edp Renewables

Change Control Information for Ticket: CHG808080

Change Control	**Security Update**
CHG808080	Yes
Date Opened	**Security Review Date**
08/02/2023	01/02/2023

Description

This is a description of the work done by the change control....

Test Environment

Test Approval Date

01/02/2023

Test Install Date

01/08/2023

Test Worknotes

Test environment successfully passed with no reported issues.

Test Environment Screenshot Evidence (Before)

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.

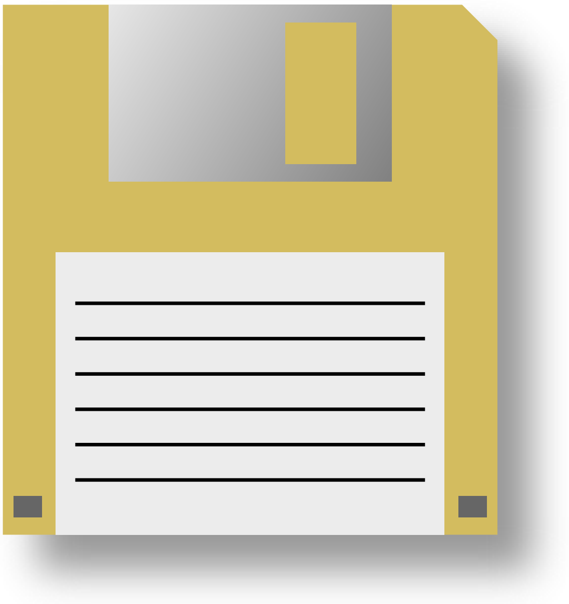

Figure 1: This figure is showed for illustrational purposes only; floppy disks are not required to use this template.

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.

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 `figcaption` 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

Test Environment Screenshot Evidence (After)

Invoice

YesLogic Pty. Ltd.
7 / 39 Bouverie St
Carlton VIC 3053
Australia

www.yeslogic.com
ABN 32 101 193 560

Customer Name
Street
Postcode City
Country

Invoice date:	Nov 26, 2016
Invoice number:	161126
Payment due:	30 days after invoice date

Description	**From**	**Until**	**Amount**
Prince Upgrades & Support	Nov 26, 2016	Nov 26, 2017	USD $950.00
Total			USD $950.00

Please transfer amount to:

Bank account name:	Yes Logic Pty Ltd
Name of Bank:	Commonwealth Bank of Australia (CBA)
Bank State Branch (BSB):	063010
Bank State Branch (BSB):	063010
Bank State Branch (BSB):	063019
Bank account number:	13201652
Bank SWIFT code:	CTBAAU2S
Bank address:	231 Swanston St, Melbourne, VIC 3000, Australia

The BSB number identifies a branch of a financial institution in Australia. When transferring money to Australia, the BSB number is used together with the bank account number and the SWIFT code. Australian banks do not use IBAN numbers.

Production Environment

Prod Approval Date

01/15/2023

Prod Install Date

01/28/2023

Prod Worknotes

Production environment successfully updated with no issues.

Production Environment Screenshot Evidence (Before)

The *magic* of Prince

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

Table of Contents

Hyphenation 1
Rounded borders 1
Character substitution 1
HTTP support.................1
Web fonts 1
Image resolution.............1
Columns 1
CMYK colors.................1
Counters 2
Crop and cross marks.....2
Cross-references 2
Math..............................2
Footnotes.......................2
Leaders..........................2
SVG 2
Page floats......................2
Headers and footers 2
PDF bookmarks 2

#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 supp and can fetch pages, images, DT and style sheets from the web. T image to the left was automatica fetched when the PDF version w generated.

#5: Web fonts

Prince 6 can fetch fonts from the web and use them wi out installing them on your system. The fonts used the PDF version of this document are web fonts. We a grateful to Ray Larabie, Dieter Steffmann, and Red H for making hi-quality fonts freely available.

#6: Image resolution

Sometimes images should be scaled to a certain reso tion, rather than to an absolute size. In Prince, you can the resolution of an image as a property in the style she The smiley face in the previous section was scaled tl way. The smiley also represents the challenging Aci 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 layo with gap and a rule b tween. The width of the g and the style of the rule set in the style sheet.

#8: CMYK colors

Printers don’t use RGB colors, they mostly use CMY cyan, magenta, yellow and black. Prince 6 can re CMYK colors and will use them, if present. The headi above this paragraph has both an RGB color (red) a a CMYK color (bluish). Therefore, the text is red www.princexml.com

browsers, but blue in the PDF version. This is for demonstration purposes; normally the colors would be close to each other.

#9: Counters

If you are reading the HTML source code of this document, look for the `h2` elements. You will notice that they contain the text of the headings, but not their number. The list item number, including the “#” and “:” are automatically generated by the style sheet. Generated content and counters are especially useful for complex documents. They are described in CSS level 2.1.

#10: Crop and cross marks

In printing, crop marks are used to indicate where the printed paper should be cut. Cross marks are used to align prints of different colors to improve color reproduction. Prince 6 adds support for crop and cross marks, and the PDF version of this document includes both.

#11: Cross-references

Prince can read hyperlinks inside a document and generate page numbers accordingly. For example, it will automatically find out which page *Headers and Footers* are discussed on (page 2). Cross-references are used to generate the *Table of contents* (page 1).

#12: Math

Prince 6 add experimental support for MathML. Here is an example:

$$x \xrightarrow{\text{maps to}} y = f_n(x) = \left(1 + \frac{1}{x^n}\right)^n$$

#13: Footnotes

Footnotes[1] are essential in printed documents and Prince knows how to generate them. Unlike what some people think, footnotes are not the place to put information you don’t want to see. More often, footnotes will actually
tract attention. 9 of 10 of readers will read the footno
before they read the text from where the footnotes are a
chored.[2]

#14: Leaders

Leaders consist of dots or dashes in a row leading t
eye across a page. For example, the *Table of contents* h
leaders in it. The leaders are not found in the document
self, but rather in the style sheet.

#15: SVG

Scalable Vector Graphics (SVG) is a la
guage for describing two-dimensional grap
ics for the web. SVG images scale better th
traditional bitmapped images and are suital
for printing. The crown is generated by two SVG e
ments.

#16: Page floats

On paged media, elements can be set float to the top
bottom of pages. The big URL at the top of this pa
comes after this paragraph in the source code, but is flo
ed to the top by the style sheet.

#17: Headers and footers

Printed documents often have page headers and foote
For example, page numbers are often printed at the b
tom of the page, and the document title is shown at the t
– except on title pages.

#18: PDF Bookmarks

Prince will automatically generate PDF bookmarks fro
heading elements in HTML. The feature is set with
property in the style sheet, and can also be used with oth
markup languages.

[1] A footnote is a note placed at the bottom of a page of a book or manuscript that comments on or cites a reference for a designa
part of the text.

[2] Often, the most interesting information is found in the footnotes.

Production Environment Screenshot Evidence (After)

A Simple PDF File

This is a small demonstration .pdf file -

just for use in the Virtual Mechanics tutorials. More text. And more text. And more text. And more text. And more text. And more text.

And more text. And more text. And more text. And more text. And more text. And more text. Boring, zzzzz. And more text. And more text. And more text. And more text. And more text. And more text. And more text. And more text. And more text.

And more text. And more text. And more text. And more text. And more text. And more text. And more text. Even more. Continued on page 2 ...

Simple PDF File 2

...continued from page 1. Yet more text. And more text. And more text. And more text. And more text. And more text. And more text. And more text. Oh, how boring typing this stuff. But not as boring as watching paint dry. And more text. And more text. And more text. And more text. Boring. More, a little more text. The end, and just as well.