# I. Cas d'utilisation en La TeX

# 1. Codes « $en\ ligne$ »

La macro \docilatex permet de taper du code en ligne via un usage similaire à \verb comme dans les deux exemples suivants.

- 1.  $\docilatex | \$a^b = c\$ | produit \$a^b = c\$$ .
- 2.  $\docilatex|\$a^b = c\$| vient de \docilatex+\docilatex|\$a^b = c\$|+$ .

Remarque. Le nom de la macro  $\setminus docilatex$  vient de « i-nline LATEX » soit « LATEX en ligne » en anglais.

# 2. Codes tapés directement

#### Exemple 1 – Face à face

```
\begin{doclatex}

$A = B + C$
\end{doclatex}
```

Ceci donne:

$$\$A = B + C\$$$
  $A = B + C$ 

# Exemple $2 - \hat{A}$ la suite

Via \begin{doclatex-flat} ... \end{doclatex-flat}, on obtient un code à plat \circ comme ci-dessous.

```
\$A = B + C\$
A = B + C
```

### Exemple 3 – Juste le code

 $\begin{doclatex-alone} \dots \end{doclatex-alone} affiche le code seul $^2$ comme ciaprès.$ 

```
$A = B + C$
```

## 3. Codes importés

Pour les codes suivants, on considère un fichier examples/listing-xyz.tex dont le chemin est donné relativement au document présent. Le contenu de ce fichier est l'unique ligne x y z = 1.

<sup>1.</sup> Le suffixe flat signifie « plat » en anglais.

<sup>2.</sup> Le suffixe alone signifie  $\ll seul \gg$  en anglais.

Notez que les 1<sup>res</sup> macros sont nommées de façon similaire aux environnements précédents en ajoutant le préfixe input tout en ignorant les tirets.

Remarque. Il faut savoir que les macros imprimant automatiquement du texte tiennent code de la langue choisie lors du chargement du package bdoc.

#### Exemple 1 – Face à face

\inputdoclatex{examples/listing-xyz.tex}

Ceci produit la mise en forme suivante.

$$xyz = 1$$

# Exemple $2 - \lambda$ la suite

\inputdoclatexflat produit un code à plat comme ci-dessous.

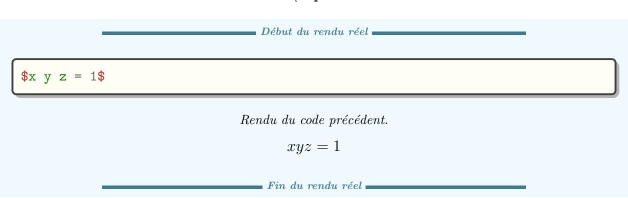
$$\begin{cases} \$x \ y \ z = 1\$ \\ xyz = 1 \end{cases}$$

### Exemple 3 – Juste le code

\inputdoclatexalone sert à n'avoir que le code comme ci-après.

## Exemple 4 – Code suivi du rendu centré

Le rendu suivant est obtenu en utilisant \inputdoclatexbefore.



Il est possible de changer le texte entre le code et son rendu via un argument optionnel. Ainsi \inputdoclatexbefore[Voici ce que cela donne.]{...} aboutit au résultat suivant.

x y z = 1

Voici ce que cela donne.

$$xyz = 1$$

lacksquare  $Fin\ du\ rendu\ rcute{e}el \ll {
m personnalise} \gg lacksquare$ 

#### Exemple 5 – Rendu centré suivi du code

Le rendu suivant<sup>3</sup>, similaire au précédent, est obtenu en appellant \inputdoclatexafter au lieu de \inputdoclatexbefore.

Début du rendu réel

$$xyz = 1$$

Le rendu précédent a été obtenu via le code suivant.

x y z = 1

Fin du rendu réel

Via  $\invertext{input}doclatexafter[Cette formule se tape comme suit.]{...}, on obtient le résultat ci-après.$ 

 $D\acute{e}but\ du\ rendu\ r\acute{e}el \ll {
m personnalis\'e} \gg$ 

$$xyz = 1$$

Cette formule se tape comme suit.

x y z = 1

lacksquare  $Fin\ du\ rendu\ rcute{e}l \ll ext{personnalise} \gg lacksquare$ 

#### Exemple 6 – Code importé et son rendu réel

Pour un code et son rendu réel non centré, on utilisera \inputdoclatexreal qui va produire ce qui suit

<sup>3.</sup> Il faut savoir que le  $1^{er}$  espace vertical disgracieux vient de l'emploi de  $\texttt{begin}\{\texttt{center}\}$  ...  $\texttt{end}\{\texttt{center}\}$  en coulisse.

Début du rendu dans cette doc.
\$x y z = 1\$
Ceci donne:
Début du rendu réel
xyz = 1
Fin du rendu réel
Fin du rendu dans cette doc.
Là aussi le texte par défaut peut être « $personnalisé$ ». Par exemple, ce qui suit a étét obtenu en tapant \inputdoclatexreal[On obtient le rendu réel ci-après.] $\{\}$ .
$D\acute{e}but\ du\ rendu$ « personnalisé » $dans\ cette\ doc.$
\$x y z = 1\$
On obtient le rendu réel ci-après.
$D\'ebut\ du\ rendu\ r\'eel$
xyz = 1
Fin du rendu réel

lacksquare  $Fin\ du\ rendu$  « personnalisé »  $dans\ cette\ doc.$