

**Title of script: New Commands**

**Author: Ambika Vanchinathan**

**Keywords: New commands, renewcommand, define commands in LaTeX, \newcommand, \renewcommand, video tutorial**

Visual Cue	Narration
Slide	Welcome to the spoken tutorial on creating new <b>commands</b> in <b>LaTeX</b> .
Slide: <b>Learning Objectives</b>	This tutorial will help the learners to <ul style="list-style-type: none"> <li>• create or define new <b>commands</b> to get customized output.</li> <li>• redefine the existing <b>commands</b>.</li> </ul>
Slide: <b>System Requirements</b>	To record this tutorial, I am using <ul style="list-style-type: none"> <li>• <b>Ubuntu Linux 14.04</b> operating system</li> <li>• <b>TeXWorks 0.5</b></li> </ul>
Slide: <b>Pre-requisites</b>	To follow this tutorial, <ol style="list-style-type: none"> <li>1. You should have knowledge of <b>commands</b> that are necessary to create a document in <b>LaTeX</b>.</li> <li>2. You should also have knowledge of <b>compiling</b> and producing a <b>PDF output</b> in <b>LaTeX</b>.</li> <li>3. If not, please go through the relevant <b>LaTeX</b> spoken tutorials on this website.</li> </ol>
slide: <b>New command</b>	We all know that <b>commands</b> are special words that are provided to the <b>compiler</b> .  <b>Commands</b> start with a <b>backslash</b> and sometimes accept <b>parameters</b> .  Example: <code>\noindent</code> , <code>\textbf</code> , <code>\centering</code>  We can define our own custom <b>commands</b> in <b>LaTeX</b> to save time or for the code to look cleaner.
slide: <b>New command</b>	Ensure that the custom <b>command</b> you create is not an already existing <b>LaTeX command</b> .  To define a new <b>command</b> , we write <code>\newcommand{command}{definition}</code>

	at the beginning of the document.
Open the <b>TeXWorks</b> editor	Let us open the <b>TeXWorks editor</b> .
(copy/paste the following in TeXWorks)  <code>\documentclass{article}</code> <code>\usepackage{amsfonts}</code> <code>\newcommand{\bbr}{\mathbb R}</code> <code>\begin{document}</code> Let $\bbr$ be a set of Rational numbers. <code>\end{document}</code>	Copy the program given here into the <b>tex</b> file.
	Let's try to understand the program.
Point to the exact line as per narration	The definition <code>\newcommand{\bbr}{\mathbb R}</code> means that <code>\bbr</code> is the <b>newcommand</b> .  And when used, it will typeset <b>R</b> in the <b>mathbb font</b> .
Point to the exact line as per narration	Note that the required <b>packages</b> are loaded in the preamble; in this case <code>\usepackage{amsfonts}</code>
Point to the exact line as per narration	So, instead of <code>{\mathbb R}</code> , we can use <code>\bbr</code> within <b>dollars</b> while typesetting.
Save as <b>newComm.tex</b>	Let us save the file as <b>newComm.tex</b>
Compile >> Point to the letter R in the <b>pdf</b> output	On compilation, I get the <b>pdf</b> output.  Observe the letter <b>R</b> in <b>mathbb font</b> .
	<<PAUSE>>
slide: <b>New command with parameters</b>	Now, we will see <b>newcommand</b> with <b>parameters</b> .  It is typically defined as - <code>\newcommand{\command}[parameters]{definition}</code>
open a new file in <b>TeXWorks</b>	Let's open a new file in <b>TeXWorks</b> .
(copy/paste the following)	Now, we will copy the code given here into our <b>TeX</b> file.

<pre>\documentclass{article} \usepackage{amsmath} \newcommand{\bb}[1]{\mathbb {#1}} \begin{document} Let <math>\mathbb{R}</math> be a set of Rational numbers and <math>\mathbb{Z}</math> be the set of Complex numbers. \end{document}</pre>	
<p>Point to the exact line as per narration</p>	<p>Let us understand the code now.</p> <p>We have  <code>\newcommand{\bb}[1]{\mathbb {#1}}</code>  before the <b>begin document statement</b>.</p>
<p>Point to the exact line as per narration</p>	<p>Here <b>\bb</b> is the <b>command</b>.</p> <p><b>[1]</b> means that the command has one parameter</p>
<p>Point to the exact line as per narration</p>	<p><code>\mathbb {#1}</code> defines the <b>command</b>.</p> <p><b>#1</b> means the first parameter.</p>
<p>Point to the exact line as per narration</p>	<p>Carefully observe the usage after <code>\begin{document}</code></p> <p>We use it within <b>dollars</b> because it contains <b>math symbols</b>.</p>
<p>Point to the exact line as per narration</p>	<p><b>\bb</b> followed by the <b>open brace</b> and <b>parameter</b> and again we close the <b>brace</b>.</p>
<p>Point to the exact line as per narration</p>	<p>We have used it twice with <b>R</b> as the <b>parameter</b> in the first usage  and <b>Z</b> as the <b>parameter</b> in the second usage.</p>
<p>Save as <b>newCommP1.tex</b>  Compile</p>	<p>Save the program as <b>newCommP1.tex</b> and compile it.</p>
<p>And point to the pdf output</p>	<p>In the output, observe that the <b>R</b> and <b>Z</b> are in the <b>mathbb</b> or the <b>blackboard font</b>.</p>
	<p>&lt;&lt;PAUSE&gt;&gt;</p>
	<p>Next, let's define a <b>newcommand</b> with <b>2 parameters</b>.</p>

Open a new <b>tex</b> file.	Open a new <b>tex</b> file.
(copy/paste the following)  <pre>\documentclass{article} \newcommand{\add}[2]{\left( #1+#2\right)}</pre> <pre>\begin{document} Adding abc and xyz we get \$\add{abc}{xyz}\$ \end{document}</pre>	Type the code as shown here.
Point to the exact line as per narration	Observe the line $\newcommand{\add}[2]{\left( #1 + #2 \right) }$ in the document.
Point to the exact line as per narration	This has been used as $\$add{abc}{xyz}\$$  Here in the definition, $\backslashadd$ is the <b>newcommand</b> .
Point to the exact line as per narration	<b>[2]</b> indicates that it has <b>2 parameters</b> .
Point to the exact line as per narration	$\{\left( #1+#2\right)\}$ is the definition of the <b>command</b> .
Point to the exact line as per narration	<b>#1</b> for the first <b>parameter</b> and <b>#2</b> for the second <b>parameter</b> .
Point to the exact line as per narration	<b>#1</b> and <b>#2</b> are replaced by the actual text $\{abc\}$ and $\{xyz\}$ in this case.
Save as <b>newCommP2.tex</b> Compile	Let us save as <b>newCommP2.tex</b> and compile the <b>tex</b> file.
	Observe the output.  Here the <b>left</b> and <b>right parenthesis</b> are added and a <b>plus symbol</b> is introduced between the first and second <b>parameters</b> .
slide: <b>Renewcommand</b>	<b>Renewcommand</b> is used to change or overwrite the existing <b>command</b> .  The syntax for <b>renewcommand</b> is $\backslashrenewcommand{\command}{definition}$

	It's quite similar to <b>newcommand</b> .
(copy/paste the following)  <pre>\documentclass{article} \usepackage{amsmath} \renewcommand{\S}{\mathcal {S}} \begin{document} Let \$\S\$ be a set. \end{document}</pre>	Type the code as shown to redefine an existing <b>command</b> using <b>renewcommand</b> .  Let us understand the program
Point to the exact line as per narration	The definition <b>\renewcommand{\S}{\mathcal {S}}</b> redefines <b>\S</b> .  <b>\S</b> normally produces the section symbol (§) whereas after redefining <b>\S</b> , produces <b>S</b> in <b>Calligraphic font</b>  Note that the required package <b>amsmath</b> is preloaded.  Also, we use <b>\S</b> within <b>dollar symbols</b> .
Save as <b>renew.tex</b> Compile Point to the " <b>S</b> " in the <b>PDF</b> output.	Let us now save the file as <b>renew.tex</b> , compile it and see the output.
Replace <b>\z</b> for <b>\S</b> , compile and point out the error.	Note that replacing <b>z</b> for <b>S</b> in the <b>renewcommand</b> does not work.  This is because there is no existing <b>command</b> defined as <b>\z</b> .  Please use <b>renewcommand</b> only when you are very sure about what you are doing.
Slide: <b>Summary</b>	To summarise, we have learnt <ul style="list-style-type: none"> <li>• <b>newcommand</b></li> <li>• <b>newcommand</b> with <b>parameters</b></li> <li>• <b>renewcommand</b></li> </ul>
slide: <b>Assignment</b>	Create a new <b>command</b> <b>\textbfit</b> which will convert the given text into bold and italics.
Slide: <b>Acknowledgement</b>	The video at the following link summarises the Spoken Tutorial project. Pls watch it.

Slide	<p>The Spoken Tutorial Project Team :</p> <p>Conducts workshops using spoken tutorials and Gives certificates to those who pass an online test For more details, please write to us.</p>
Slide	<p>Spoken Tutorial Project is funded by the NMEICT, MHRD, Government of India.</p> <p>More information on this Mission is available at <a href="#">this link</a>.</p> <p>This is Ambika Vanchinathan signing off. Thanks for joining.</p>