

Counting with \LaTeX counters ¹

S. Parthasarathy
drpartha@gmail.com

Ref.: counters1.tex
Version Code : 20120329e

Abstract

Counting is a frequently encountered activity. \LaTeX offers many pre-defined counters (e.g. page number, section number etc.). In addition to the built-in counters, one may require some special user-defined, user-managed counters. This article is a quick overview of user-defined, user-managed counters in \LaTeX .

1 \LaTeX Counters

\LaTeX provides several builtin counters which keep track of various entities used by you. These are provided by \LaTeX and automatically managed by \LaTeX . Examples of such counters are ::

section numbers, subsection numbers, equation numbers, page numbers, enumerated lists, bibliography entries, footnote numbers etc.

In addition to the above, you may like to keep a count of various objects specific to your document. In \LaTeX it is fairly easy to create new counters and even counters that reset automatically when another counter is increased (think subsection in a section for example).

You may need this feature if you write documents which are full of countable objects. For instance, you may have :

¹This report is accessible on the web at <http://www.freewebs.com/profpartha/publications/downloadables.htm#maths>. The \LaTeX source of this document, as well as this rendered file (pdf version), may be obtained by sending a request to drpartha@gmail.com. Please quote the Reference code and the Version code given above.

Examples, Theorems, Lemmas, Corollaries, Hypothesis, Conjecture, conclusion, assertion/assumption, Definition

You may like to number each of the above independently, and increment or decrement this count (for each object). You may need numbers for cross-referencing these objects.

To setup and manage counters, use ::

- With `\newcounter{NameOfTheNewCounter}` you create a new counter that is automatically set to zero.
- If you want the counter to be reset to zero every time another counter is increased, use:
`\newcounter{NameOfTheNewCounter}[NameOfTheOtherCounter]`
- To increase the counter, either use
`\stepcounter{NameOfTheNewCounter}`
or
`\refstepcounter{NameOfTheNewCounter}`
or
`\addtocounter{NameOfTheNewCounter}{number}`
here the number can also be negative. For automatic resetting you need to use `\stepcounter` .
- To set the counter value explicitly, use
`\setcounter{NameOfTheNewCounter}{number}`

There are several ways to get access to a counter.

- `\theNameOfTheNewCounter` will print the formatted string related to the counter.
- `\value{NameOfTheNewCounter}` will return the counter value which can be used by other counter, or for calculus. It is not a formatted string, so it cannot be used in text.
- `\arabic{NameOfTheNewCounter}` will print the formatted counter using arabic numbers.

Instead of `\arabic` you could also use `\alph`, `\Alph`, `\roman`, or `\Roman`. Note that `\arabic{NameOfTheNewCounter}` may be used as a value too, but not the others.

1.1 Working with counters

Let us start by creating three counters : Doctors, Engineers, Farmers (D,E,F in short). We use `\newcounter{NameOfTheNewCounter}` This is an invisible operation, you see nothing on your rendered text. So, do not panic !

Let us see what value they contain, using `\arabic{NameOfTheNewCounter}`

Doctor 0 , Engineer 0, Farmer 0

Let us add some doctors (4), engineers (6), and farmers (5) to our crowd, using `\addtocounter{NameOfTheNewCounter}{number}`

How many of each, do we have now ?

Doctor 4 , Engineer 6, Farmer 5

Imagine the day when an engineer becomes a farmer.

How many of each, do we have now ?

Doctor 4 , Engineer 5, Farmer 6

Imagine the day when the farmer's son also becomes a farmer.

Now, we have 7 farmers.

Let us see the number of farmers, using Roman numerals :: We have VII farmers in this village. We have vii farmers in this village.

Or, see the same as (English) letters :: We have G farmers in this village. We have g farmers in this village. (G is the seventh letter of the English alphabet)

We can make things much more enjoyable by defining newcommands which will give a shortcut way to get the above results.

```
\newcommand{\showengineer}{\arabic{Engineer}}
\newcommand{\incengineer}{\addtocounter{Engineer}{1}}
\newcommand{\decengineer}{\addtocounter{Engineer}{-1}}
```

We have 5 engineers now
Let us increment Engineer by one.
We have 6 engineers now
Let us decrement Engineer by one.
We have 5 engineers now

We can do some simple arithmetic too, with counters. We have just now 4 doctors, and 7 farmers. Using a temporary counter sumdf, and

```
\newcounter{sumdf}
\setcounter{sumdf}{\value{Farmer}+\value{Doctor}}
```

we can get the sum of two counters. Remember, we need the package “calc”, to do this in line arithmetic.

So, we have 11 farmers and doctors, put together.

Now, we will see a simple common usage of counters. Imagine making a list of engineers in our community. Let us say, we start with a small number of engineers and make a list of them. We define a new counter, and a new command ::

```
\newcounter{Student}
\addtocounter{Student}{1}
\newcommand{\itemstudent}{Student\#\arabic{Student}\addtocounter{Student}{1}}
```

Student #1 : Likes pizzas with hot pepper topping.
Student #2 : Plays the piano. Specialises in Chopin oeuvres.
Student #3 : Mathematician. Genius
Student #4 : Gives coaching lessons in football.

This behaves like an enumerated list. Now let us try to add another student after Student #2.

But, first we have to restart the counter from 1.

Student #1 : Likes pizzas with hot pepper topping.
Student #2 : Plays the piano. Specialises in Chopin oeuvres.

Student #3 : Owns a poultry farm
Student #4 : Mathematician. Genius
Student #5 : Gives coaching lessons in football.

Notice that the numbering has changed. We have not kept count of the Students explicitly. This saves us a lot of headache , especially when the list is very long or scattered.

You get the idea of how all this works ? It is now up to you, to use this tool, the way you want. Because we have not hard coded any value to any counter, you can dynamically increment and decrement values, and do some simple arithmetic too. \LaTeX does the book-keeping for you.

2 Concluding Remarks

Copyright policy :: This document is distributed under a Creative-commons "Attribution + Noncommercial + ShareAlike (by-nc-sa)" license. This license is described in ::

http://en.wikipedia.org/wiki/Creative_Commons_Attribution_2.5#Original_licenses

An official description of the "Attribution + Noncommercial + ShareAlike (by-nc-sa)" license is given in :

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

If you have any queries, or need clarifications, you can ask drpartha@gmail.com.

This report was typeset in \LaTeX by the author (using Kile, under a Suse Linux system). You can obtain the \LaTeX source of this report, by sending a request to the author drpartha@gmail.com .

The author welcomes comments, suggestions and remarks on this document. Please report any flaws you may find, to drpartha@gmail.com.

* * *