The *magic* of Prince

Prince is a computer program that converts XML and HTML into PDF. It is simple, yet very powerful, and it creates beautiful documents. The purpose of this small document is to showcase the formatting magic Prince can do. We have chosen to highlight eighteen of our favorite features. This document is written in HTML and converted to PDF by Prince. The source file is a compact 13k document, including the embedded CSS, SVG and MathML.

Table of contents

#1: Hyphenation

Prince 6 supports automatic hyphenation which can break words across several lines, adding a hyphen at the word break. Hyphenation is controlled with a set of experimental CSS properties, and hyphenation patterns for different languages can be supplied. Notice how the text in this document is hyphenated.

#2: Rounded borders

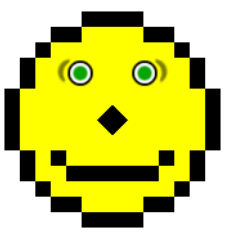
CSS3 introduces support for rounded borders. In the table below, some of the corners have been rounded. On purpose, the bottom right corner has an asymmetrical shape.

	fruit	computer
apple	yes	yes
orange	yes	no

#3: Character substitution

It's sometimes convenient to replace one character with another without changing the source document. For example, the apostrophe character is easily found on keyboards, but in print it's common to replace it with a quotation character. Notice how Prince 6 has replaced the apostrophes in this paragraph.

#4: HTTP support



Prince 6 has built-in HTTP support and can fetch pages, images, DTDs and style sheets from the web. The image to the left was automatically fetched when the PDF version was generated.

#5: Web fonts

Prince 6 can fetch fonts from the web and use them without installing them on your system. The fonts used in the PDF version of this document are web fonts. We are grateful to Ray Larabie, Dieter Steffmann, and Red Hat for making hi-quality fonts freely available.

#6: Image resolution

Sometimes images should be scaled to a certain resolution, rather than to an absolute size. In Prince, you can set the resolution of an image as a property in the style sheet. The smiley face in the previous section was scaled this way. The smiley also represents the challenging Acid2 test, which Prince 6 passes.

#7: Columns

Columns are commonly used on paper and Prince supports multi-column layouts described in CSS. This document is laid out in two columns. Also, this section uses a two-column layout with gap and a rule between. The width of the gap and the style of the rule is set in the style sheet.

#8: CMYK colors

Printers don't use RGB colors, they mostly use CMYK: cyan, magenta, yellow and black. Prince 6 can read CMYK colors and will use them, if present. The heading above this paragraph has both an RGB color (red) and a CMYK color (bluish). Therefore, the text is red in

USENIX Example Paper

Pekka Nikander
Aalto University

Jane-Ellen Long
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.

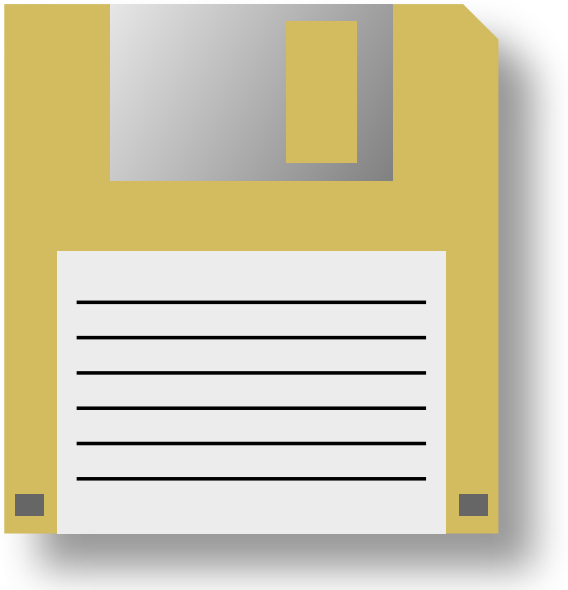


Figure 1: This figure is showed for illustrational purposes only; floppy disks are not required to use this template.

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.

PRINCE
furniture

SPRING 2013 CATALOGUE

www.princexml.com