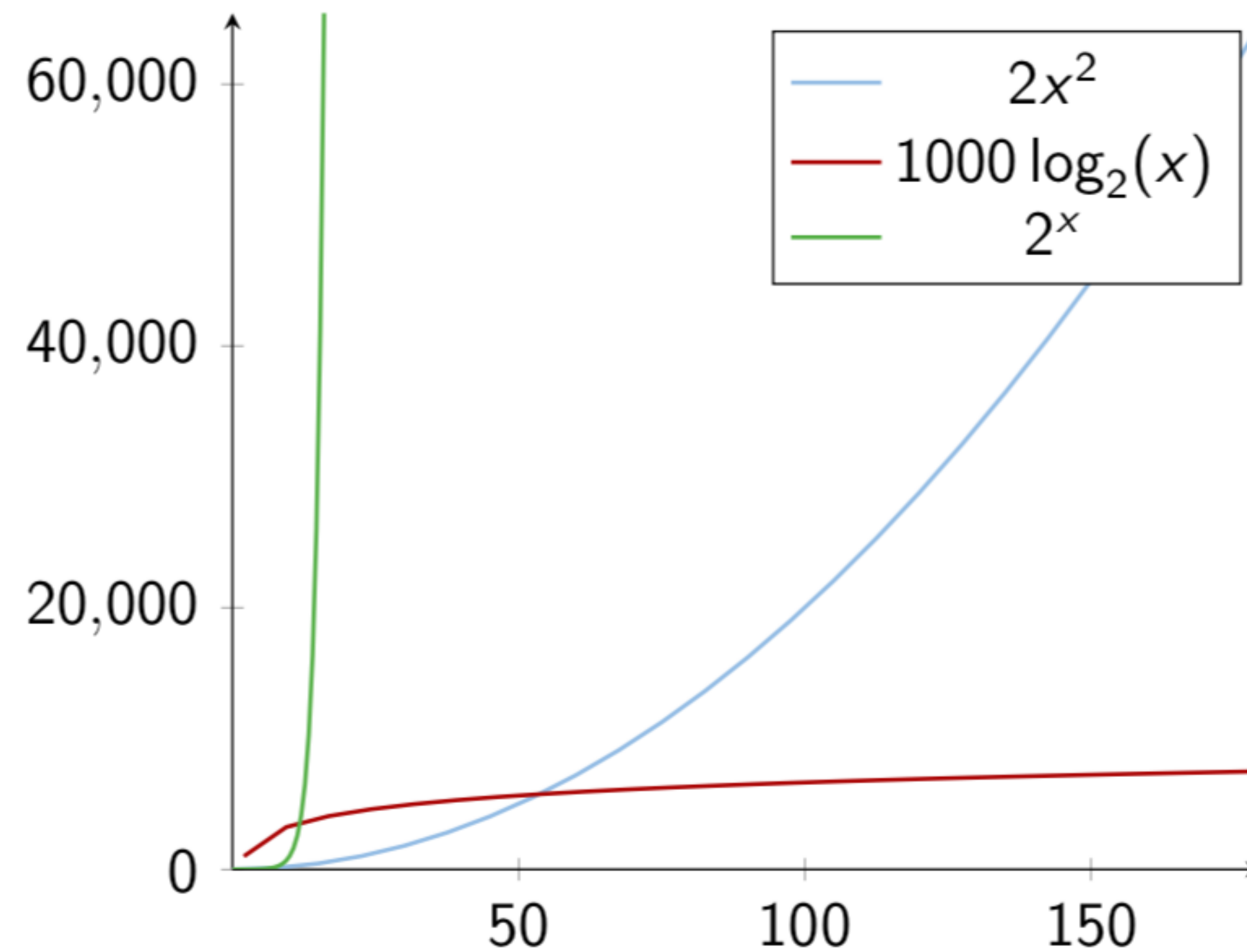
Efficiency d'un programme / algorithmme

	A ($1000 \log_2 n$ ms)	B ($2n^2$ ms)	C (2^n ms)
$n = 2$	1 s	8 ms	4 ms
$n = 4$	2 s	32 ms	16 ms
$n = 8$	3 s	128 ms	256 ms
$n = 16$	4 s	512 ms	1 m 5s 536 ms
$n = 32$	5 s	2 s 48 ms	\approx 49 days 18h
$n = 64$	6 s	8 s 192 ms	> age of the universe



Introduction à la programmation

Efficiences d'un programme / algorithme



Introduction à la programmation

Sommaire

- Informations manipulés
 - Variables, constante, paramètres
 - Portée, visibilité, effets de bords
- Les classes, objets
 - Attriuts, méthodes



Introduction à la programmation

Information manipulées / Données

- Valeur numérique
 - Entière
 - Flottante
 - Signée
 - Non-signée
- Caractère
- Chaîne de caractère
- Date / heure
- Booléen
- Ensemble d'octets (bytes)



Introduction à la programmation

Représentation e l'information

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



Introduction à la programmation

Représentation e l'information

Information	Type ?	Exemple ?
Nom	String	Pisanello
Rue	String	Route des acacias
Téléphone	String	079 999 99 99
Numéro postal	Integer	1227
Année de naissance	DateTime	Thu Oct 19 1995 08:00:00 GMT+0000
Date d'inscription	Date	Thu Oct 19 1995
Est membre?	Boolean	1, true
Remarque	String	Ceci est une remarque
Photo	Blob	011001111000101001110011



Introduction à la programmation

Information manipulées / Données

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



Introduction à la programmation

Constante

- Variable a 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 ou ensemble de valeur
- Avantage
 - Manipulation d'un identificateur (taux, nom) plutôt qu'une valeur
 - Espace temporaire de stockage entre les instructions
 - Persistance temporaire



Introduction à la programmation

Ensemble de valeurs

- Les informations peuvent être stockées / manipulées dans des structures
 - Liste
 - Tableau
 - ...
 - Objets (voir POO)



Introduction à la programmation

Effet de bord

- Effet de bord: Quand une fonction modifie un état en dehors de son environnement (niveau)
- Une variable n'est visible que dans le niveau et les sous-niveaux dans lesquels elle est définie
- Pour minimiser les risques (effet de bord), définir les variables au plus petit. Niveau possible



Introduction à la programmation

Effet de bord



Introduction à la programmation

Sommaire

- Informations manipulés
 - Variables, constante, paramètres
 - Portée, visibilité, effets de bords
- Les classes, objets
 - Attriuts, méthodes



Introduction à la programmation

Les classes, objets

- Structure de données et traitement, organisés ensemble: Date, String
- Les données sont appelées attributs ou propriétés
- Les traitements sont appelés méthodes



Introduction à la programmation

Les classes Date

- Attributs: date et heure
- Méthodes:

Method	Description		
<code>getDate()</code>	Returns the day of the month (from 1-31)	<code>setDate()</code>	Sets the day of the month of a date object
<code>getDay()</code>	Returns the day of the week (from 0-6)	<code>setFullYear()</code>	Sets the year (four digits) of a date object
<code>getFullYear()</code>	Returns the year (four digits)	<code>setHours()</code>	Sets the hour of a date object
<code>getHours()</code>	Returns the hour (from 0-23)	<code>setMilliseconds()</code>	Sets the milliseconds of a date object
<code>getMilliseconds()</code>	Returns the milliseconds (from 0-999)	<code>setMinutes()</code>	Set the minutes of a date object
<code>getMinutes()</code>	Returns the minutes (from 0-59)	<code>setMonth()</code>	Sets the month of a date object
<code>getMonth()</code>	Returns the month (from 0-11)	<code>setSeconds()</code>	Sets the seconds of a date object

