Introduction to LaTeX

Lecture 5: Advanced usages of LaTeX

Liu Yihao

SJTU-UMJI Technology Department

February 25, 2018

Newcommand

Sometimes you are building a huge project (like this lecture), and you may use certain type of syntax for many many times. Now it's time to define your own command with \newcommand in the beginning of the document (where the \usepackage commands appear).

Command

\newcommand{\yourcommand}[arg_num]{code}

- arg_num - number of arguments in your command
- code - the code of your command, use #1, #2, . . . , #n to represent the arguments

Example

\newcommand{\samplecommand}[1]{\alert{\textbackslash #1}}

It is defined to simply display the commands in red in this lecture.

Renewcommand

Another times you need to redefine the commands, then \renewcommand can be used. It's very similar to \newcommand, the only difference is that you must use \newcommand when the command doesn't exists, while using \renewcommand when the command has been defined (by you or LaTeX packages) before.

Command

```
\renewcommand{\definedcommand}[arg_num]{code}
```

Example

```
\renewcommand{\thesection}{\Roman{section}}
\renewcommand{\thesubsection}{\Alph{subsection}}
```

By default, the number before the section titles of \section is 1, 2, 3, etc, this command will change them to a capital form of roman numbers, I, II, III, etc. And subsection numbers become A, B, C, etc.

New/Renewenvironment

Environments can also be defined.

Command

```
\newenvironment{name}[arg_num]{begdef}{enddef}
\renewenvironment{name}[arg_num]{begdef}{enddef}
```

- name - the name of your environment
- arg_num - number of arguments in your environment
- begdef - the code to substitute the begin clause of your environment
- enddef - the code to substitute the end clause of your environment

Example

```
\newenvironment{command}{\begin{block}{Command}}{\end{block}}
```

Include and Input

When you are building a huge project, if you write all of the code in a single file, the compiling of the whole project will be very slow, and the length of the file will also confuse you. Then you can use \include and \input to avoid this.

Command

\include{file} - Include the file on a new page, the files are compiled separately.
\input{file} - Directly replace the command with the whole file, doesn't start a new page, but the compiling won't speed up.

If you are including a .tex file, then the extension name can be omitted. Another command \includeonly{list} can be added to the beginning of the document, so that only the include files in list are compiled and others are ignored, this is very useful in debugging huge projects.

Hyperlink

Hyperlinks are supported in LaTeX, use the hyperref package.

Command

```
\usepackage{hyperref}
\hypersetup{options}
\url{url}
\href{url}{text}
```

Some common options are listed below:

- colorlinks - boolean (default false)
- urlcolor - color for linked URLs (default magenta)
- linkcolor - color for normal internal links (default red)

Listings

Sometimes you are asked to attach your code about your report or homework. Using listings package will avoid dealing with various special symbols and rearranging all of your code. (texdoc listings for more information)

Example

```
\usepackage{listings}
\lstset{language=[LaTeX]TeX, numbers=left, tabsize=4,
↪   keywordstyle=\color{blue}\bfseries, identifierstyle=\bf,
↪   breaklines=true, basicstyle=\tiny, rulecolor=\color{brown},
↪   numberstyle=\color[RGB]{20,20,20}}
\begin{lstlisting}
%code here
\end{lstlisting}
```

minted

Minted is another way to include code into LaTeX. Unlike listings, it is easier to use because there is no need to pre-set the syntax highlighting. However, you need to download pygments, and add –shell-escape in the configuration of XeLaTeX.

Example

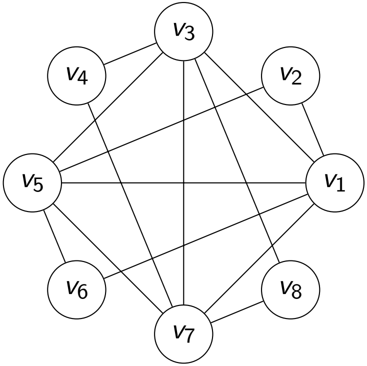
```
\usepackage{minted}
\begin{minted}{c++}
#include <iostream>
int main{
    std::cout<<"Hello world";
}
\end{minted}
```

```
#include <iostream>
int main{
    std::cout<<"Hello world";
}
```

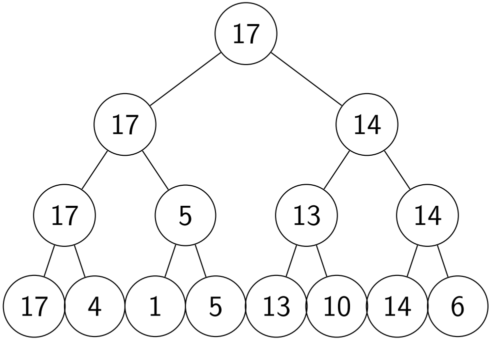
Draw graphs with TikZ and PGF

In your VE203 or some projects, you may need this package to draw graphs. There is a document of more than one thousand pages about it (texdoc tikz or texdoc pgf)

```latex
\begin{tikzpicture}[scale=2,
  ↪ bend angle=22.5]
\tikzstyle{every
  ↪ node}=[draw,shape=circle];
\foreach \i in {1,...,8}
{
\path (45*\i-45:1cm) node (v\i)
  ↪ {$v_\i$};
}
\draw
(v1) -- (v2) (v3) -- (v4) (v5)
  ↪ -- (v6) (v7) -- (v8)
(v1) -- (v3) (v3) -- (v5) (v5)
  ↪ -- (v7) (v7) -- (v1)
(v2) -- (v5) (v4) -- (v7) (v6)
  ↪ -- (v1) (v8) -- (v3)
(v1) -- (v5) (v3) -- (v7);
\end{tikzpicture}
```


v_1
v_2
v_3
v_4
v_5
v_6
v_7
v_8

```
\begin{tikzpicture}[scale=0.8]
\tikzstyle{every node}=[draw,shape=circle,minimum size=0.8cm];
\node {17}[sibling distance=4cm]
child { node {17}[sibling distance=2cm]
    child {
        node {17}[sibling distance=1cm]
        child { node {17} }
        child { node {4} }
    }
    child {
        node {5}[sibling distance=1cm]
        child { node {1} }
        child { node {5} }
    }
}
child { node {14}[sibling distance=2cm]
    child {
        node {13}[sibling distance=1cm]
        child { node {13} }
        child { node {10} }
    }
    child {
        node {14}[sibling distance=1cm]
        child { node {14} }
        child { node {6} }
    }
};
\end{tikzpicture}
```


17
17
14
17
5
13
14
17
4
1
5
13
10
14
6

The process of memorizing the code in TikZ is quite hard, so while you need to plot some graph with TikZ, it is highly recommended that you refer to the `http://www.texample.net/tikz/` for the codes of examples shown in it.