

De l'indentation correcte des blocs *If...else*

Martin Vassor

October 2, 2016

Abstract

Alors que l'indentation est une aide aujourd'hui indispensable pour comprendre la sémantique d'un programme, le fait qu'elle soit optionnelle (dans la plupart des langages) a induit en erreur des groupes entiers. À tel qu'on peut aujourd'hui lire du code où l'indentation erronée induit le lecteur en erreur. Un exemple des plus frappants est la mauvaise habitude partagée par la majorité de la communauté de l'indentation de blocs *if...else if ... else*. Voyons pourquoi le style qui veut mettre les trois éléments au même niveau est faux et contraire à la sémantique binaire des *if*.

1 Introduction

Le but de l'indentation est de faciliter la lecture du code. Ainsi, un ensemble de lignes constituant un bloc d'exécution linéaire sont alignées, et lorsque le flot d'exécution est modifié, on augmente le décalage. Ainsi, par exemple une boucle *for* a l'indentation suivante :

Ainsi, la syntaxe utilisée va dans le sens de la sémantique, à savoir que le bloc *loop statement* constitue le corps de la boucle. L'indentation utilisée indique que les trois lignes constituent un bloc d'exécution linéaire, dont le flot concernant la ligne centrale est modifié. De même, la ligne 2 constitue un bloc d'exécution linéaire, et toutes les lignes constituant le bloc sont indentées par rapport aux lignes 1 et 3.

```
1: For <statement> to <end condition> do
2:     <loop statement>
3: <Next statement>
```

Figure 1: Indentation des *for*

```
1: if <condition>
2:     <conditionnal statement>
3: else
4:     <conditionnal statement>
5: <unconditionnal statement>
```

Figure 2: Indentation des *if...else*

```
1:  if <condition>
2:      {
3:      <conditionnal statement1>
4:      ...
5:      }
6:  else
7:      {
8:      <conditionnal statement1>
9:      ...
10:     }
```

Figure 3: Les accolades font partie du statement (respect de la sémantique du C) et sont donc indentés

2 Le cas du *if...else*

2.1 Cas général

Pour le *if...else*, les styles utilisés proposent en général quelque chose similaire à l'exemple en Figure 2. Et cette syntaxe va dans le sens que l'on veut, chaque niveau d'indentation représente un bloc linéaire.

2.2 Élision des accolades

On assiste à trois écoles concernant l'indentation des accolades. En effet, certains langages de programmation (dont les plus classiques comme le C) délimitent un unique statement un bloc encadré d'accolades. Cette syntaxe permet d'exécuter plusieurs instructions dans un seul statement. Les trois placements les plus courant pour les accolades dans un bloc sont indiqués dans les figures 3, 4 et 5. La syntaxe utilisée en 3 s'explique un accord strict de la sémantique du C et de notre sémantique d'indentation. On observe la syntaxe 4 par abus ou par faignandise. La syntaxe 5 s'explique par le gain de ligne, gain d'autant plus important au temps des éditeurs en lignes tels que `ed`.

```
1:  if <condition>
2:  {
3:      <conditionnal statement1>
4:      ...
5:  }
6:  else
7:  {
8:      <conditionnal statement1>
9:      ...
10: }
```

Figure 4: Les accolades ne font pas partie du statement (différent de la sémantique du C) et ne sont donc pas indentés

```
1: if <condition> {
2:     <conditionnal statement1>
3:     ...
4: } else {
5:     <conditionnal statement1>
6:     ...
7: }
```

Figure 5: Les accolades sont rajoutées en début et fin de lignes des lignes entourant les statements conditionnels.

```
1:  if <condition>
2:      <conditionnal statement1>
3:  else
4:      if <condition>
5:          <conditionnal statement2>
6:      else
7:          <conditionnal statement3>
```

Figure 6: Indentation correcte de structures if...else imbriquées.

```
1:  if <condition>
2:      <conditionnal statement1>
3:  else if <condition>
4:      <conditionnal statement2>
5:  else
6:      <conditionnal statement3>
```

Figure 7: Indentation incorrecte de structures if...else imbriquées.

3 *if...else* imbriqués

Une structure *if...else* est un statement en soit. De fait, en accordant l'indentation des accolades à la sémantique du langage lors de structures imbriquées, on devrait avoir une indentation semblable à celle en Figure 6. Force est de constater que cette indentation n'est que trop rarement utilisée, au profit d'autres faisant fi du lien qu'il y a entre l'indentation et la sémantique du langage (voir Figure 7).