CHAPITRE 2

La lecture et l'écriture

Dialoguer avec l'utilisateur



AU PROGRAMME

1. Le dialogue avec l'utilisateur
2. L'écriture : ECRIRE
3. La concaténation : &
4. La lecture : LIRE
5. Calculer avec les valeurs lues

Nom : ..... **Classe :** .....

AVANT DE COMMENCER

Ce que tu vas apprendre

Un programme qui ne montre rien et ne demande rien ne sert pas à grand-chose. Avec deux instructions, ECRIRE et LIRE, ton algorithme va afficher des messages et recevoir ce que l'utilisateur tape au clavier. Tu apprendras aussi à assembler des textes et à calculer avec les valeurs reçues.

1. **Expliquer la différence entre la lecture et l'écriture.**
2. **Afficher un texte ou la valeur d'une variable avec `ECRIRE`.**
3. **Assembler textes et variables avec la concaténation `&`.**
4. **Demander une valeur à l'utilisateur avec `LIRE`.**
5. **Calculer avec `+`, `-`, `*`, `/` et `mod` (le reste de la division).**
6. **Écrire un programme complet qui demande, calcule et affiche.**

PARTIE 1

Le dialogue avec l'utilisateur

Entre l'utilisateur et la machine, l'information circule dans **les deux sens**. Le programme affiche des messages à l'écran ; l'utilisateur lui répond au clavier.



Figure 2.1 — LIRE fait entrer une information dans le programme ; ECRIRE la fait sortir vers l'écran.

L'écriture

Le programme **affiche** une information à l'écran : une question, un résultat, un message d'erreur.

La lecture

Le programme **attend** que l'utilisateur tape une valeur au clavier, puis la range dans une variable.

Exemple : la connexion à Windows

1. **Écriture**
   Windows affiche « Nom d'utilisateur » et « Mot de passe ».

2. **Lecture**
   Tu tapes tes informations au clavier.

3. **Vérification**
   Le programme compare ce que tu as tapé avec ce qui est enregistré (c'est un test : chapitre 3).

4 **Écriture**
Windows affiche ton bureau... ou un message d'erreur.

ATTENTION
Lire ou écrire ?
Attention au point de vue : on se place toujours du côté du **programme**. LIRE, c'est le programme qui lit le clavier ; ECRIRE, c'est le programme qui écrit à l'écran.

PARTIE 2
L'écriture : ECRIRE

DÉFINITION
L'instruction ECRIRE
Elle affiche à l'écran ce qui la suit : un **texte entre guillemets**, la **valeur d'une variable** (son nom, sans guillemets) ou un calcul.

PSEUDOCODE
```
variable age en entier
DEBUT
    age <- 16
    ECRIRE "Hello World"
    ECRIRE age
    ECRIRE "age"
    ECRIRE age * 2
FIN
```

À L'ÉCRAN
```
Hello World
16
age
32
```

Remarque les deux dernières lignes : `ECRIRE age` affiche la **valeur** 16, alors que `ECRIRE "age"` affiche le **mot** age. Les guillemets font toute la différence, comme pour l'affectation.

À RETENIR

Ce qu'il faut savoir

- Chaque `ECRIRE` affiche sa ligne puis passe à la ligne suivante.
- Un texte s'écrit entre guillemets ; une variable, sans guillemets.
- On peut afficher le résultat d'un calcul : `ECRIRE prix * 2`.

PARTIE 3

La concaténation : &

DÉFINITION

La concaténation

Coller plusieurs morceaux de texte (et des valeurs de variables) pour en faire **une seule** chaîne de caractères. En pseudocode, elle se note `&`.

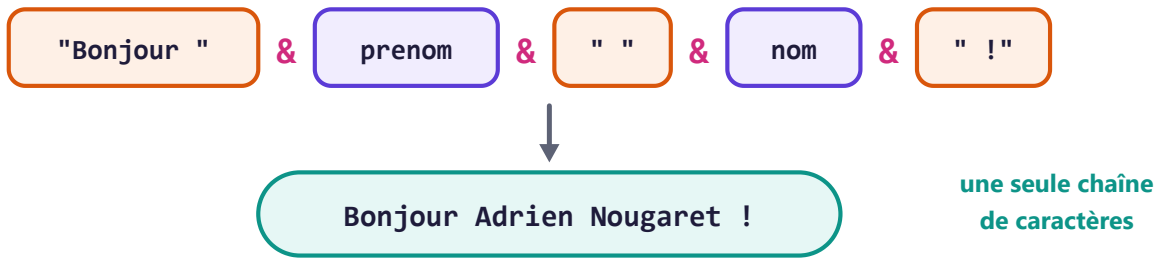


Figure 2.2 — Cinq morceaux collés par & : une seule phrase à l'écran.

PSEUDOCODE

```
variables prenom, nom en chaine de caracteres
DEBUT
    prenom <- "Adrien"
    nom <- "Nougaret"
    ECRIRE "Bienvenue " & prenom & " " & nom & " !"
FIN
```

À L'ÉCRAN

```
Bienvenue Adrien Nougaret !
```

La machine ne met **aucun espace** toute seule : il faut le prévoir dans les guillemets (`"Bienvenue "`) ou ajouter un morceau `" "` entre deux variables.

ATTENTION

Jamais de virgule

Dans un `ECRIRE`, on relie les morceaux avec `&`, **jamais avec une virgule**. `ECRIRE "Tu as ", age` est faux ; on écrit `ECRIRE "Tu as " & age & " ans"`.

Concaténer dans une affectation

La concaténation fonctionne aussi à droite de la flèche : on prépare le message dans une variable, puis on l'affiche. Un nombre collé à du texte devient du texte.

PSEUDOCODE

```
variable message en chaine de caracteres
variable score en entier
DEBUT
    score <- 12
    message <- "Score : " & score & " points"
    ECRIRE message
FIN
```

À L'ÉCRAN

```
Score : 12 points
```

PARTIE 4

La lecture : LIRE

DÉFINITION

L'instruction LIRE

Elle **met le programme en pause**, attend que l'utilisateur tape une valeur et appuie sur Entrée, puis range cette valeur dans la variable indiquée.

PSEUDOCODE

```
variable prenom en chaine de caracteres
DEBUT
    ECRIRE "Quel est ton prénom ?"
    LIRE prenom
    ECRIRE "Bonjour " & prenom & " !"
FIN
```

À L'ÉCRAN

```
Quel est ton prénom ?
Lina
Bonjour Lina !
```

Dans VS Code, ce que l'utilisateur tape s'affiche sur sa propre ligne : c'est la ligne « Lina » de l'écran ci-dessus.

À RETENIR

Les règles de LIRE

- La variable doit être **déclarée** avant, avec le bon type.
- On écrit toujours un `ECRIRE` **avant** le `LIRE` : sinon l'utilisateur ne sait pas ce qu'on attend de lui.
- La valeur tapée **remplace** l'ancienne valeur de la variable.

Le type compte

Si la variable est un `entier`, l'utilisateur doit taper un nombre entier : s'il tape « douze », le programme s'arrête sur une erreur. Un `booleen` se lit **directement** : l'utilisateur tape `true` ou `false`.

PSEUDOCODE

```
variable permis en booleen
DEBUT
    ECRIRE "As-tu ton permis ? (true ou false)"
    LIRE permis
    ECRIRE "Permis : " & permis
FIN
```

À L'ÉCRAN

```
As-tu ton permis ? (true ou false)
True
Permis : True
```

POUR ALLER PLUS LOIN

L'utilisateur a tapé `true`, mais l'écran montre `True` : pour afficher un booléen, le C# (qui exécute ton pseudocode dans VS Code) écrit `True` ou `False` avec une majuscule.

PARTIE 5

Calculer avec les valeurs lues

Opérateur	Rôle	Exemple	Résultat
`+`	addition	`7 + 2`	9
`-`	soustraction	`7 - 2`	5
`*`	multiplication	`7 * 2`	14
`/`	division	`7 / 2`	3 entre deux entiers (3.5 entre numeriques)
`mod`	reste de la division entière	`7 mod 2`	1

DÉFINITION

Le modulo (mod)

Le **reste** de la division entière. `17 mod 5` vaut 2, car 17 = 3 × 5 + **2**. Il sert par exemple à savoir si un nombre est pair : `n mod 2` vaut 0 pour un nombre pair.

ATTENTION

Deux pièges de VS Code

Dans VS Code, `/` entre deux `entier` donne un **quotient entier** : `7 / 2` vaut 3, la virgule est perdue. Et l'addition `+` n'accepte que des `entier` : pour additionner des nombres à virgule, attends le C# (chapitre 6).

PSEUDOCODE **Un programme complet : demander, calculer, afficher**

```
variables largeur, longueur, surface en entier
DEBUT
    ECRIRE "Largeur de la pièce (en m) ?"
    LIRE largeur
    ECRIRE "Longueur de la pièce (en m) ?"
    LIRE longueur
    surface <- largeur * longueur
    ECRIRE "Surface : " & surface & " m²"
FIN
```

À L'ÉCRAN

```
Largeur de la pièce (en m) ?
4
Longueur de la pièce (en m) ?
5
Surface : 20 m²
```

À RETENIR

Le schéma de presque tous les programmes

- **Demander** : un `ECRIRE` pour la question, un `LIRE` pour la réponse.
- **Calculer** : une ou plusieurs affectations.
- **Afficher** : un `ECRIRE` avec `&` pour présenter le résultat.

BILAN DU CHAPITRE 2

L'essentiel à retenir

1. **Écrire**, c'est afficher à l'écran ; **lire**, c'est recevoir une valeur tapée au clavier. On se place toujours du côté du programme.
2. `ECRIRE "texte"` affiche le texte ; `ECRIRE variable` affiche sa **valeur** ; chaque ECRIRE passe à la ligne.
3. La **concaténation** `&` colle textes et variables en une seule chaîne. Les espaces se mettent dans les guillemets. Jamais de virgule.
4. `LIRE variable` met le programme en pause et range la valeur tapée dans la variable, qui doit être déclarée avec le bon type.
5. Un `ECRIRE` qui pose la question vient toujours **avant** le `LIRE`.
6. On calcule avec `+`, `-`, `*`, `/` et `mod`, le **reste** de la division entière.

Le lexique

Écriture
Le programme affiche une information à l'écran (`ECRIRE`).

Lecture
Le programme attend une valeur tapée au clavier et la range dans une variable (`LIRE`).

Concaténation
Coller textes et valeurs en une seule chaîne avec `&` .

Modulo
Le reste de la division entière (`mod`) : `17 mod 5` vaut 2.

Quotient
Le résultat de la division entière : 17 / 5 donne 3.

Écran d'exemple
Ce qu'un programme doit afficher, ligne par ligne, avec les valeurs tapées.

Je fais le point

Coche une case par ligne : où en es-tu avant l'interro ?

Je suis capable de…	Oui	Presque	Pas encore
Expliquer la différence entre lecture et écriture.	☐	☐	☐
Afficher un texte, une variable ou un calcul avec `ECRIRE`.	☐	☐	☐
Construire une phrase avec `&` sans oublier les espaces.	☐	☐	☐
Demander une valeur avec `ECRIRE` puis `LIRE`.	☐	☐	☐
Calculer un reste avec `mod`.	☐	☐	☐
Écrire un programme complet qui demande, calcule et affiche.	☐	☐	☐

CHAPITRE 2 · EXERCICES

Teste tes connaissances

avant l'interro

COMMENT T'Y PRENDRE

1. **Relis la synthèse** : l'essentiel, le lexique et « Je fais le point ».
2. **Fais les exercices sans le cours**, comme à l'interro.
3. **Corrige-toi** avec la correction vue en classe ou sur le site.
4. **Reviens sur tes erreurs** : relis la partie du cours concernée.

LES 6 EXERCICES

1. **Lire ou écrire ?**
2. **Qu'affiche chaque instruction ?**
3. **Les règles de la lecture et de l'écriture**
4. **Prédire l'écran**
5. **Corrige l'algorithme**
6. **Mes premiers dialogues**

Exercice 1 Lire ou écrire ?

▸ **Coche la bonne réponse.**

1. Quelle instruction affiche un message à l'écran ?

☐ LIRE ☐ ECRIRE ☐ DEBUT

2. Que fait le programme quand il arrive sur `LIRE note` ?

☐ il affiche la note ☐ il attend que l'utilisateur tape une valeur

☐ il efface la note

3. Quel symbole sert à la concaténation ?

☐ + ☐ , ☐ & ☐ ;

4. Combien vaut `14 mod 4` ?

☐ 2 ☐ 3 ☐ 3.5 ☐ 0

Exercice 2 Qu'affiche chaque instruction ?

▸ **La variable nom contient "Lina". Relie chaque instruction à ce qu'elle affiche (écris la lettre).**

1	`ECRIRE "nom"` →	**A**	Salut Lina
2	`ECRIRE nom` →	**B**	SalutLina
3	`ECRIRE "Salut " & nom` →	**C**	la valeur de la variable nom, par exemple Lina
4	`ECRIRE "Salut" & nom` →	**D**	le mot nom

Exercice 3 Les règles de la lecture et de l'écriture

▸ **Coche vrai ou faux. Si c'est faux, corrige la phrase sur la ligne.**

Affirmation	Vrai	Faux
Un `LIRE` doit être précédé d'un `ECRIRE` qui pose la question.	☐	☐
`ECRIRE "Tu as ", age` affiche correctement l'âge.	☐	☐
La concaténation ajoute automatiquement un espace entre deux morceaux.	☐	☐
`LIRE age` fonctionne même si age n'est pas déclarée.	☐	☐
`10 mod 2` vaut 0 : 10 est pair.	☐	☐

Exercice 4 Prédire l'écran

▸ **Les variables entières a et b valent 8 et 3. Écris ce que chaque instruction affiche.**

Instruction	Ce qui s'affiche
`ECRIRE a + b`	
`ECRIRE "a + b"`	
`ECRIRE a & " et " & b`	
`ECRIRE a mod b`	
`ECRIRE a / b`	

Exercice 5 Corrige l'algorithme

▸ **Cet algorithme contient deux erreurs.**

PSEUDOCODE

```
1  variable age en entier
2  DEBUT
3      // demander l'âge
4      LIRE age
5      ECRIRE "Quel âge as-tu ?"
6      ECRIRE "Dans 10 ans, tu auras ", age + 10, " ans"
7  FIN
```

1. Ligne 4 : quelle est l'erreur ? Corrige l'instruction.

2. Ligne 6 : quelle est l'erreur ? Corrige l'instruction.

Exercice 6 Mes premiers dialogues

- **Écris l'algorithme complet : déclarations, DEBUT, instructions, FIN.**

1. Demande à l'utilisateur son prénom et son année de naissance, puis affiche l'âge qu'il aura en 2026.

EXEMPLE D'ÉCRAN

```
Quel est ton prénom ?
Lina
En quelle année es-tu né(e) ?
2010
Lina, en 2026 tu auras 16 ans.
```

2. Demande une durée en minutes et affiche-la en heures et minutes (pense à `/` et à `mod`).

EXEMPLE D'ÉCRAN

```
Durée en minutes ?
135
135 minutes = 2 h 15 min
```