Introduction to LaTeX
Seminar: Common Security Flaws in JavaScript based Applications

Paul Muntean
paul@sec.in.tum.de

Chair for IT Security I20
Prof. Dr. Claudia Eckert
Technical University of Munich

18.04.2016
Outline

1. Latex in General
2. General Information on the Elaboration
3. Latex Presentation
Why \LaTeX? 

- Design has nothing to do with content.
- Documents can be written in the text editor.
- SVN, GIT, ... compatible!
- Design is added by compiling the contents.
- Simple insertion of mathematical formulas.
- Packages available for German texts as well.
Literature

- http://www.sec.in.tum.de/hilfestellungen-zu-seminaren/
- http://www.dante.de/tex/TeXAnfaenger.html
- http://www.ctan.org/
- http://en.wikibooks.org/wiki/LaTeX/
- https://www.google.de/
Basic Structure

Code

\documentclass{article}
\begin{document}
: 
Hello World!
: 
\end{document}

- Document class gives basic document structure.
- Content is added in the \textit{document} environment.
The Basic Class

Code

\documentclass[12pt, a5paper, landscape]{scrartcl}
\begin{document}

: Hello World!

:\
\end{document}

article → scrartcl; book → scrbook; report → scrreprt; letter → scrlttr2

There are also komma scripts for german texts as well

The basic class is \textit{scrartcl}.

Other options are: \textit{a4paper,11pt}

\textit{twoside} - Double-sided printing (margins)

\textit{landscape} - Landscape
Packages

Code
\documentclass[optionen]{scrartcl}
\usepackage[options]{package}
\begin{document}
[...]
\end{document}

- babel (with language option) culturally determined typographical rules
- inputenc (with option utf8) for correct input encoding
- graphicx for figures
- todonotes comments inside the document
- Other packages for figures, hyperlinks, ...
Environment

Code

\begin{document}
\begin{center}
centered
\end{center}
\end{document}

Different environments to structure content.
- \textit{center} for centering text.
- \textit{itemize} or \textit{enumerate} for enumerations
- \textit{tabular} for tables
- \textit{figure} for graphics
Environment - Examples (1)

\begin{document}
\begin{itemize}
\item one
\item two
\item three
\end{itemize}
\end{document}

\begin{document}
\begin{enumerate}
\item one
\item two
\item three
\end{enumerate}
\end{document}


\begin{document}
\begin{tabular}{lcr}
  1 & 2 & 3 \\
  4 & 5 & 6 \\
  7 & 8 & 9 \\
\end{tabular}
\end{document}

\begin{figure}[h]
\includegraphics[width=3cm]{foo}
\caption{Caption!}
\label{fig:example}
\end{figure}

\begin{document}
\begin{figure}[h]
\includegraphics[width=3cm]{foo}
\caption{Caption!}
\label{fig:example}
\end{figure}
\end{document}
Typesetting

Code

\begin{document}
\begin{itemize}
\item \textbf{Bold}, \textit{Italic}, \texttt{Typewriter}
\item {\Large Large}, {\Huge Huge}, {\footnotesize footnotesize}, \ldots
\item better \texttt{\emph{\textbackslash emph}}
\end{itemize}
\end{document}

- **Bold**, *Italic*, Typewriter
- Large, **Huge**, footnotesize, \ldots
- better \texttt{\emph{\textbackslash emph}}
Document Structuring

Code

\[\text{[...]}\]
\begin{document}
\tableofcontents
\chapter{} \textit{ <-- Usually used with books, srcbook and LLCS.}
\section{Foo}
\subsection{Bar}
\subsubsection{FooBar}
[...]
\end{document}

- Dividing the document into sections (Sections).
- In Book (and LNCS) parent chapter (Chapter).
- Outline with \texttt{\tableofcontents}.
- The .tex source file has to be compiled several times.
Hello World Example!
Bibtex (bibtex.bib) file creation
(if you want you can use JabRef for administration).

Citation - Integrating the references

\begin{document}

[...]

This was demonstrated 1997 in Freely \cite{freely1997}.

[...]

\bibliographystyle{ieeeetr}
\bibliography{bibtex} <-- bibtex.bib

[...]

\end{document}

- Entries can be added with \texttt{\cite{}}.
- Embed literature list in the document.
- Only references used in the bibliography.
  - \texttt{\nocite{}} do not use.
Before delivering the report check the overfull and underfull \hboxes{}. 
- Hyphenation for a word unknown.
- Long object leaves no separation possibility.
- Table is too wide.

Avoid widows and orphans
(see: https://en.wikipedia.org/wiki/Widows_and_orphans)

\clubpenalty = 10000 % No "widows"
\widowpenalty = 10000
\displaywidowpenalty = 1000 % No "orphans"
\setlength{\baselineskip}{3ex}
Generally

- Complete sentences in tables, captions and footnotes.
- Figures, Tables, ... always reference and give description in text.
- No duplicate headers.
- Pay attention to spelling errors.
- For numbers and formulas use math mode ($...$)
- Use glossary for abbreviations.
Latex can be used also for slides.

- `beamer` as design package.
- `frame` environment for single slides.
- `block` environment for emphasising.
Summary of the actual message on the last slide.
Summarize important points again.
No questions on the last slide.
This slides should remain in memory after the presentation.
- LaTeX is good for writing academic papers.
- After a brief introduction LaTeX usage becomes intuitive.
- LaTeX cares largely by itself of the design of the work.
- Additional packages allow expansion of the \LaTeX functionality.
- The report can be relatively quickly transformed into a presentation.
- A lot of documentation available.
<table>
<thead>
<tr>
<th>Termine</th>
<th>Topic</th>
<th>Presenter</th>
</tr>
</thead>
<tbody>
<tr>
<td>09.05.16</td>
<td>Too LeJIT to Quit</td>
<td>Atakan D.</td>
</tr>
<tr>
<td>09.05.16</td>
<td>RockJIT</td>
<td>Andrej K.</td>
</tr>
<tr>
<td>23.05.16</td>
<td>Expl. and Protecting Dynamic Code</td>
<td>Paul T.</td>
</tr>
<tr>
<td>23.05.16</td>
<td>JITScope</td>
<td>Karl K.</td>
</tr>
<tr>
<td>30.05.16</td>
<td>JSand</td>
<td>Roman H.</td>
</tr>
<tr>
<td>30.05.16</td>
<td>The Devil is in the Constants</td>
<td>Jonathan R.</td>
</tr>
<tr>
<td>06.06.16</td>
<td>The Spy in the Sandbox</td>
<td>Lukas H.</td>
</tr>
<tr>
<td>06.06.16</td>
<td>JaTE</td>
<td>Fabian E.</td>
</tr>
<tr>
<td>13.06.16</td>
<td>The unexpected Dangers of Dynamic JS</td>
<td>Niklas S.</td>
</tr>
<tr>
<td>13.06.16</td>
<td>Isomeron</td>
<td>Sebastian W.</td>
</tr>
<tr>
<td>20.06.16</td>
<td>Modular Control-Flow Integrity</td>
<td>Mykola O.</td>
</tr>
<tr>
<td>20.06.16</td>
<td>Just-In-Time Code Reuse</td>
<td>Hendrik E.</td>
</tr>
<tr>
<td>27.06.16</td>
<td>Readactor</td>
<td>Andreas S.</td>
</tr>
<tr>
<td>27.06.16</td>
<td>Per-Input Control-Flow Integrity</td>
<td>Togi D.</td>
</tr>
</tbody>
</table>