

## 1. Les blocs d'instruction

L'organisation des blocs d'informations se fait en respectant les règles suivantes.

1. Les commentaires se font uniquement à la sauce L<sup>A</sup>T<sub>E</sub>X via `% Mon commentaire` par exemple.
2. Les lignes vides sont autorisées.<sup>1</sup>
3. La syntaxe pour les contenus est la suivante.
  - Les différents contenus sont séparés par des points-virgules, les retours à la ligne étant autorisés.
  - Les espaces initiaux et finaux sont ignorés (voir la note 1 de bas de page).
  - Tout contenu est associé à un contexte.
  - Un nouveau contexte s'indique en début de contenu via `nom_contexte = contenu_1 ; ...` sans nécessité d'indiquer à chaque fois le contexte.
  - **Aucun contexte par défaut n'est initialement défini.**
4. Le point-virgule est optionnel à la fin du tout dernier contenu.

Voici un code fictif illustrant les règles précédentes.

```
% Commentaire no.1
ctxt_1 = cont. no.1 ;

% Commentaire no.2
ctxt_2 = cont. no.2-a ;
        cont. no.2-b ;
        cont. no.2-c

% Plus de contenu.
```

Voici la version verbeuse, un peu trop, du code précédent.

```
% Commentaire no.1
ctxt_1 = cont. no.1 ;

% Commentaire no.2
ctxt_2 = cont. no.2-a ;
ctxt_2 = cont. no.2-b ;
ctxt_2 = cont. no.2-c

% Plus de contenu.
```

Il est possible d'écrire des contenus à la suite comme dans l'exemple suivant.<sup>2</sup>

```
% Commentaire no.1
ctxt_1 = cont. no.1 ;

% Commentaire no.2
ctxt_2 = cont. no.2-a ; cont. no.2-b ; cont. no.2-c
```

On peut même utiliser l'horreur suivante.<sup>3</sup>

```
ctxt_1=cont. no.1;ctxt_2=cont. no.2-a;cont. no.2-b;cont. no.2-c
```

---

1. Ceci est très pratique pour aérer et organiser le code.

2. Nous déconseillons ce style de codage car cela nuit à la lisibilité des instructions.

3. Ce mode de rédaction est en fait idéal pour la fabrication automatisée de codes par un script extérieur.