

0.1 Un premier ensemble de macros additionnelles ou francisées

À SAVOIR – Le préfixe *u*

*Certains macros peuvent être préfixées par un **u** pour « unclosed » qui signifie « non fermé ». Ceci sert à ne pas refermer un bloc via un trait horizontal.*

0.1.1 Entrée / Sortie

Nous donnons ci-dessous les versions au singulier de tous les mots disponibles de type « entrée / sortie ». Excepté pour `\InState` et `\OutState`, toutes les autres macros ont une version pour le pluriel obtenu en rajoutant un **s** à la fin du nom de la macro. Par exemple, le pluriel de `\In` s'obtient via `\Ins`.

Code *L^AT_EX*

```
\begin{algo}
  \In{donnée 1}
  \Out{donnée 2}
\end{algo}
```

Mise en forme correspondante.

Entrée : donnée 1
Sortie : donnée 2

Code *L^AT_EX*

```
\begin{algo}
  \Data{donnée 1}
  \Result{donnée 2}
\end{algo}
```

Mise en forme correspondante.

Donnée : donnée 1
Résultat : donnée 2

Code *L^AT_EX*

```
\begin{algo}
  \InState{donnée 1}
  \OutState{donnée 2}
\end{algo}
```

Mise en forme correspondante.

État initial : donnée 1
État final : donnée 2

Code *L^AT_EX*

```
\begin{algo}
  \PreCond{donnée 1}
  \PostCond{donnée 2}
\end{algo}
```

Mise en forme correspondante.

Précondition : donnée 1
Postcondition : donnée 2

0.1.2 Bloc principal

Voici comment indiquer le bloc principal d'instructions avec deux textes au choix pour le moment.

Code L^AT_EX

```
\begin{algo}
  \Actions{Instruction 1}
  \Begin{Instruction 2}
\end{algo}
```

Mise en forme correspondante.

Actions

└ Instruction 1

Début

└ Instruction 2

0.1.3 Boucles POUR et TANT QUE

Voici les boucles de type POUR et TANT QUE proposées par le package.

Code L^AT_EX

```
\begin{algo}
  \For{$i$ \in uneliste$}{
    Instruction 1
  }
  \ForAll{$i$ \in uneliste$}{
    Instruction 2
  }
  \ForEach{$i$ \in uneliste$}{
    Instruction 3
  }
  \While{$i$ \in uneliste$}{
    Instruction 4
  }
\end{algo}
```

Mise en forme correspondante.

Pour $i \in \text{uneliste}$:
└ Instruction 1
Pour Tout $i \in \text{uneliste}$:
└ Instruction 2
Pour Chaque $i \in \text{uneliste}$:
└ Instruction 3
Tant Que $i \in \text{uneliste}$:
└ Instruction 4

0.1.4 Boucles RÉPÉTER JUSQU'À AVOIR

Voici comment rédiger une boucle du type RÉPÉTER JUSQU'À AVOIR.

Code L^AT_EX

```
\begin{algo}
  \Repeat{$i$ \in uneliste$}{
    Instruction
  }
\end{algo}
```

Mise en forme correspondante.

Répéter
└ Instruction
Jusqu'à Avoir $i \in \text{uneliste}$;

0.1.5 Disjonction de cas SUIVANT CAS

La syntaxe pour les blocs conditionnels du type SUIVANT CAS ne pose pas de difficultés de rédaction.

Code L^AT_EX

```
\begin{algo}
  \Switch{$i$}{
    \uCase{$i = 0$}{Instruction 1}
    \uCase{$i = 1$}{Instruction 2}
    \Case{$i = 2$}{Instruction 3}
  }
\end{algo}
```

Mise en forme correspondante.

Suivant i :
└ **Cas** $i = 0$:
└ └ Instruction 1
└ **Cas** $i = 1$:
└ └ Instruction 2
└ **Cas** $i = 2$:
└ └ Instruction 3

0.1.6 Tests conditionnelles SI - SINON

Les blocs conditionnels SI - SINON se rédigent très naturellement.

Code $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

```
\begin{algo}
  \uIf{$i = 0$}{
    Instruction 1
  }
  \uElseIf{$i = 1$}{
    Instruction 2
  }
  \Else{
    Instruction 3
  }
\end{algo}
```

Mise en forme correspondante.

Si $i = 0$:
| Instruction 1
Sinon Si $i = 1$:
| Instruction 2
Sinon
└ Instruction 3

0.1.7 Diverses commandes

Pour finir voici un ensemble de mots supplémentaires qui pourront vous rendre service. Le préfixe **m** permet d'utiliser des versions maculines des textes proposés.

Code $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

```
\begin{algo}
  A \And B \Or C
  \\\ \Return RÉSULTAT
  \\\ \Ask "Quelque chose"
  \\\ \Print "Quelque chose"
  \\\ $k$ \From $1$ \To $n$
  \\\ $k$ \ComingFrom $1$ \GoingTo $n$
  \\\ $e$ \InThis $\{1, 2, 3\}$
  \\\ $L$ \LToR
  \\\ $L$ \LToRm
  \\\ $L$ \RToL
  \\\ $L$ \RToLm
\end{algo}
```

Mise en forme correspondante.

A et B **ou** C
Renvoyer RÉSULTAT
Demander "Quelque chose"
Afficher "Quelque chose"
 k de 1 à n
 k allant de 1 jusqu'à n
 e dans $\{1, 2, 3\}$
 L parcourue de gauche à droite
 L parcouru de gauche à droite
 L parcourue de droite à gauche
 L parcouru de droite à gauche

0.2 Citer les outils de base en algorithmique

Pour faciliter la rédaction de textes sur les algorithmes, des macros standardisent l'impression des noms des outils classiques de contrôle. Dans les exemples qui suivent, les préfixes **TT** et **AL** font référence à « *True Type* » pour une police à chasse fixe, et à « *AL-gorithme* » pour une écriture similaire à celle utilisée dans les algorithmes.

Liste des commandes de type « *True Type* ».

1. **\TTif** donne **SI - SINON**.
2. **\TTfor** donne **POUR**.
3. **\TTwhile** donne **TANT QUE**.
4. **\TTrepeat** donne **RÉPÉTER JUSQU'À AVOIR**.

5. \TTswitch donne SUIVANT CAS.

Liste des commandes de type « *algorithme* ».

1. \ALif donne **Si – Sinon**.
2. \ALfor donne **Pour**.
3. \ALwhile donne **Tant Que**.
4. \ALrepeat donne **Répéter Jusqu'à Avoir**.
5. \ALswitch donne **Suivant Cas**.