

# Ordered lists in L<sup>A</sup>T<sub>E</sub>X using the `enumerate` environment

Jorge A. R. Duran  
jorge.a.r.duran@gmail.com

Samuel G. R. Duran  
samuelbrasil96@hotmail.com.br

May 15, 2020

## Contents

<a href="#">1 Ordered Lists using the <code>enumerate</code> environment</a>	<a href="#">1</a>
<a href="#">2 Modifying the numbering styles</a>	<a href="#">3</a>
<a href="#">3 Placing some fixed test in a nested list</a>	<a href="#">8</a>
<a href="#">4 Including section number in list number</a>	<a href="#">9</a>
<a href="#">References</a>	<a href="#">9</a>

## 1 Ordered Lists using the `enumerate` environment

Under the `\enumerate` environment, ordered lists are generated using the control sequence `\item` in each entry. Each enumerated item can be labeled and referred through the `\label{ }` and `\ref{ }` commands, respectively. The default **numbering styles** in the nested four listings under the `enumerate` environment are:

- 1, 2, ... 4 for the first level (arabic numbers)
- (a), (b) ... (d) for the second level (lowercase letters)
- i, ii, ... iv for the third level (lowercase Roman numerals)
- A, B ... C for the fourth level (uppercase letters)

while their **referring styles** are:

- 1, 2, ... 4 for the first level

- 1(a), 1 (b) ... 1(d) for the second level
- 1(a)i, 1(a)ii, ... 1(a)iv for the third level
- 1(a)iA, 1(a)iB ... 1(a)iC for the fourth level

We will use as an example of nested list up to the fourth depth level, a list consisting of countries, states, regions and cities from our continent. The L<sup>A</sup>T<sub>E</sub>X code and its output are shown side by side with the help of package `fancyvrb-ex` [2]. This will be done with all the examples in this article.

	1	<code>\begin{enumerate}</code>
	2	<code>\item Brazil \label{it:br}</code>
1. Brazil	3	<code>\begin{enumerate}</code>
	4	<code>\item Rio de Janeiro \label{it:rj}</code>
(a) Rio de Janeiro	5	<code>\item Sao Paulo \label{it:sp}</code>
	6	<code>\item Parana \label{it:pr}</code>
(b) Sao Paulo	7	<code>\begin{enumerate}</code>
(c) Parana	8	<code>\item Oeste Par. \label{it:oest_par}</code>
i. Oeste Par.	9	<code>\item Centro-Sul Par. \label{it:met_cur}</code>
ii. Centro-Sul Par.	10	<code>\begin{enumerate}</code>
A. Curitiba	11	<code>\item Curitiba \label{it:cur}</code>
B. S. J. dos Pinhais	12	<code>\item S. J. dos Pinhais \label{it:sjp}</code>
iii. Centro-Sul Par.	13	<code>\end{enumerate}</code>
iv. Sudoeste Par.	14	<code>\item Centro-Sul Par. \label{it:csp}</code>
(d) Bahia	15	<code>\item Sudoeste Par.</code>
(e) Mato Grosso do Sul	16	<code>\end{enumerate}</code>
	17	<code>\item Bahia</code>
	18	<code>\item Mato Grosso do Sul</code>
2. Cuba	19	<code>\end{enumerate}</code>
(a) Villa Clara	20	<code>\item Cuba \label{it:cub}</code>
(b) Havana	21	<code>\begin{enumerate}</code>
	22	<code>\item Villa Clara</code>
	23	<code>\item Havana</code>
3. Argentina	24	<code>\end{enumerate}</code>
	25	<code>\item Argentina</code>
	26	<code>\end{enumerate}</code>

If at any part of the current document we need to refer to the item `S. J. dos Pinhais`, for example, which is at the deepest level of our list (the fourth), we can invoke the `\ref{}` command using the corresponding `label` we associated to this item in L<sup>A</sup>T<sub>E</sub>X code, in this

case `{it:sjp}`, as the command's argument. The same logical is applied to indicate the region, state and country where S. J. dos Pinhais is located as shown below:

<p>The city <code>1(c)iiB</code> is located at the region of <code>1(c)ii</code> in the state <code>1c</code> that belong to the country <code>1</code>.</p>	<pre> 1 The city \ref{it:sjp} is located 2 at the region of \ref{it:met_cur} 3 in the state \ref{it:pr} that belong 4 to the country \ref{it:br}. </pre>
--	--

## 2 Modifying the numbering styles

In  $\LaTeX$  lists each level has an associated `\labelenumXX` and `\theenumXX` variables, where  $XX = i \dots iv$  point for the level. The `enumXX` is the counter that increase its value at each call to the `\item` inside of the `enumerate` environment. Let's check the current values of these variables at level one:

<ul style="list-style-type: none"> <li>• <code>\theenumi = 3</code></li> <li>• <code>\labelenumi = 3.</code></li> </ul>	<pre> 1 \begin{itemize} 2   \item \verb!\theenumi! = \theenumi 3   \item \verb!\labelenumi! = 4   \labelenumi 5 \end{itemize} </pre>
---	--

Since the default numbering styles for the first level are arabic numbers, the first time that  $\LaTeX$  code found an `\item` inside the `enumerate` environment the counter `enumi` is set to the value 1. But this happens three times at the first level in our example list (it has three countries) so a call to the value stored in the counter, which is made through `\theenumi` returns a value of 3. The `\labelenumi` just adds a lower point to `\theenumi`. The same logical applies to the second level, as shown below:

<ul style="list-style-type: none"> <li>• <code>\theenumii = b</code></li> <li>• <code>\labelenumii = (b)</code></li> </ul>	<pre> 1 \begin{itemize} 2   \item \verb!\theenumii! = \theenumii 3   \item \verb!\labelenumii! = 4   \labelenumii 5 \end{itemize} </pre>
--	--

The `enumii` has counted only from `a \dots b` since in the example list only two countries (Brazil and Cuba) have some of their states listed. The numbering styles for each depth

can be modify by the user through the `\renewcommand{label}{style}` command, where `label` is the list depth being modified and `style` is how you want that number to be shown. We can styled both, the `\labelenumXX` and the `\theenumXX` variables using the following macros:

- `\alph{number}` : lowercase letters
- `\Alph{number}` : uppercase letters
- `\arabic{number}` : numbers
- `\roman{number}` : lowercase roman numerals
- `\Roman{number}` : uppercase roman numerals

For example, let's change the number style to uppercase roman numerals at the first depth of our example list. This can be accomplished through the command

```
\renewcommand{\labelenumi}{\Roman{enumi}:}
```

If we invoke the new values for the controlling variables, this is the result:

<ul style="list-style-type: none"> <li>• <code>\theenumi = 3</code></li> <li>• <code>\labelenumi = III:</code></li> </ul>	<pre> 1 \begin{itemize} 2   \item \verb!\theenumi! = \theenumi 3   \item \verb!\labelenumi! = 4     \labelenumi 5 \end{itemize} </pre>
---	--

We have styled only the `\labelenumi` variable but the `\theenumi` remains counting using the default arabic numbers. The result of invoking a **cross reference** to Cuba e.g. which has the `\label{it:cub}` is the following:

<pre>\ref{it:cub} = 2</pre>	<pre> 1 \begin{itemize} 2   \item[] \verb!\ref{it:cub}! = 3     \ref{it:cub} 4 \end{itemize} </pre>
-----------------------------	---

In spite of the fact that Cuba is now identified by a II: in the example list, as can be easily verified reprinting the original code:

	1	<code>\begin{enumerate}</code>
	2	<code>\item Brazil \label{it:br}</code>
I: Brazil	3	<code>\begin{enumerate}</code>
	4	<code>\item Rio de Janeiro \label{it:rj}</code>
(a) Rio de Janeiro	5	<code>\item Sao Paulo \label{it:sp}</code>
(b) Sao Paulo	6	<code>\item Parana \label{it:pr}</code>
(c) Parana	7	<code>\begin{enumerate}</code>
i. Oeste Par.	8	<code>\item Oeste Par. \label{it:oest_par}</code>
ii. Centro-Sul Par.	9	<code>\item Centro-Sul Par. \label{it:met_cur}</code>
A. Curitiba	10	<code>\begin{enumerate}</code>
B. S. J. dos Pinhais	11	<code>\item Curitiba \label{it:cur}</code>
iii. Centro-Sul Par.	12	<code>\item S. J. dos Pinhais \label{it:sjp}</code>
iv. Sudoeste Par.	13	<code>\end{enumerate}</code>
(d) Bahia	14	<code>\item Centro-Sul Par. \label{it:csp}</code>
(e) Mato Grosso do Sul	15	<code>\item Sudoeste Par.</code>
	16	<code>\end{enumerate}</code>
II: Cuba	17	<code>\item Bahia</code>
	18	<code>\item Mato Grosso do Sul</code>
(a) Villa Clara	19	<code>\end{enumerate}</code>
(b) Havana	20	<code>\item Cuba \label{it:cub}</code>
	21	<code>\begin{enumerate}</code>
III: Argentina	22	<code>\item Villa Clara</code>
	23	<code>\item Havana</code>
	24	<code>\end{enumerate}</code>
	25	<code>\item Argentina</code>
	26	<code>\end{enumerate}</code>

In order to maintaining the equivalence between the true label of the entry in the list and its cross reference we should also styled `\theenumi` variable using this command (without the colon):

```
\renewcommand{\theenumi}{\Roman{enumi}}
```

Invoking again the position of Cuba in the list we have:

	1	<code>\begin{itemize}</code>
<code>\ref{it:cub} = 2</code>	2	<code>\item[] \verb!\ref{it:cub}! =</code>
	3	<code>\ref{it:cub}</code>
	4	<code>\end{itemize}</code>

Clearly the cross reference remains incorrect. Remember that the variable `\theenumi` is updated at each call to the command `\item` and this only can be done if we run again the code for our example list. But we will not do that because it will destroy all our previous demonstrations, unless we change the `\label` of each item. Indeed, the style of any cross reference to our example list will change to uppercase roman numerals in the whole document as it should, since that means that the `\ref{}` command always find at the whole document for the most recent value of the associate identifier.

Any of the forms we have seen in this section for modifying the numbering styles in nested lists should be valid unless any other instruction, placed right before of the `\begin{enumerate}` modify its effects. For example, consider the following list:

	1 <code>\begin{enumerate}</code>
	2 <code>\item Local approaches. \label{it:la}</code>
I: Local approaches.	3 <code>\begin{enumerate}</code>
(a) The NSIF Concept.	4 <code>\item The NSIF Concept.</code>
(b) The SED Concept.	5 <code>\item The SED Concept.</code>
	6 <code>\end{enumerate}</code>
II: Global approaches.	7 <code>\item Global approaches. \label{it:ga}</code>
(a) The NS Concept.	8 <code>\begin{enumerate}</code>
(b) The SS Concept.	9 <code>\item The NS Concept.</code>
	10 <code>\item The SS Concept.</code>
	11 <code>\end{enumerate}</code>
	12 <code>\end{enumerate}</code>

As expected, the labels at the first level depth remains using uppercase roman numerals. We can return to the default configuration placing the following command at at the beginning of the environment, as follows:

```
\renewcommand{\labelenumi}{\arabic{enumi}}
```

1 Local approaches.

(a) The NSIF Concept.

(b) The SED Concept.

2 Global approaches.

(a) The NSC Concept.

(b) The SS Concept.

```
1 \renewcommand{\labelenumi}{\arabic{enumi}}
2 \begin{enumerate}
3   \item Local approaches. \label{it:la}
4     \begin{enumerate}
5       \item The NSIF Concept.
6       \item The SED Concept.
7     \end{enumerate}
8   \item Global approaches. \label{it:ga}
9     \begin{enumerate}
10      \item The NSC Concept.
11      \item The SS Concept.
12    \end{enumerate}
13 \end{enumerate}
```

### 3 Placing some fixed text in a nested list

Quite often it is necessary to put some fixed text in each entry of a numbered list. More common examples include words as **Example**, **Question**, etc. This can be done by redefining the `\labelenumXX` commands. The following example consist of a hypothetical test with two questions, two items each. We will use only the first letter of the word **Question**. The `\renewcommand` is necessary only due to the introduction of the letter **Q** plus a call to the counter at the first level depth. Except for this letter, the rest of the code would be unnecessary because `\theenumi` is the default value for the `\labelenumi` in the `enumerate` environment.

```
\renewcommand{\labelenumi}{Q - \theenumi}
```

```
1 \renewcommand{\labelenumi}{Q - \theenumi}
2 \begin{enumerate}
Q - 1 Discourse on these local approaches: 3 \item Discourse on these local approaches:
4 \begin{enumerate}
5 \item The NSIF \label{it:nsif}
6 \item The SED \label{it:sed}
7 \end{enumerate}
Q - 2 Discourse on these global ap- 8 \item Discourse on these global approaches:
proaches: 9 \begin{enumerate}
10 \item The NSC \label{it:nsc}
11 \item The SSC \label{it:ssc}
12 \end{enumerate}
13 \end{enumerate}
```

If the label for the second item of the first question is `\label{it:sed}` we realize that the `cross reference` to this line does not include the letter **Q**:

```
\ref{it:sed} = 1b
1 \begin{itemize}
2 \item [] \verb!\ref{it:sed}! = \ref{it:sed}
3 \end{itemize}
```

This is not a main drawback since the only thing that we do for fixing is to insert the letter **Q** or even the entire word **Question** before calling the `\ref` command. There are other alternatives using, for example the `enumerate` or `enumitem` packages. Note that the names for the package and environment are the same (`enumerate`) but this is just a



coincidence.

## 4 Including section number in list number

We arrive to `\thesection = 4`, which obviously tell us that is the `LATEX` macro we should use in order to incorporate the section number to the `labels` and `counters` in the `enumerate` environment. Once again, the `\renewcommand` will be used to accomplish this task:

```
1 \renewcommand{\theenumi}{\thesection.\arabic{enumi}}
2 \begin{enumerate}
Q - 4.1 Discourse on these local approaches: 3 \item Discourse on these local approaches:
4 \begin{enumerate}
5 \item The NSIF \label{it:nsif1}
6 \item The SED \label{it:sed1}
7 \end{enumerate}
Q - 4.2 Discourse on these global ap- 8 \item Discourse on these global approaches:
proaches: 9 \begin{enumerate}
10 \item The NSC \label{it:nsc1}
11 \item The SSC \label{it:ssc1}
12 \end{enumerate}
13 \end{enumerate}
```

Since section 3 we add the letter Q to the `\labelenumi` of any `enumerate` environment and we know from the section 2 that this setup will be valid if no other modification is found before the `\begin{enumerate}`. For this reason the letter is preserved in the above list. Also note that we changed the `\label` of the entries at the second level. Can you guess the reason of that? If you answered *not to modify this cross reference*, you are right. Let's make a cross reference to the topic *SED* which stands for *Strain Energy Density* in the last version of our test example:

```
\ref{it:sed} = 4.1b
1 \begin{itemize}
2 \item [] \verb!\ref{it:sed}! =
3 \ref{it:sed1}
4 \end{itemize}
```

As expected, the cross reference now includes the section number.

## References

- [1] D. Datta. *L<sup>A</sup>T<sub>E</sub>X in 24 Hours: A practical Guide for Scientific Writing*. Springer International Publishing, Gewerbestrasse 11, 6330 Cham, Switzerland, 2017. ISBN 978-3-319-47830-2.
- [2] T. V. Zand. The fancyvrb package: Fancy Verbatims in L<sup>A</sup>T<sub>E</sub>X. Technical report, Princeton University, Princeton, USA, 2020. URL <https://ctan.org/pkg/fancyvrb>.