

# **Chapitre 2**

## **Variables, constantes et instructions de base en algorithmique**

# Objectifs pédagogiques du chapitre

## Objectif général

A la fin du chapitre, l'étudiant est capable d'utiliser une suite d'instructions algorithmiques simples pour réaliser l'enchaînement logique des opérations en rapport avec un traitement donné

# Objectifs pédagogiques du chapitre

## Objectifs spécifiques

A la fin du chapitre, l'étudiant est capable :

- de décrire correctement une variable
- de choisir entre 1 variable et 1 constante
- d'écrire une expression arithmétique ou logique combinant des variables, des constantes et des opérateurs
- d'utiliser une instruction d'affectation
- d'utiliser une instruction d'entrée-sortie

# Contenu

- Notion de variable
- Notion de constante
- Portée/visibilité des variables et des constantes
- Instructions de base (instruction d'affectation, instruction d'entrée ou de lecture, instruction de sortie ou d'écriture)

# Notion de variable (1/7)

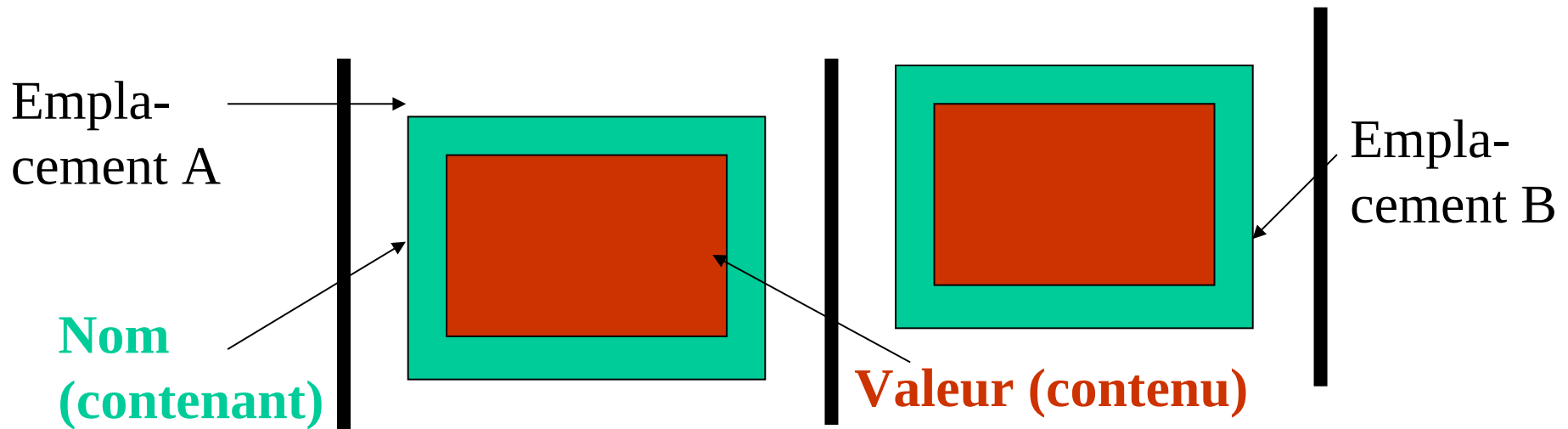
**Une variable** : est comparable à un récipient dans lequel il est possible de ranger une valeur donnée à un instant donné.



Au fil du temps, une variable peut être amenée à conserver de nombreuses valeurs, mais jamais deux à la fois.

# Notion de variable (1/7)

- Cette boîte est caractérisée par au moins :
  - un **nom** ou une étiquette ou un identificateur
  - un **contenu** ou une valeur rangée dans la boîte
  - un **emplacement** ou une adresse



# Notion de variable (2/7)

- **Le nom**

- est défini selon une syntaxe correspondant à un ensemble de règles (caractères alphanumériques, insensibilité par rapport à la casse, début du nom avec une lettre, non usage du caractère blanc « espace » et des lettres accentuées, usage du « souligné », etc.)
- est lisible et facilement interprétable
- permet en général de repérer ou faire référence au contenu correspondant

# Notion de variable (3/7)

- **Le contenu**

- est une valeur appartenant à un ensemble connu appelé **TYPE** en mesure de supporter un ensemble donné d'opérations
- est susceptible d'évoluer/varier au cours de l'algorithme, une caractéristique expliquant le nom de variable



# Notion de variable (4/7)

- Deux catégories de TYPES :
  - **Types scalaires ou non structurés** : ils ne sont plus décomposables car élémentaires. Ils sont soit prédéfinis (entiers naturels, entiers relatifs, réels, booléens, caractères, etc.), soit construits par le programmeur par énumération des valeurs (ensembles, intervalles, chaînes de caractères)
  - **Types structurés** : construits à partir des types scalaires grâce à des constructeurs de type (exemple : tableau, enregistrement ou article, liste)

# Notion de variable (5/7)

**ATTENTION !!!** Il y a deux différences majeures entre les *variables* en mathématiques et celles en informatique.

## Le nom

En général  $x, y, a, b$  ..... désignent des variables en mathématiques.

En informatique, il sera préférable de donner à chaque variable un ***nom significatif ou évocateur***.

## Le contenu

Une variable informatique ne peut contenir qu'une ***seule valeur à un moment donné***, alors qu'en mathématiques le contenu peut correspondre à plusieurs valeurs (ex. : **x** la **racine** d'un polynôme du 2<sup>nd</sup> degré)

# Notion de variable (6/7)

## Exemples

*Surface* est de type réel, correspond à la superficie d'un rectangle ;

*Age* est de type entier, correspond à l'âge d'une personne ;

*Rayon* est de type réel, correspond au rayon d'un cercle :

*AnneeNaiss* est de type réel, correspond à l'année de naissance de l'utilisateur ;

*Utilisateur* est de type chaîne de caractères, correspond au nom complet de l'utilisateur ;

*Nom* est de type chaîne de caractères, correspond au nom de l'utilisateur ;

*Prenom* est de type chaîne de caractères, correspond au prénom de l'utilisateur ;

*Reponse* est de type de caractère, correspond à la réponse de l'utilisateur valant « o/O » pour OUI et « n/N » pour NON ;

# Notion de constante (1/3)

- Une **constante**
  - est une **valeur particulière** d'un type donné, valeur **pas susceptible d'être modifiée au sein de l'algorithme**
  - porte un **nom évocateur** ou **symbolique** permettant de référencer la valeur associée
  - est un moyen pour améliorer la **lisibilité** et la **maintenabilité** des algorithmes
  - est **utilisable dans les expressions et instructions** portant sur des objets de son type

# Notion de constante (2/3)

## Principaux types de constantes

- Entier naturel
- Entier relatif
- Réel
- Logique (booléen)
- Caractère
- Chaîne de caractères

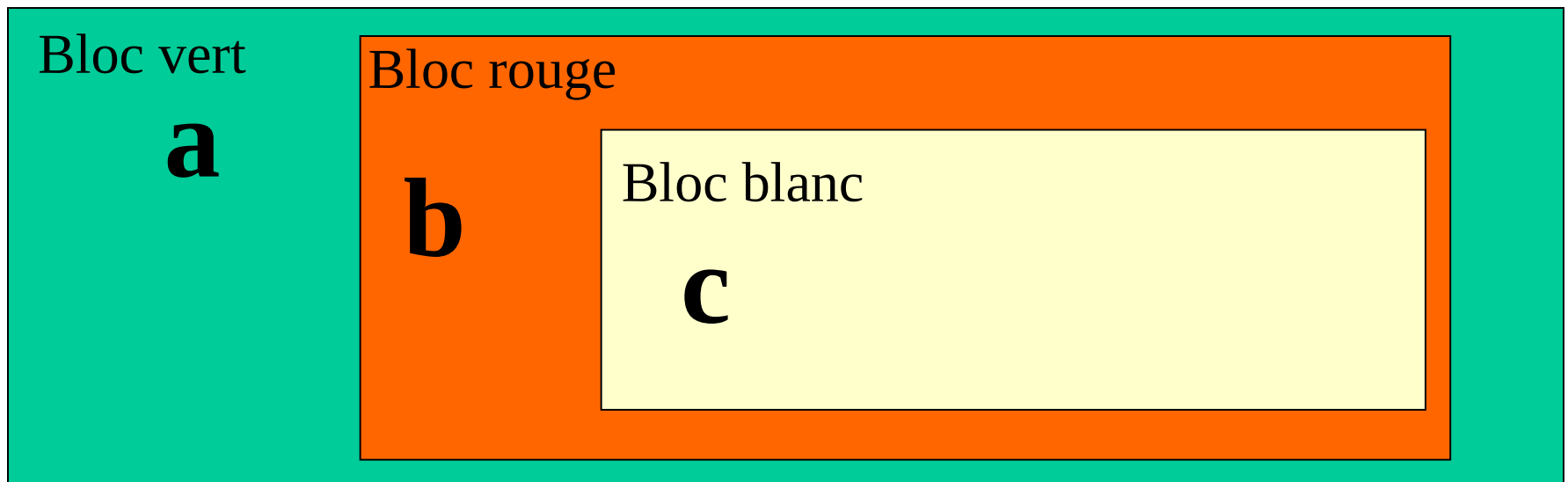
# Notion de constante (3/3)

## Exemples

Constantes  $PI \leftarrow 3,1415$  ;  
 $IndicatifTelPays \leftarrow 226$  ;  
 $baseSysteme \leftarrow 16$  ;  
 $nomPays \leftarrow \text{“Burkina Faso”}$  ;  
 $Separateur \leftarrow '/'$  ;  
 $Trouve \leftarrow \text{Faux}$  ;  
 $TaillePage \leftarrow 90$ .

# Portée des variables et constantes (1/6)

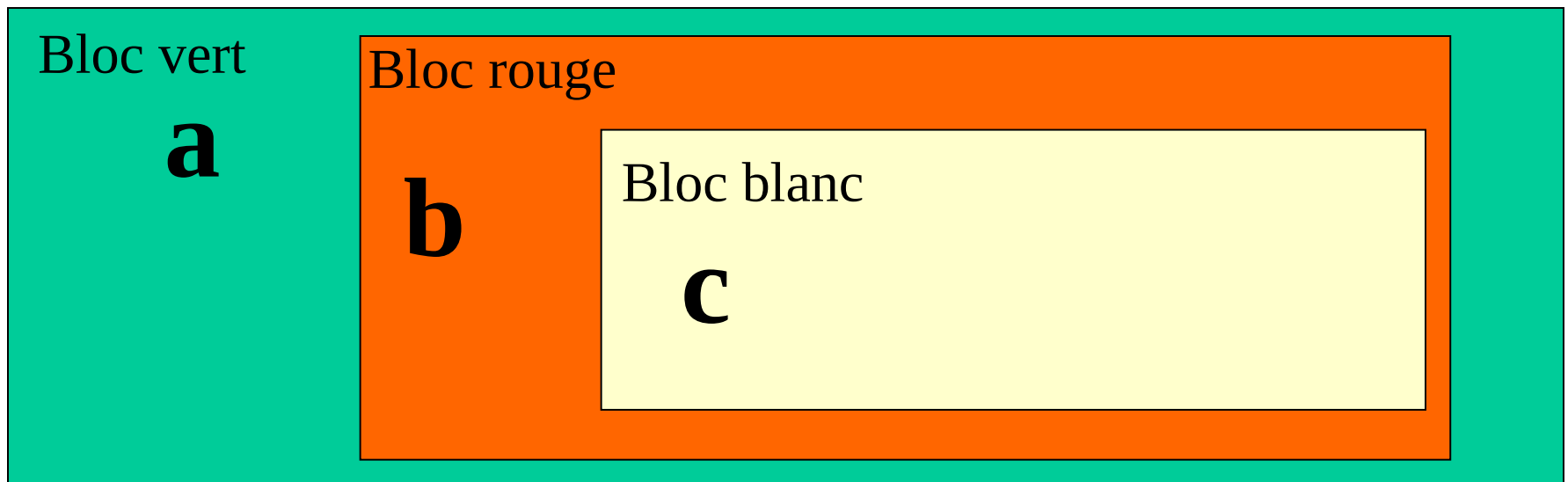
- Une variable ou une constante est **définie/déclarée** dans un bloc et n'est **visible/manipulable** qu'au sein de ce bloc (**portée**)



La variable **a** est déclarée dans le bloc vert, et du coup est connue dans le bloc rouge et le bloc blanc

# Portée des variables et constantes (2/6)

- Une variable ou une constante est **définie/déclarée dans un bloc** et n'est **visible/manipulable** qu'au sein de ce bloc (**portée**)

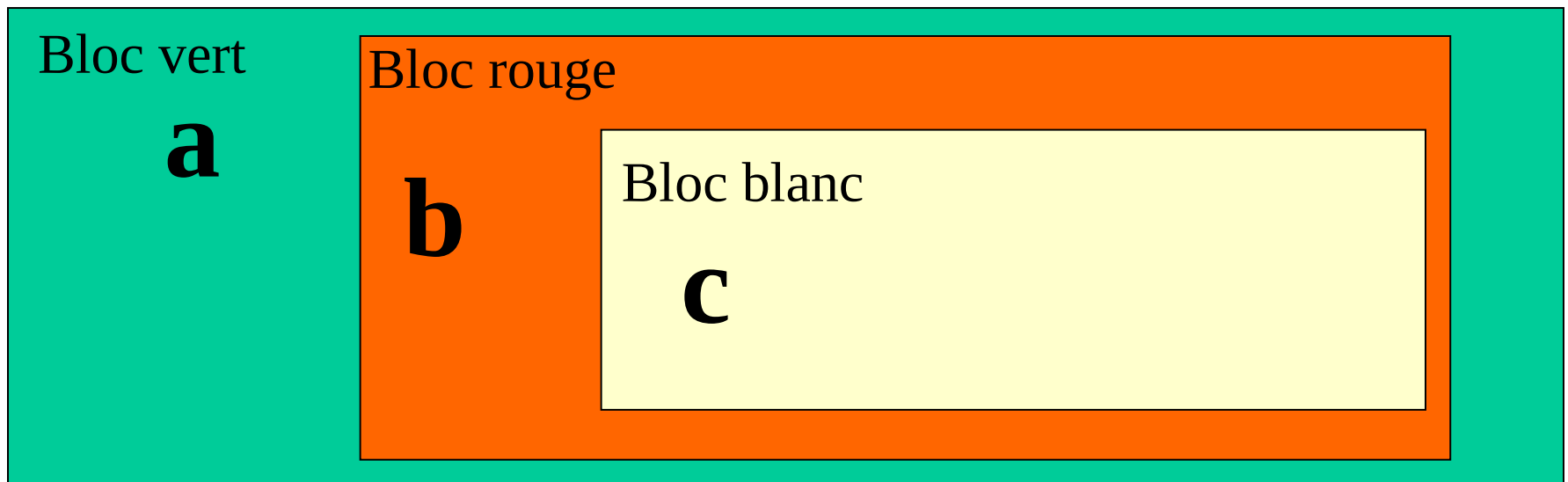


La variable **b** est déclarée dans le bloc rouge, n'est pas connue dans le bloc vert mais est connue dans le bloc blanc



# Portée des variables et constantes (3/6)

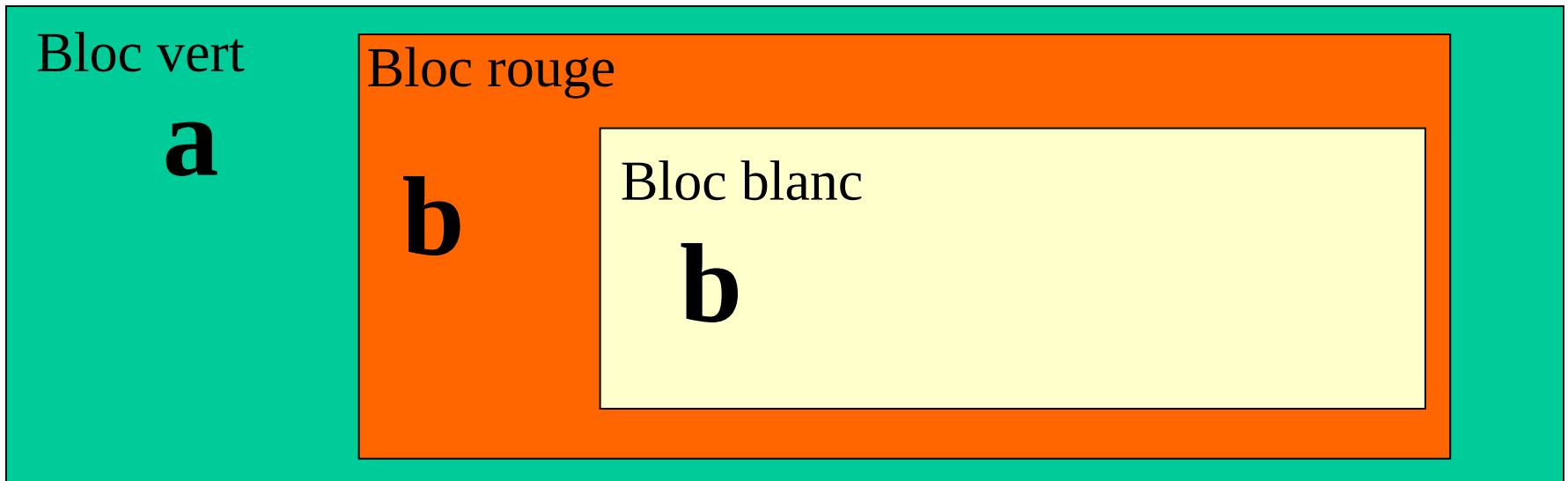
- Une variable ou une constante est **définie/déclarée** dans un bloc et n'est **visible/manipulable** qu'au sein de ce bloc (**portée**)



La variable **c** est déclarée dans le bloc blanc, n'est connue ni dans le bloc rouge ni dans le bloc vert

# Portée des variables et constantes (4/6)

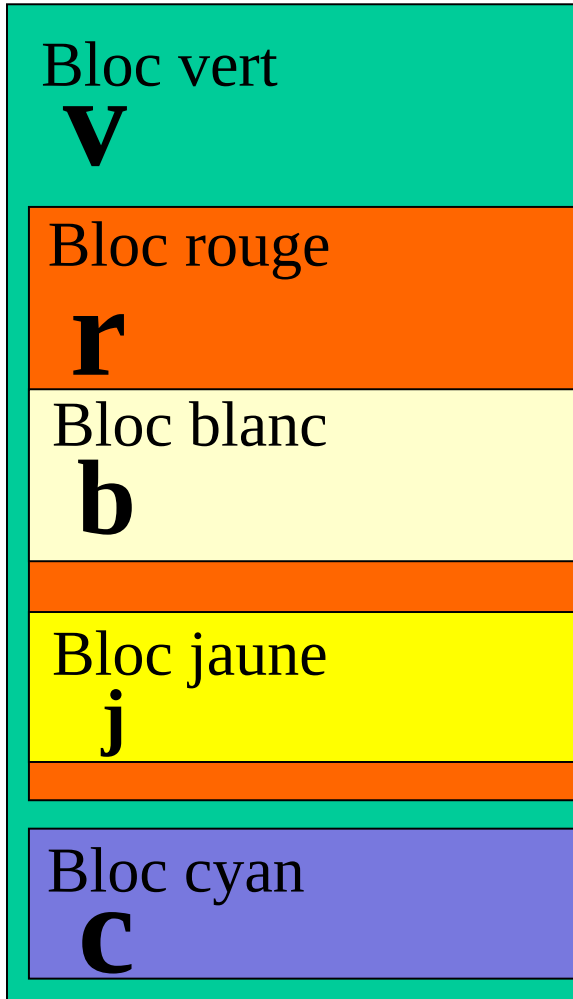
- Une variable ou une constante est **définie/déclarée dans un bloc** et n'est **visible/manipulable** qu'au sein de ce bloc (**portée**)



La variable **b** du bloc rouge n'est pas accessible dans le bloc blanc du fait qu'une autre variable de même nom y est déclarée. On dit que la variable **b** du **bloc blanc** rend **invisible celle du bloc rouge**

# Portée des variables et constantes (4/6)

## • Exercice 1

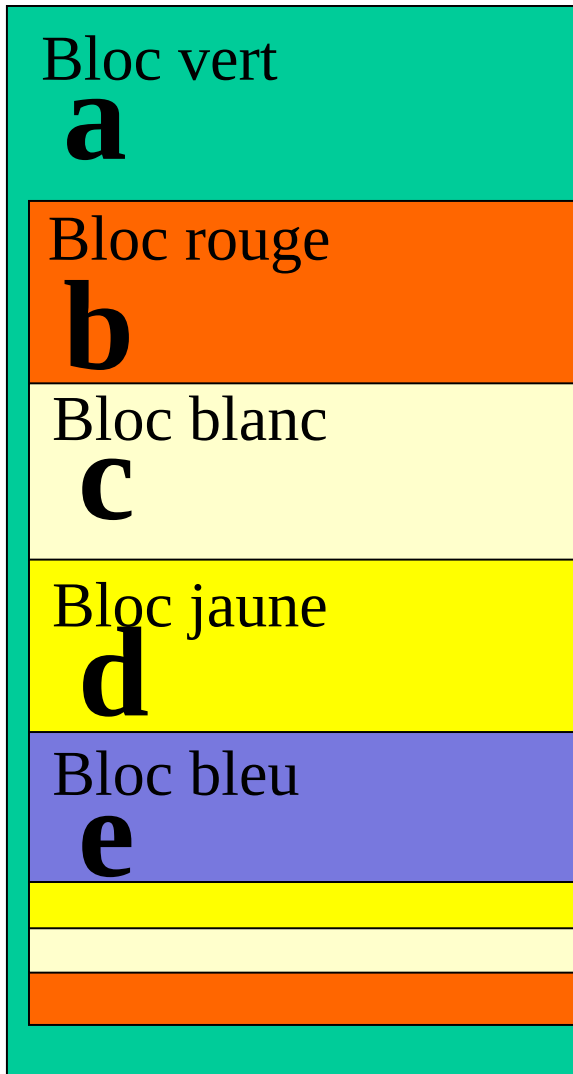


On demande d'indiquer la visibilité des différentes variables

	Bloc vert	Bloc rouge	Bloc blanc	Bloc jaune	Bloc cyan
v					
r					
b					
j					
c					

# Portée des variables et constantes (4/6)

## • Exercice 2



On demande d'indiquer la visibilité  
des différentes variables

# Instructions de base (1/12)

- Elles ont une présentation (**syntaxe**) et une signification (**sémantique**)
- Les instructions de base sont :

- l'affectation

- la lecture

- l'écriture

**Instructions  
d'entrées-sorties (E/S)**

**Instructions  
de transfert  
(mouvement)**

# Instructions de base (2/12)

- **L'affectation**

- Sa syntaxe

*nomVariable*  $\leftarrow$  *expression*

- Sa sémantique

ranger dans la variable *nomVariable*  
la valeur associée à *expression*

# Instructions de base (3/12)

- Expressions de différents types
  - entier
  - réel
  - caractère
  - chaîne de caractères
  - booléen ou logique
  - etc.

# Instructions de base (4/12)

- Expressions impliquant différents opérateurs et fonctions
  - opérateurs arithmétiques (entiers et réels) : +, -, x, /, etc.
  - opérateurs logiques ou booléens : **ET**, **OU**, >, <, ≤, ≥, =, ≠, etc.
  - opérateurs et fonctions pour les caractères : + de la concaténation, fonctions **ord**, **car**, **pred**, etc.
  - opérateurs et fonctions pour les chaînes de caractères : +, fonctions **taille**, **teteDeChaine**, **queueDeChaine**, etc.



# Instructions de base (5/12)

- Exemples

Surface  $\leftarrow$  0 ; /\* on range la valeur 0 dans la variable Surface \*/  
Rayon  $\leftarrow$  25.4 ; /\* on range la valeur 25.4 dans la variable Rayon \*/  
Surface  $\leftarrow$  2xPIxRayon ; /\* on suppose la constante PI connue \*/  
Age  $\leftarrow$  2015 - AnneeNaiss ; /\* on suppose AnneeNaiss déjà  
renseignée \*/

Nom  $\leftarrow$  "OUEDRAOGO" ;

Message  $\leftarrow$  Nom ;

Prenom  $\leftarrow$  "Justin" ;

Utilisateur  $\leftarrow$  Nom + " " + Prenom ;



Concaténation

# Instructions de base (5/12)

- Exemples

Nom  $\leftarrow$  "OUEDRAOGO" ;

Prenom  $\leftarrow$  "Justin" ;

Message  $\leftarrow$  "Bonjour" + " " + Nom + " " + Prenom ;

Utilisateur  $\leftarrow$  Nom + " " + Prenom ;

indicatif  $\leftarrow$  *IndicatifTelBF* ;

pays  $\leftarrow$  *nomPays* ;

estAdmis  $\leftarrow$  (Moyenne  $\geq$  12) ;

trouve  $\leftarrow$  (Prenom = "Jean") ;

estPresent  $\leftarrow$  (Moyenne  $\neq$  0) ;

toto  $\leftarrow$  "23456" ;

titi  $\leftarrow$  "23456" ;

somme  $\leftarrow$  toto + titi ;

car  $\leftarrow$  " " ;

# Instructions de base (6/12)

- **La lecture d'une valeur**

- Sa syntaxe

*Lire* (*nomVar*)

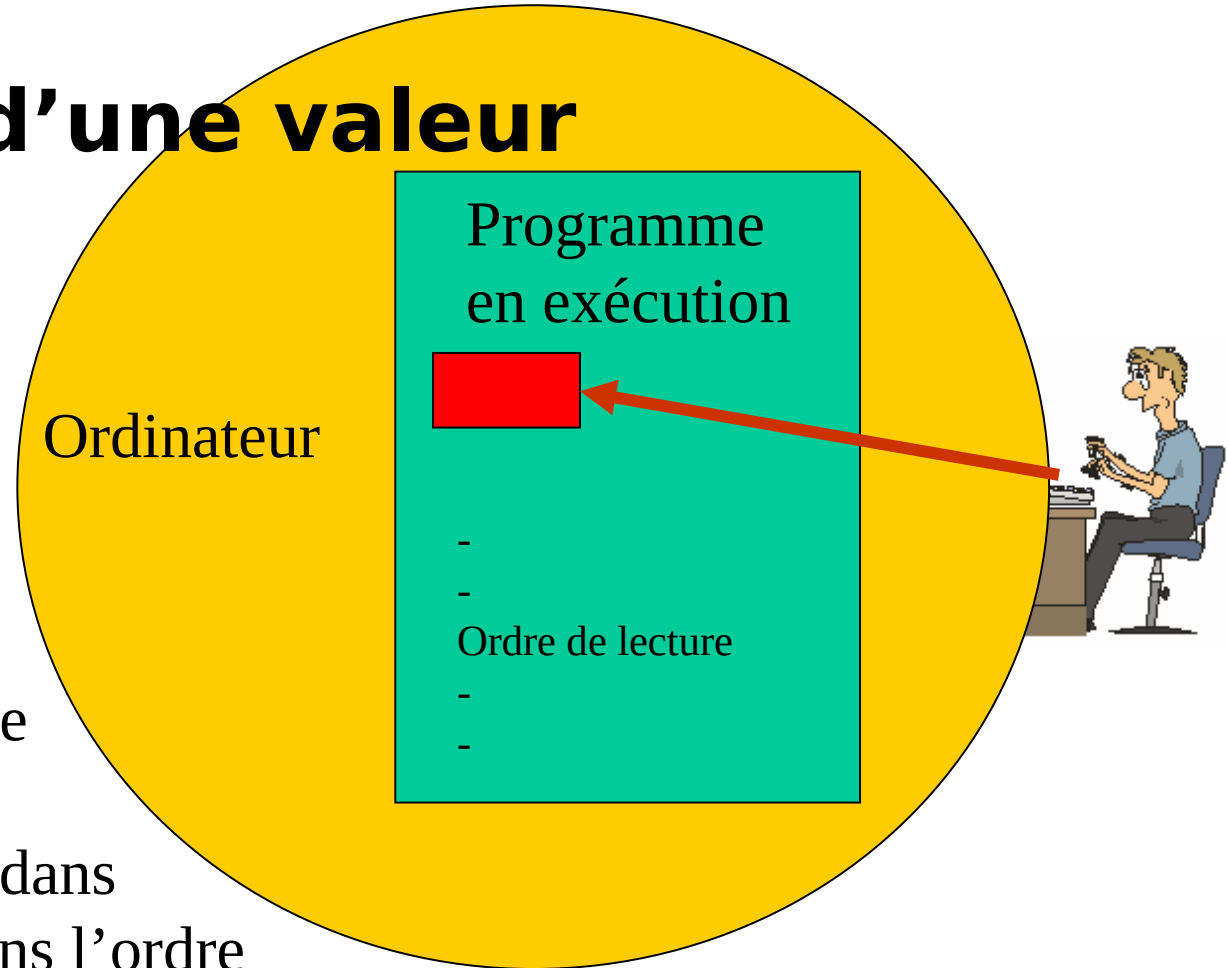
- Sa sémantique

**Récupérer une valeur introduite par l'utilisateur et la ranger dans la variable de nom *nomVar***

# Instructions de base (7/12)

## • La lecture d'une valeur

L'exécution de l'instruction de lecture correspond à une requête de fourniture d'une donnée adressée à l'utilisateur. La donnée sera rangée dans la variable spécifiée dans l'ordre



# Instructions de base (8/12)

- **Généralisation de la lecture**

- Sa syntaxe

***Lire** (*nomVar1, nomVar2, ..., nomVarN*)*

- Sa sémantique

**Récupérer**, au moyen d'une unique instruction , **N** valeurs introduites par l'utilisateur et qui seront respectivement stockées dans les variables de nom *nomVar1, nomVar2, ..., nomVarN*

# Instructions de base (9/12)

- **L'écriture**

- Sa syntaxe

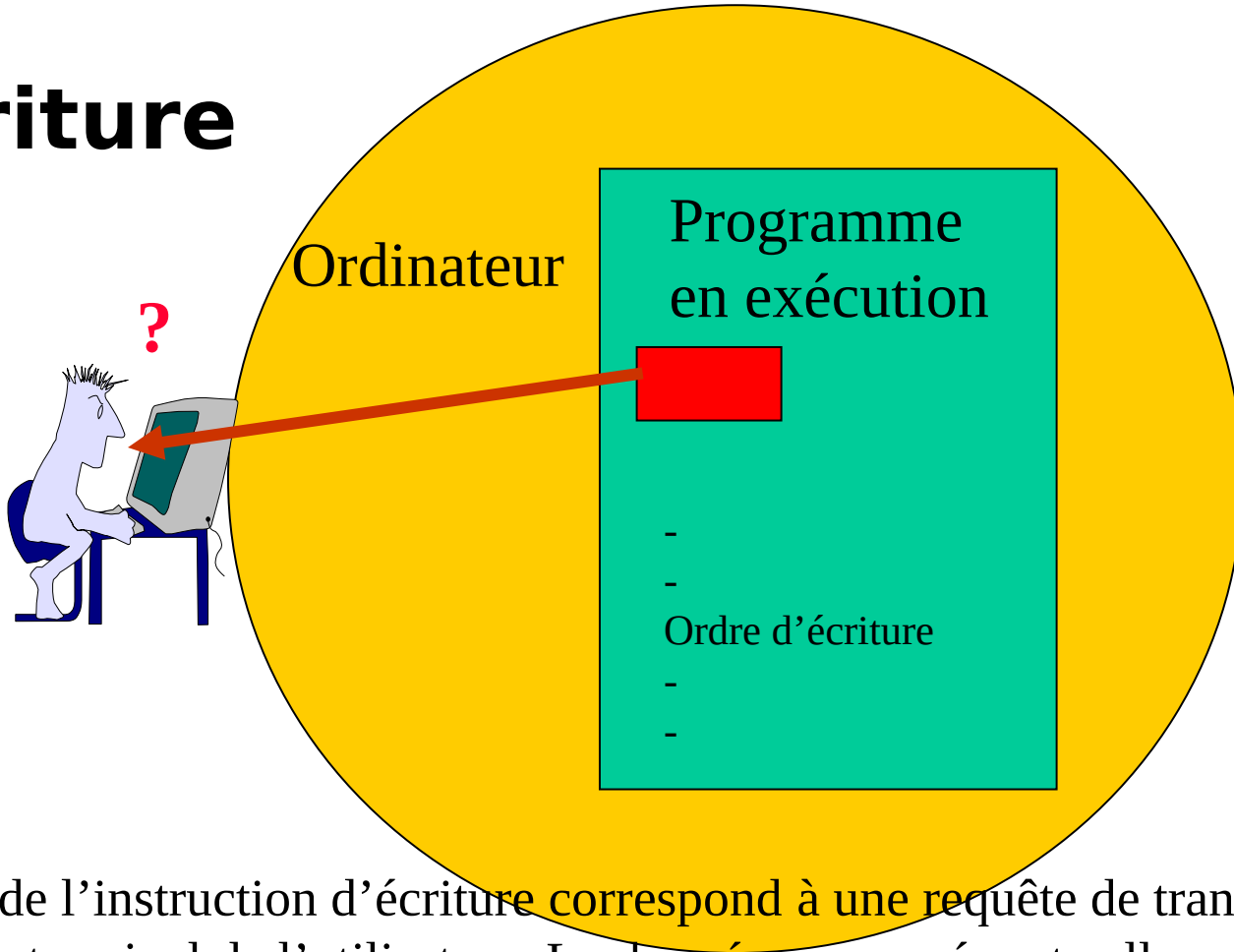
*Ecrire (nomVar)*

- Sa sémantique

récupérer la valeur présente dans la variable *nomVar* et l'envoyer vers le terminal de l'utilisateur

# Instructions de base (10/12)

## • L'écriture



L'exécution de l'instruction d'écriture correspond à une requête de transfert d'une valeur vers le terminal de l'utilisateur. La donnée concernée est celle rangée dans la variable spécifiée dans l'ordre.

# Instructions de base (11/12)

- **Généralisation de l'écriture**

- Sa syntaxe

*Ecrire* (*nomVar1*, *nomVar2*, ..., *nomVarN*)

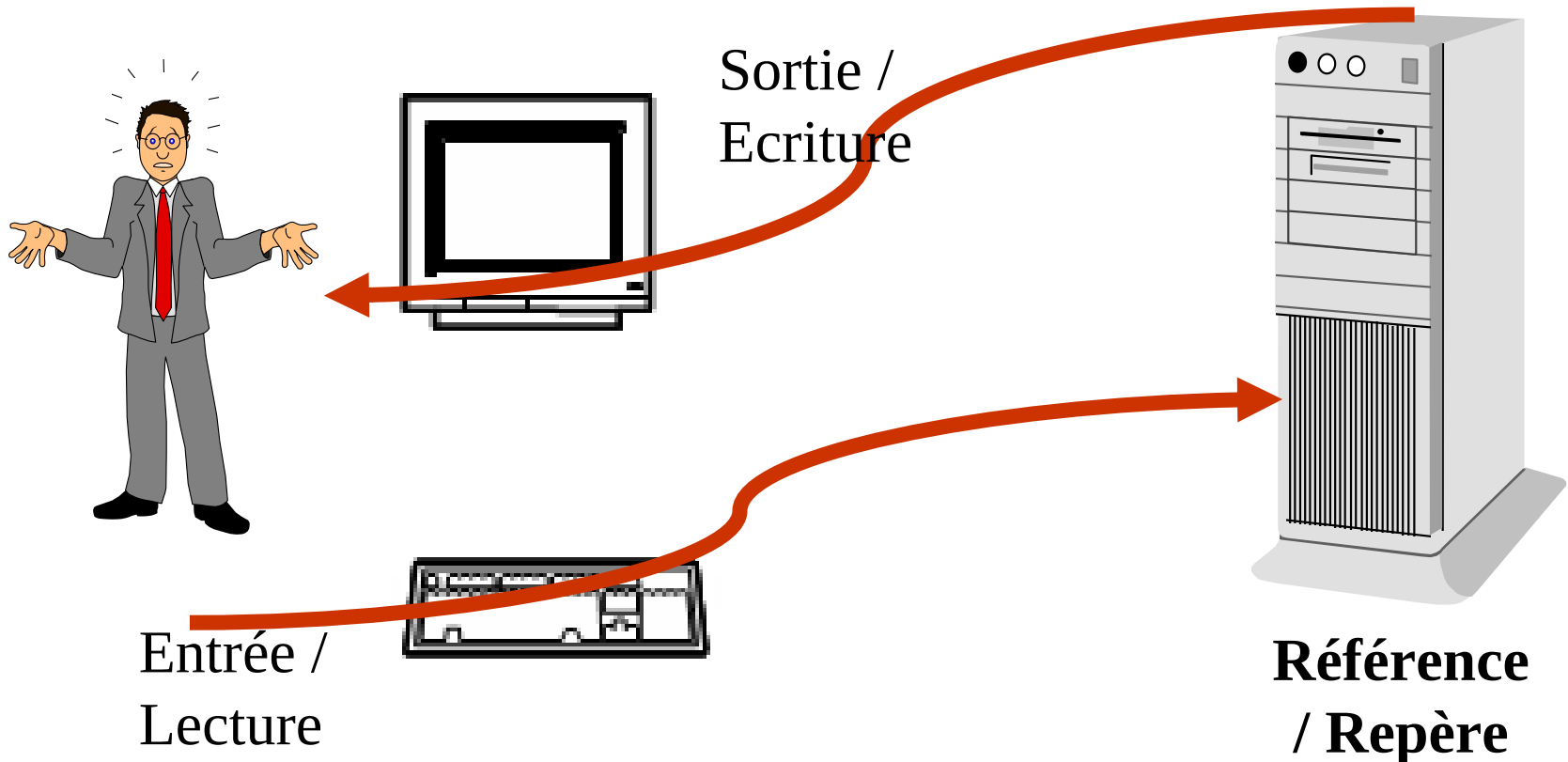
- Sa sémantique

**Récupérer**, au moyen d'une unique instruction , le contenu des **N variables** appelées respectivement *nomVar1*, *nomVar2*, ..., *nomVarN*, et les **envoyer vers le périphérique de sortie standard**



# Instructions de base (12/12)

- **Lecture (entrée) / écriture (sortie)**



**FIN**

**QUESTIONS ?**

## Exercice 1

Ecrire l'algorithme permettant de ranger la valeur entière 100 dans une variable dénommée X, de ranger la valeur 900 dans une variable de nom Y, de faire leur somme arithmétique et enfin de ranger le résultat dans la variable Z.

## Exercice 2

Ecrire l'algorithme permettant à l'utilisateur de saisir une valeur entière et d'afficher cette valeur augmentée de 100.

### Exercice 3a

Ecrire l'algorithme calculant et affichant le périmètre d'un rectangle dont les caractéristiques doivent être fournies par l'utilisateur.

### Exercice 3b

Ecrire l'algorithme calculant et affichant le demi-périmètre et le périmètre d'un rectangle dont les caractéristiques doivent être fournies par l'utilisateur.

## Exercice 4

En supposant A, B et C correspondre à des variables de type entier, donnez la valeur de chacune d'elles après l'exécution du bloc d'instructions suivantes.

**Début**

$A \leftarrow 5$

$B \leftarrow 3$

$C \leftarrow A + B$

$A \leftarrow 2$

$C \leftarrow B - A$

**Fin**

## Exercice 5

En supposant A, B et C correspondre à des variables de type entier, donnez la valeur de chacune d'elles après l'exécution du **traitement** correspondant au bloc d'instructions suivantes.

**Début**

$A \leftarrow 3 ; B \leftarrow 10 ; C \leftarrow A + B ; B \leftarrow A + B ; A \leftarrow C$

**Fin**

## Exercice 6

Écrire un algorithme permettant de calculer le prix TTC (toutes taxes comprises) d'un lot d'un même article pris dans une boutique.

## Exercice 7

Écrire un algorithme permettant de calculer la moyenne de fin de semestre d'un étudiant de Lst1 en anglais, sachant que le DAR compte pour  $1/6$ , le DST  $1/3$  et l'examen pour  $1/2$  de cette moyenne sur 20.

## Exercice 8

Écrire un algorithme permettant de calculer et d'afficher le nombre de milliers, le nombre de centaines, le nombre de dizaines et le nombre d'unités correspond à une valeur entière fournie par l'utilisateur.

## Exercice 9

Écrire un algorithme qui, à partir de la fourniture par l'utilisateur d'un nombre entier correspondant à un nombre de secondes, calcule en résultats l'équivalent en heures, minutes et secondes.

## Exercice 10

Écrire un algorithme qui, à partir de la fourniture par l'utilisateur d'un nombre entier correspondant à un nombre de secondes, calcule et affiche l'équivalent en heures, minutes et secondes.