Article.cls

Wannes Meert
February 21, 2012

The articlecls stylesheet and javascript module are an attempt to create a nicely looking style for print and summarize the typographical advancements made for html. It is inspired by the features found in LaTeX and triggered by the idea that most of these features are also supported in today's CSS3 and HTML5. A live version is available on http://wannesm.be/articlecls/ and a generated PDF version can be found on http://wannesm.be/articlecls/articlecls_print.pdf. The sources (including this text) can be found at https://github.com/wannesm/articlecls. This is very much work in progress and all feedback is appreciated.

Table of Contents

1. Structure

1.1. Header

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
        <title>Article.cls</title>
        <meta charset="utf-8" />
        <meta name="generator" content="articlecls" />
        <meta name="author" content="Wannes Meert" />
        <meta name="dcterms.issued" content="2012-02-06" />
        <link rel="author" href="mailto:wannes.meert@cs.kuleuven.be" />

        <link rel="stylesheet" href="css/normalize.css" type="text/css" />
        <link rel="stylesheet" href="css/articlecls.css" type="text/css" />

        <script src="js/jquery-1.7.1.min.js"></script>
        <script src="js/articlecls.js"></script>
</head>
```

You can pass options to the `article.cls` class. For example, if you want a modern version of the style you can use the following line:

```
<script data-options="modern" src="js/articlecls.js"></script>
```

All options that are available (pass multiple options to `data-options` as a comma separated list):

`altfootnotes`	See footnotes section
`bibliography=filename`	See bibliography section
`citationstyle=stylename`	See bibliography section
`hyphenator`	See hyphenation section
`interactive`	Adds interactive web features
`noglossary`	See glossary section
`nomathjax`	See mathematics section
`notitle`	See title section
`notoc`	See table of contents section
`tocsearch`	See table of contents section
`twocolumns`	See two columns section
`prince`	Make browser output html that can be interpreted by Prince (See Prince section)

Table 1: Options accepted by Article.cls

1.2. Body

Overview:

```
<body>
<article>
        <section>
                <h2>Introduction</h2>
                <p>Some introductionary text.</p>
        </section>
        <section>
                <h2>Another section</h2>
                <p>Some more text.</p>
                <section>
                        <h3>A subsection</h3>
                        <p>Some subsection text.</p>
                </section>
        </section>
        <section>
                <h2 class="notoc">References</h2>
                <p>Some references.</p>
        </section>
</article>
</body>
</html>
```

The `<section>` tags are optional. Header tags (e.g., `h1`) implicitely introduces a new section.

1.3. Mininal html5 header and body

The html5 standard allows you to omit optional tags like `html`, `head`, `body`. Also not all closing tags (e.g., `</p>`) are required. This can make your document significantly simpler. An equivalent version of the header and body together as shown above would be:

```
<!DOCTYPE html>
<title>Article.cls</title>
<meta charset="utf-8" />
<meta name="generator" content="articlecls" />
<meta name="author" content="Wannes Meert" />
<meta name="dcterms.issued" content="2012-02-06" />
<link rel="author" href="mailto:wannes.meert@cs.kuleuven.be" />

<link rel="stylesheet" href="css/normalize.css" type="text/css" />
<link rel="stylesheet" href="css/articlecls.css" type="text/css" />

<script src="js/jquery-1.7.1.min.js"></script>
<script src="js/articlecls.js"></script>
<article>
        <h2>Introduction</h2>
        <p>Some introductionary text.
        <h2>Another section</h2>
```

```
        <p>Some more text.
                <h3>A subsection</h3>
                <p>Some subsection text.
        <h2 class="notoc">References</h2>
        <p>Some references.
</article>
```

1.4. Title

The title, authors and date are automatically inserted at the beginning of the article based on the meta-information found in the header unless the `notitle` option is given in the header or a `header` block is present.

If you want to avoid Javascript you can add the title block manually within the `article` tags as follows:

```
<header>
        <h1>Title</h1>
        <div class=authors>
                <span class=name>Firstname1 Lastname1</span>
                <span class=name>Firstname2 Lastname2</span>
        </div>
        <time datetime="2012-01-17" pubdate>January 17, 2012</time>
</header>
```

1.5. Abstract

Add a section with id `abstract` to your document.

```
<section id="abstract">
        My abstract ...
<\section>
```

1.6. Headers and footers

The `articlecls` module adds a header which contains the document title on the left page and the current section on the right page. The footer contains the current page number. These features are supported by prince but not by all browsers.

1.7. Two columns

If you prefer a two-column layout you can add the `twocolumns` option to the articlecls module.

```
<script data-options="twocolumns" src="js/articlecls.js"></script>
```

1.8. Page breaks

You can force a line break by inserting the following snippet:

```
<div class="page-break"></div>
```

1.9. Appendices

Appendices can be introduced by adding the `appendix` class to an `<h3>` tag.

For example:

```
<h3 class=appendix>My Appendix Title</h3>
```

2. Referencing

2.1. Table of Contents

Add the id `toc` to the section where you want your table of contents. Headers having as class `notoc` will not be included in the table of contents and header with the class `nonumber` will be unnumbered but included into the table of contents.

```
<section id="toc">
        <h2 class="nonumber">Table of Contents</h2>
<\section>
```

If you want an input field for live searching the table of contents (like in this document) you can add the `tocsearch` option to the `article.cls` class.

2.2. Cross-referencing

To make a reference from one part of your text to another you can use the id-tag. For example a link to the parent section. Inserting automatically the section number when cross-referencing to a section can be done with the `h2ref` and `h3ref` classes. Cross-referencing to figures and tables is explained further (requires css properties that are not yet implemented in all browswer).

```
...
<h3 id="mytitle">Tagged title</h3>
...
<figure id="myfig">...</figure>
...
<p>Some text referencing the <a href="#mytitle" class="h3ref">tagged title</a> and
the <a href="#myfig" class="figref">tagged figure</a>.</p>
```

2.3. Footnotes

Inserting footnotes can be done as follows:[1]
Note that the necessary CSS properties are not yet supported in most browser and articlecls uses an extra option `altfootnotes` to cope with that.

1 Example footnote.

```
<span class="footnote">Example footnote.</span>
```

2.4. Glossary

While writing a text, you can introduce glossary items like HTML and CSS with the `abbr` tag. If you add the `noglossary` class, the abbreviated term will not appear in the glossary. The class `hidden` can be used to introduce a term into the glossary without showing it in the main text. A glossary is automatically inserted into the section with id `glossary`:

```
Introducing the term <abbr title="HyperText Markup Language">HTML</abbr> in a text.
<section id="glossary">
        <h2>Glossary</h2>
</section>
```

For an example, see the glossary section.

3. Formatting

3.1. Hyphenation

CSS3 supports hyphenation and articlecls applies it to all paragraphs. A more detailed explanation and an extension in javascript can be found on the website of Hyphenator.js which is a Javascript solution for browser that do not yet support advanced hyphenation. The hyphenator.js extension can be activated by giving the `hyphenator` option to the articlecls.js module.

Example: Once upon a midnight dreary, while I pondered, weak and weary, Over many a quaint and curious volume of forgotten lore — While I nodded, nearly napping, suddenly there came a tapping, As of some one gently rapping, rapping at my chamber door. “‘Tis some visiter,” I muttered, “tapping at my chamber door — Only this and nothing more.” (Edgar Allan Poe)

3.2. Quotes

> If it disagrees with experiment, it is wrong. In that simple statement is the key to science. It doesn’t make any difference how beautiful your guess is, how smart you are, who made the guess, or what his name is. If it disagrees with experiment, it’s wrong. That’s all there is to it.
> — Richard Feynman

```
<blockquote cite="http://en.wikiquote.org/wiki/Richard_Feynman">
        <p>If it disagrees with ... That's all there is to it.</p>
        <footer>Richard Feynman</footer>
</blockquote>
```

3.3. Aside box

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

> Some extra information on the side. Or a quote?

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

```
<aside>
        <p>Some extra information on the side. Or a quote?</p>
</aside>
<p>Lorem ...
```

3.4. Address

Wannes Meert
Departement of Computer Science, KU Leuven
Celestijnenlaan 200A
3001 Heverlee (Leuven)
België
http://wannesm.be

```
<script>insertAddress({
        'fn': 'Wannes Meert',
        'org': 'Departement of Computer Science, KU Leuven',
        'adr': {
                'street-address': 'Celestijnenlaan 200A',
                'postal-code': '3001',
                'locality': 'Heverlee (Leuven)',
                'country-name': 'België',
        },
        'url': 'http://wannesm.be',
})</script>
```

4. Lists

4.1. Inserting a numbered list

```
<ol>
        <li>Item 1
        <li>Item 2
</ol>
```

Output:

1. Item 1
2. Item 2

4.2. Inserting a bullet list

```
<ul>
        <li>Item 1
        <li>Item 2
</ul>
```

Output:

- Item 1
- Item 2

4.3. Inserting a description list

```
<dl>
        <dt>Label 1 <dd>Item 1
        <dt>Label 2 <dd>Item 2
</dl>
```

Output:

Label 1
    Item 1
Label 2
    Item 2

5. Floats

5.1. Inserting an image

HTML allows to insert image formats like .jpg, .png, .gif. In browsers that support all CSS3 properties you can autonumber references (e.g., Figure 2). See also the section about cross-referencing.

Figure 2: Example image.

```
<p>Reference <a class="figref" href="#firstfig">this figure</a>.</p>
<figure id="firstfig">
        <img alt="image" src="image.jpg" />
        <figcaption>Example image.</figcaption>
</figure>
```

Listing 6: Code to insert an image.

5.2. Inserting a pdf

Next to image files, you can also include pdfs as images (works in Safari, for Chrome and Firefox you'd need to use the `object` tag).

Figure 3: Example pdf.

```
<figure>
        <img src="img/results.pdf" title="results" width="60%" >
        <figcaption>Example pdf.</figcaption>
</figure>
```

Figure 4: Code to insert pdf.

5.3. Inserting code

Some inline `code-snippet` to show how it looks like.

```
<p>Inline <code>code-snippet</code>...</p>
<figure class=listing>
        <pre><code>
                <figure class=listing>
                ...
                </figure class=listing>
        </code></pre>
        <figcaption>Example code.</figcaption>
</figure>
```

Listing 7: Example code.

5.4. Inserting a table

Table title.

Col1	Col2	Col3
Item 1 1	Item 1 2	Item 1 3
Item 2 1	Item 2 2	Item 2 3

Table 2: Tabular caption text.

```
<figure class=table>
<table>
        <caption>Table title.</caption>
        <thead>
                <tr> <th> Col1 <th> Col2 <th> Col3
        <tbody>
                <tr> <td> Item 1 1 <td> Item 1 2 <td> Item 1 3
                <tr> <td> Item 2 1 <td> Item 2 2 <td> Item 2 3
</table>
<figcaption>Tabular caption text.</figcaption>
</figure>
```

5.5. Styling a table

For style you can use CSS.

Table title

Col1	Col2	Col3
Item 1 1	Item 1 2	Item 1 3
Item 2 1	Item 2 2	Item 2 3
Item 3 1	Item 3 2	Item 3 3
Item 4 1	Item 4 2	Item 4 3

Table 3: Stylized table

```
<style>
        table.styled {
                width: 40%;
                border: 1px rgba(10%, 10%, 10%, 0.5) solid;
        }
        table.styled caption {
                font-weight: bold;
        }
        table.styled .col1 {
                background-color: #ffffda;
        }
        table.styled thead {
                background-color: rgba(10%, 10%, 10%, 0.5);
                color: white;
        }
        table.styled tbody tr:nth-child(even) {
                background-color: rgba(85%, 85%, 85%, 0.5);
        }
</style>
<table class=styled>
...
```

5.6. Float positioning

Add `top` or `bottom` class to your figure environment to position the figure at the top or botton of a page (requires css properties not yet supported in all browsers).

```
<figure class="top">...
```

6. Fonts

6.1. Switch fonts

All fonts installed on your computer can also be used by setting the css `font-family` property. The following paragraph, for example, uses your local Gill Sans font (if that is not available the Vollkorn font will be used):

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

```
<style>
        .otherfont {
                font-family: 'Gill Sans', 'Vollkorn', sans-serif;
        }
</style>
<p class="otherfont">...
```

6.2. Customized symbols

You can create your own fontfile with customized symbols by using, for example, Inkscape or SVG directly. Such fontfiles can be loaded with css and used in your documents. As an example, the font described in `fonts/articlecls.svg` contains a customized star symbol: s or an accolade: $_{a}$. It can be called as follows:

```
<style scoped>
        @font-face {
                font-family: "articlecls";
                src: url(fonts/articlecls.svg#articlecls) format("svg");
        }
        .star { font-family: "articlecls"; }
        .star:before { content: "s"; }
</style>
... star symbol: <span class=star></span>
```

7. Mathematics

7.1. Inserting an equation

articlecls uses MathJax to display mathematical equations with LaTeX inline $\frac{a}{b}$ and as blocks:

$$\frac{a}{b} \tag{1}$$

Or a more extensive example:

$$\begin{aligned}
\nabla \times \vec{\mathbf{B}} -\, \frac1c\, \frac{\partial \vec{\mathbf{E}}}{\partial t} &= \frac{4\pi}{c}\vec{\mathbf{j}} \\
\nabla \cdot \vec{\mathbf{E}} &= 4 \pi \rho \\
\nabla \times \vec{\mathbf{E}}\, +\, \frac1c\, \frac{\partial \vec{\mathbf{B}}}{\partial t} &= \vec{\mathbf{0}} \\
\nabla \cdot \vec{\mathbf{B}} &= 0
\end{aligned} \tag{2}$$

And we can reference equations (2) and (1).

```
\[
\begin{aligned}
\label{eqmaxwell}
\nabla \times \vec{\mathbf{B}} -\, \frac1c\,
\frac{\partial \vec{\mathbf{E}}}{\partial t} & = \frac{4\pi}{c}\vec{\mathbf{j}} \\
\nabla \cdot \vec{\mathbf{E}} & = 4 \pi \rho \\
\nabla \times \vec{\mathbf{E}}\, +\, \frac1c\,
\frac{\partial \vec{\mathbf{B}}}{\partial t} & = \vec{\mathbf{0}} \\
\nabla \cdot \vec{\mathbf{B}} & = 0
\end{aligned}
\]
<p>And we can reference equations \eqref{eqmaxwell}.</p>
```

8. Bibliography

`Article.cls` uses citeproc-js, which is an implementation of the Citation Style Language (CSL), to handle citations.

8.1. Inserting citations

Citations can be added to a Javascript file that is included as an option to the `article.cls` class: `bibliography=articlecls_bib.js`. Citing within the document has the following result: see reference (Meert and Lastname 2012) and (Meert 2011). The necessary html-code is the following:

```
see reference <a href="#Meert2012" class=cite></a>.

<section id=bibliography>
        <h2>Bibliography</>
</section>
```

The bib entries themselves are entered as Javascript according to the citeproc-js format and is based on the MODS bibliography metadata format. The file `articlecls_bib.js`, for example, contains the following:

```
var bibdata = {
        "Meert2012" : {
                "id": "Meert2012",
                "title": "Some example reference title",
                "author": [
                        {"family": "Meert", "given": "Wannes"},
                        {"family": "Lastname", "given": "Firstname"},
                ],
                "container-title":"One or Another Journal",
                "volume": "18",
                "page": "463-509",
                "issued": {
                        "date-parts": [
                                [2012, 1]
                        ],
                },
                "type": "article-journal"
        },
        ...
```

The result can be seen in the bibliography section.

8.2. Citation Format

The JSON format to enter citation entries is based on MODS and is briefly explained in the citeproc-js documentation. Underneath is a summary of often used fields:

- id (string): The unique identifier
- title (string): The title of the article
- author (array): The authors
  - family (string): Family name, last name
  - surname (string): Family name, last name
  - given (string): First name
  - forename (string): First name
- editors (array): The editors, format is the same as for authors
- container-title (string): Name of journal, proceeding
- published_in (string): Name of journal, proceeding
- series (string)
- volume (string): Volume number
- issue (string): Issue number
- page (string): Pagenumber(s) of publication
- pages (string): Pagenumber(s) of publication
- chapter (string)

- publisher (string)
- institution (string)
- year (string)
- issued (dictionary): Date when article was issued
  - date-parts (array):
    - Array to identify date [year, month], or [year, month, day]
- type (string): Type of publication
  - article-journal
  - journal
- identifiers (dictionary):
  - issn (string)
  - isbn (string)
  - doi (string)
  - pmid (string)
- website (string)

8.3. Using bibliography styles

Citation styles can be set as an `article.cls` option:

```
<script data-options="bibliography=articlecls_bib.js, citationstyle=ieee" src="js/
articlecls.js"></script>
```

Extra citation and bibliography styles can be found at the CLS: citation styles website or on the CSL Visual Editor website. The last one also allows you to easily change the found style to your needs. Once you have found or created a bibliography style you can copy it into the citationstyles variable (e.g., in your bibliography javascript file) as follows:

```
citationstyles['myieee'] = "<style xmlns=\"http://purl.org/net/xbiblio/csl\"
class=\"in-text\"><info>...
```

The new style is now available with the name 'myieee' to the `citationstyle` option. The result can be seen in the bibliography section.

9. Converting to PDF

9.1. Page size

The output is printed to an A4-pdf.

9.2. HTML to PDF convertor

Browser

Use print functionality of Safari or Firefox. Not all new print-related css features are yet available.

Prince

Prince allows direct printing from html to pdf. Prince supports new css features targeted towards print but has limited support for javascript. To overcome this you can pre-process the html file and let Prince parse the resulting html-file. In the support folder you can find a helper script based upon PhantomJS that does this task for you (`support/processjs.sh`).

Usage:

```
$ prince articlecls.html -o articlecls.pdf
```

10. Conclusions

10.1. Dependencies

- Normalize.css: Start from a normalized CSS state.
- jQuery: More easily alter the DOM-tree using Javascript.
- MathJax: For displaying mathematical equations.
- Hyphenator.js: Improved hyphenation.
- Citeproc-js: Citation formatting.
- CSS3: The CSS standard that contains the paged media options.
- HTML5: HTML with more semantic tags usable for print

10.2. Comparison to LaTeX

Advantages of HTML to LaTeX

- **Future-proof**: HTML becomes more and more important as a format for traditional and ebook publishing. It serves as the basis for the .epub format and the .ibooks format. Also, it is used for online word processors like Google Docs and Zoho Writer.
- **Mobile**: Easy reformatting for mobile usage.
- **Online**: Easy distribution of articles over the web (native).
- **Debugging**: Easy css and javascript debugging with advanced tools (in-browser tools, Firebug, Web inspector, htmllint, jslint, csslint, ...).
- **Stylize**: It is more easy to make visually appealing layouts with html and css.

- **Interactive**: In html it is easy to incorporate dynamic content for ebooks and online articles.

Disadvantages of HTML to LaTeX

- **Consistency**: LaTeX has more consistent commands.
  In html you mix html-commands, css-commands, javascript-commands, and, in our case, MathJax-commands which all have a different syntax.
  Applying Handlebars.js would make the syntax more consistent but at the cost of introducing yet another (non-standard) style.
- **Cross-referencing**: Cross-referencing is not as easy and consistent as in LaTeX.
  For example, you need to add a class to notify css that it is a reference to an image or a table to get the correct count automatically inserted (although this is probably solvable with some Javascript pre-processing).
- **Redefinability**: LaTeX allows hooks on all levels. With html you are limited to the hooks defined beforehand (although powerful things are possible as is proven by the MathJax library).

Glossary

CSS
: Cascading Style Sheets

HTML
: HyperText Markup Language

References

- W3C, *CSS Generated Content for Paged Media Module*, 2011
- W3C, *HTML5 (Working draft)*, 2012
- W3C, *HTML5 (Editor's draft)*, 2012
- WhatWG, *HTML Standard*, 2012
- A List Apart, *Printing a Book with CSS: Boom!*, 2005
- A List Apart, *Going to print*, 2002
- nettuts+, *28 HTML5 Features, Tips, and Techniques You Must Know*, 2011
- w3schools, *CSS3 Browser Support Reference*, 2011
- Google, *HTML/CSS Style Guide*, 2012
- Smashing Magazin, *Smashing Magazin*, 2012

Bibliography

Meert W (2011) A first title. MyPublisher, New York

Meert W, Lastname F (2012) Some example reference title. One or Another Journal 18:463-509.

w3c html validation