USENIX Example Paper

Pekka Nikander　　*Jane-Ellen Long*
Aalto University　　*USENIX Association*

Abstract

This is an example for a USENIX paper, in the form of an HTML/CSS template. Being heavily self-referential, this template illustrates the features included in this template. It is expected that the prospective authors using HTML/CSS would create a new document based on this template, remove the content, and start writing their paper.

Note that in this template, you may have a multi-paragraph abstract. However, that it is not necessarily a good practice. Try to keep your abstract in one paragraph, and remember that the optimal length for an abstract is 200-300 words.

1 Introduction

For the purposes of USENIX conference publications, the authors, not the USENIX staff, are solely responsible for the content and formatting of their paper. The purpose of this template is to help those authors that want to use HTML/CSS to write their papers. This template has been prepared by Håkon Wium Lie, and is based on a guide to using FrameMaker for USENIX papers, written by Pekka Nikander with the help of Jane-Ellen Long.

The rest of this paper is organized as follows. Section 2 gives a brief overview of related work, such as other templates and style manuals. Section 3 discusses the details of this template, and Section 4 contains our conclusions.

2 Related Work

Preparing good-looking publications is not easy. It requires understanding of style and typography. The purpose of the templates provided by the USENIX organization is to lift the burden of caring about typography from the authors. However, the authors still remain, and will always remain, responsible for the style.

2.1 Word and LaTeX templates

The USENIX website includes a template for Microsoft Word, as well as LaTeX templates. Many of the settings in the CSS style sheet of this template have been copied from the LaTeX templates.
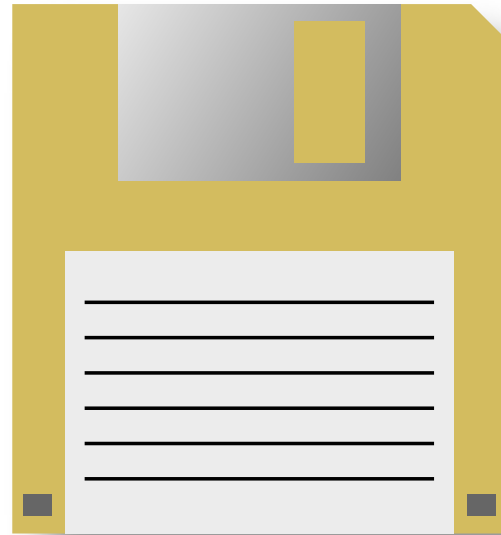

Figure 1: This figure is showed for illustrational purposes only; floppy disks are not required to use this template.

2.2 Style manuals

Besides typography, style is the second element of preparing easy-to-read publications. There are tens of good style manuals available. To mention just a couple, The Elements of Style by Strunk and White [1] is a classic, and has remained a bestseller since its introduction in 1930's. From the more contemporary ones, Writing for Computer Science by Justin Zobel [2] seems appropriate.

3 Implementation

In this section we cover the features included in this template. Our goal has been that the authors do not need to make modifications to the template; instead, they should be able to concentrate on the content and style. With this in mind, this template includes a number of features. On the other hand, we have also tried to keep this document simple and easy to maintain.

This template is written in HTML, with CSS to provide styling, and a small JavaScript to help format references.

Figure 2: This figure floats to the top of the page, spanning both columns.

3.1 HTML5

This template uses HTML5 elements to aid in representing the document structure. The `section` element is used to split the text into sections, and the `header` element holds the headlines. The `figure` element is used to include figures and their corrensponding captions live inside the `fig-caption` element. The `cite` element holds all references.

A small microformat, based on a convention of class names, is used to encode the name and affiliation of the authors.

3.2 CSS

A CSS style sheet describes how to format the HTML document into a PDF file. CSS is a declarative language which attaches property values to HTML elements and documents. Many aspects of CSS is used to achieve the presentation of USENIX papers, including:

- multi-column layout
- footnotes
- page and column floats
- multi-level counters

Some commonly used features are absent from the above list: page numbers and running headers should not be specifed by USENIX authors, these are added by those who compile the Proceedings.

3.3 JavaScript

This template uses JavaScript to process references. References are added at the point where they appear, and a script is later used to move the references to the end of the paper, leaving behind a numeric marker.

3.4 PDF

(This section has been added by Håkon Wium Lie)

In order to convert the document to PDF, a formatter is needed. Common browsers support HTML and CSS, but they do not support all the CSS functionality for page-based formatting. For example, browsers do not support footnotes or page floats. This paper has been formatted with Prince,[a] a purpose-built program for converting HTML and XML documents into PDF by way of CSS. Prince is a commercial product, but can be downloaded and used for free for non-commercial purposes.

In order for Prince to process the script included in this template, a command line option must be specified:

```
$ prince --javascript example.html
```

4 Tables

The table below lists recipients of the USENIX Lifetime Achievement Award in the 1900s. Notice how notes inside the table are moved to the end of the table.

Year	Recipient
1999	X Window System*
1998	Tim Berners-Lee
1997	Brian W. Kernighan

[a] www.princexml.com

1996	The Software Tools Project
1995	The Creation of USENET **
1994	Networking Technologies
1993	Berkeley UNIX
* Given to the Community at Large ** Given to Jim Ellis and Tom Truscott	

5 Conclusions

Each good paper concludes the most significant findings in the end.

Acknowledgments

A polite author always includes acknowledgments. Thank everyone, especially those who funded the work.

Availability

Please include a section at the end of your paper providing availability information. If the system you describe is available to others, and if more information (reports, etc.) may be obtained, indicate terms and contact information.

References

[1] STRUNK, W. JR., AND WHITE, E.B. The Elements of Style, 4th Ed, Allyn and Bacon, August, 1999, ISBN 020530902X

[2] ZOBEL, J. Writing for Computer Science, Springer-Verlag, December 1997, ISBN 9813083220