Formatting Open Science: agilely creating multiple document formats for academic manuscripts with Pandoc Scholar

Albert Krewinkel (Pandoc Development Team)

Robert Winkler (CINVESTAV Unidad Irapuato, Department of Biochemistry and Biotechnology)

**Keywords:** open science, document formats, markdown, latex, publishing, typesetting

# Abstract

The timely publication of scientific results is essential for dynamic advances in science.

# Introduction

w cost. Yochai Benkler (2006) predicts important changes in the information production economy, which are based on three observations:

1. A nonmarket motivation in areas such as education, arts, science, politics and theology.
2. The actual rise of nonmarket production, made possible through networked individuals and coordinate effects.
3. The emergence of large-scale peer production, e.g. of software and encyclopedias.

Immaterial goods such as knowledge and culture are not lost when consumed or shared – they are ‘nonrival’ –, and they enable a networked information economy, which is not commercially driven (Benkler 2006).

The term *‘Open Access’* (OA) was introduced 2002 by the Budapest Open Access Initiative and was defined as:

*“Barrier-free access to online works and other resources. OA literature is digital, online, free of charge (gratis OA), and free of needless copyright and licensing restrictions (libre OA).”* (Suber 2012)

Frustrated by the difficulty to access even digitized scientific literature, three scientists founded the *Public Library of Science (PLoS)*. In 2003, *PLoS Biology* was published as the first fully Open Access journal for biology (P. O. Brown, Eisen, and Varmus 2003; M. Eisen 2003).

Thanks to the great success of OA publishing, many conventional print publishers now offer a so-called ‘Open Access option’, i.e. to make accepted articles free to read for an additional payment by the authors. The copyright in these hybrid models might remain with the publisher, whilst fully OA usually provide a liberal license, such as the Creative Commons Attribution 4.0 International (CC BY 4.0, <https://creativecommons.org/licenses/by/4.0/>).

Examples such as the *Journal of Statistical Software* (*JSS*, <https://www.jstatsoft.org/>) and *eLife* (<https://elifesciences.org/>) demonstrate the possibility of completely community-supported OA publications. **Fig. 1** compares the APCs of different OA publishing business models.

*JSS* and *eLife* are peer-reviewed and indexed by Thomson Reuters. Both journals are located in the Q1 quality quartile in all their registered subject categories of the Scimago Journal & Country Rank (<http://www.scimagojr.com/>), demonstrating that high-quality publications can be produced without charging the scientific authors or readers.

Article Processing Charge (APCs) that authors have to pay for with different Open Access (OA) publishing models. Data from (Solomon and Björk 2016) and journal web-pages.

In 2009, a study was carried out concerning the *“Economic Implications of Alternative Scholarly Publishing Models”*, which demonstrates an overall societal benefit by using OA publishing model (Houghton et al. 2009). In the same report, the real publication costs are evaluated. The relative costs of an article for the publisher are represented in **Fig. 2**.

Estimated publishing cost for a ‘hybrid’ journal (conventional with Open Access option). Data from (Houghton et al. 2009).

Conventional publishers justify their high subscription or APC prices with the added value, e.g. journalism (stated in the graphics as ‘non-article processing’). But also stakeholder profits, which could be as high as 50%, must be considered, and are withdrawn from the science budget (Van Noorden 2013).

Generally, the production costs of an article could be roughly divided into commercial and academic/ technical costs (**Fig. 2**). For nonmarket production, the commercial costs such as margins/ profits, management etc. can be drastically reduced. Hardware and services for hosting an editorial system, such as Open Journal Systems of the Public Knowledge Project (<https://pkp.sfu.ca/ojs/>) can be provided by public institutions. Employed scholars can perform editor and reviewer activities without additional cost for the journals. Nevertheless, ‘article processing’, which includes the manuscript handling during peer review and production represents the most expensive part.

Therefore, we investigated a strategy for the efficient formatting of scientific manuscripts.

## Current standard publishing formats

Generally speaking, a scientific manuscript is composed of contents and formatting. While the content, i.e. text, figures, tables, citations etc., may remain the same between different publishing forms and journal styles, the formatting can be very different. Most publishers require the formatting of submitted manuscripts in a certain format. Ignoring this **Guide for Authors**, e.g. by submitting a manuscript with a different reference style, gives a negative impression with a journal’s editorial staff. Too carelessly prepared manuscripts can even provoke a straight ‘desk-reject’ (Volmer and Stokes 2016).

Currently DOC(X), LATEX and/ or PDF file formats are the most frequently used formats for journal submission platforms. But even if the content of a submitted manuscript might be accepted during the peer review ‘as is’, the format still needs to be adjusted to the particular publication style in the production stage. For the electronic distribution and archiving of scientific works, which is gaining more and more importance, additional formats (EPUB, (X)HTML, JATS) need to be generated. **Tab. 1** lists the file formats which are currently the most relevant ones for scientific publishing.

Although the content elements of documents, such as title, author, abstract, text, figures, tables, etc., remain the same, the syntax of the file formats is rather different. **Tab. 2** demonstrates some simple examples of differences in different markup languages.

Documents with the commonly used Office Open XML (DOCX Microsoft Word files) and OpenDocument (ODT LibreOffice) file formats can be opened in a standard text editor after unzipping. However, content and formatting information is distributed into various folders and files. Practically speaking, those file formats require the use of special word processing software.

From a writer’s perspective, the use of *What You See Is What You Get (WYSIWYG)* programs such as Microsoft Word, WPS Office or LibreOffice might be convenient, because the formatting of the document is directly visible. But the complicated syntax specifications often result in problems when using different software versions and for collaborative writing. Simple conversions between file formats can be difficult or impossible. In a worst-case scenario, ‘old’ files cannot be opened any more for lack of compatible software.

In some parts of the scientific community therefore LATEX, a typesetting program in plain text format, is very popular. With LATEX, documents with highest typographic quality can be produced. However, the source files are cluttered with LATEX commands and the source text can be complicated to read. Causes of compilation errors in LATEX are sometimes difficult to find. Therefore, LATEX is not very user friendly, especially for casual writers or beginners.

**Table 1**. Current standard formats for scientific publishing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Type** | **Description** | **Use** | **Syntax** | **Reference** |
| DOCX | Office Open XML | WYSIWYG editing | XML, ZIP | (Ngo 2006) |
| ODT | OpenDocument | WYSIWYG editing | XML, ZIP | (Brauer et al. 2005) |
| PDF | portable document | print replacement | PDF | (International Organization for Standardization 2013) |
| EPUB | electronic publishing | e-books | HTML5, ZIP | (Eikebrokk, Dahl, and Kessel 2014) |
| JATS | journal article tag suite | journal publishing | XML | (National Information Standards Organization 2012) |
| LATEX | typesetting system | high-quality print | TEX | (Lamport 1994) |
| HTML | hypertext markup | websites | (X)HTML | (Raggett et al. 1999; Hickson et al. 2014) |
| MD | Markdown | lightweight markup | plain text MD | (Ovadia 2014; Leonard 2016) |

* For the publishing of a book, with a print version in PDF and an electronic version in EPUB.
* For the distribution of a seminar script, with an online version in HTML and a print version in PDF.
* For submitting a journal manuscript for peer-review in DOCX, as well as a preprint version with another journal style in PDF.
* For archiving and exchanging article data using the Journal Article Tag Suite (JATS) (National Information Standards Organization 2012), a standardized format developed by the NLM.

Some of the tasks can be performed e.g. with LATEX, but an integrated solution remains a challenge. Several programs for the conversion between documents formats exist, such as the e-book library program calibre <http://calibre-ebook.com/>. But the results of such conversions are often not satisfactory and require substantial manual corrections.

Therefore, we were looking for a solution that enables the creation of scientific manuscripts in a simple format, with the subsequent generation of multiple output formats. The need for hybrid publishing has been recognized outside of science (DPT Collective 2015; Kielhorn 2011), but the requirements specific to scientific publishing have not been addressed so far. Therefore, we investigated the possibility to generate multiple publication formats from a simple manuscript source file.

# Concepts of markdown and pandoc

Markdown was originally developed by John Gruber in collaboration with Aaron Swartz, with the goal to simplify the writing of HTML documents <http://daringfireball.net/projects/markdown/>. Instead of coding a file in HTML syntax, the content of a document is written in plain text and annotated with simple tags which define the formatting. Subsequently, the Markdown (MD) files are parsed to generate the final HTML document. With this concept, the source file remains easily readable and the author can focus on the contents rather than formatting. Despite its original focus on the web, the MD format has been proven to be well suited for academic writing (Ovadia 2014). In particular, pandoc-flavored MD (<http://pandoc.org/>) adds several extensions which facilitate the authoring of academic documents and their conversion into multiple output formats. **Tab. 2** demonstrates the simplicity of MD compared to other markup languages. **Fig. 3** illustrates the generation of various formatted documents from a manuscript in pandoc MD. Some relevant functions for scientific texts are explained below in more detail.

# Markdown editors and online editing

The usability of a text editor is important for the author, either writing alone or with several co-authors. In this section we present software and strategies for different scenarios. **Fig. 4** summarizes various options for local or networked editing of MD files.

Markdown files can be edited on local devices or on cloud drives. A local or remote git repository enables advanced advanced version control.

## Markdown editors

Due to MD’s simple syntax, basically any text editor is suitable for editing markdown files. The formatting tags are written in plain text and are easy to remember. Therefore, the author is not distracted by looking around for layout options with the mouse. For several popular text editors, such as vim (<http://www.vim.org/>), GNU Emacs (<https://www.gnu.org/software/emacs/>), atom (<https://atom.io/>) or geany (<http://www.geany.org/>), plugins provide additional functionality for markdown editing, e.g. syntax highlighting, command helpers, live preview or structure browsing.

Various dedicated markdown editors have been published as well. Many of those are cross-platform compatible, such as Abricotine (<http://abricotine.brrd.fr/>), ghostwriter (<https://github.com/wereturtle/ghostwriter>) and CuteMarkEd (<https://cloose.github.io/CuteMarkEd/>).

The lightweight format is also ideal for writing on mobile devices. Numerous applications are available on the App stores for Android and iOS systems. The programs Swype and Dragon (<http://www.nuance.com/>) facilitate the input of text on such devices by guessing words from gestures and speech recognition (dictation).

**Fig. 5.** shows the editing of a markdown file, using the cross-platform editor Atom with several markdown plugins.

Document directory tree, editing window and HTML preview using the Atom editor.

## Online editing and collaborative writing

Storing manuscripts on network drives (*The Cloud*) has become popular for several reasons:

* Protection against data loss.
* Synchronization of documents between several devices.
* Collaborative editing options.

Markdown files on a Google Drive ([https://drive.google.com](https://drive.google.com/)) for instance can be edited online with StackEdit ([https://stackedit.io](https://stackedit.io/)). **Fig. 6** demonstrates the online editing of a markdown file on an ownCloud (<https://owncloud.com/>) installation. OwnCloud is an Open Source software platform, which allows the set-up of a file server on personal webspace. The functionality of an ownCloud installation can be enhanced by installing plugins.

Direct online editing of this manuscript with live preview using the ownCloud Markdown Editor plugin by Robin Appelman.

Even mathematical formulas are rendered correctly in the HTML live preview window of the ownCloud markdown plugin (**Fig. 6** ).

The collaboration and authoring platform Authorea (<https://www.authorea.com/>) also supports markdown as one of multiple possible input formats. This can be beneficial for collaborations in which one or more authors are not familiar with markdown syntax.

## Document versioning and change control

Programmers, especially when working in distributed teams, rely on version control systems to manage changes of code. Currently, Git (<https://git-scm.com/>), which is also used e.g. for the development of the Linux kernel, is one of the most employed software solutions for versioning. Git allows the parallel work of collaborators and has an efficient merging and conflict resolution system. A Git repository may be used by a single local author to keep track of changes, or by a team with a remote repository, e.g. on github (<https://github.com/>) or bitbucket (<https://bitbucket.org/>). Because of the plain text format of markdown, Gised for version control and distributed writing. For the writing of the present article, the co-authors (Germany and Mexico) used a remote Git repository on bitbucket. The plain text syntax of markdown facilitates the visualization of differences of document versions, <as shown in **Fig. 7**.

Version control and collaborative editing using a git repository on bitbucket.

# Example: Manuscript with output of DOCX/ ODT format and LATEX/ PDF for submission to different journals.

Scientific manuscripts have to be submitted in a format defined by the journal or publisher. At the moment, DOCX is the most common file format for manuscript submission. Some publishers also accept or require LATEX or ODT formats. Additional to the general style of the manuscript - organization of sections, fonts, etc. – the citation style of the journal must also be followed. Often, the same manuscript has to be prepared for different journals, e.g. if the manuscript was rejected by a journal and has to be formatted for another one, or if a preprint of the paper is submitted to an archive that requires a distinct document format than the targeted peer-reviewed journal. In this example, we want to create a manuscript for a *PLoS* journal in DOCX and ODT format for WYSIWYG word processors. Further, a version in LATEX/ PDF should be produced for PeerJ submission and archiving at the PeerJ preprint server.

The examples for DOCX/ ODT are kept relatively simple, to show the proof-of-principle and to provide a plain document for the development of own templates. Nevertheless, the generated documents should be suitable for submission after little manual editing. For specific journals it may be necessary to create more sophisticated templates or to copy/ paste the generic DOCX/ ODT output into the publisher’s template.

## Development of a DOCX/ ODT template

A first DOCX document with bibliography in *PLoS* format is created with pandoc DOCX output:

pandoc -S -s --csl=plos.csl --filter pandoc-citeproc  
 -o pandoc-manuscript.docx agile-editing-pandoc.md

The parameters -S -s generate a typographically correct (dashes, non-breaking spaces etc.) stand-alone document. A bibliography with the *PLoS* style is created by the citeproc filter setting --csl=plos.csl --filter pandoc-citeproc.

The document settings and styles of the resulting file pandoc-manuscript.docx can be optimized and be used again as document template (--reference-docx=pandoc-manuscript.docx).

pandoc -S -s --reference-docx=pandoc-manuscript.docx --csl=plos.csl  
 --filter pandoc-citeproc -o outfile.docx agile-editing-pandoc.md

It is also possible to directly re-use a previous output file as template (i.e. template and output file have the same file name):

pandoc -S -s --columns=10 --reference-docx=pandoc-manuscript.docx  
 --csl=plos.csl --filter=pandoc-citeproc  
 -o pandoc-manuscript.docx agile-editing-pandoc.md

In this way, the template can be incrementally adjusted to the desired document formatting. The final document may be employed later as pandoc template for other manuscripts with the same specifications. In this case, running pandoc the first time with the template, the contents of the new manuscript would be filled into the provided DOCX template. A page with DOCX manuscript formatting of this article is shown in **Fig. 8**.

Opening a pandoc-generated DOCX in Microsoft Office 365.

The same procedure can be applied with an ODT formatted document.

## Development of a TEX/PDF template

The default pandoc LATEX template can be written into a separate file by:

pandoc -D latex > template-peerj.latex

This template can be adjusted, e.g. by defining Unicode encoding (see above), by including particular packages or setting document options (line numbering, font size). The template can then be used with the pandoc parameter --template=pandoc-peerj.latex.

The templates used for this document are included as Supplemental Material (see section *Software and code availability* below).

## Styles for HTML and EPUB

The style for HTML and EPUB formats can be defined in .css stylesheets. The Supplemental Material contains a simple example .css file for modifying the HTML output, which can be used with the pandoc parameter -c pandoc.css.

# Automating document production

The commands necessary to produce the document in a specific formats or styles can be defined in a simple Makefile. An example Makefile is included in the source code of this preprint. The desired output file format can be chosen when calling make. E.g. make outfile.pdf produces this preprint in PDF format. Calling make without any option creates all listed document types. A Makefile producing DOCX, ODT, JATS, PDF, LATEX, HTML and EPUB files of this document is provided as Supplemental Material.

## Cross-platform compatibility

The make process was tested on Windows 10 and Linux 64 bit. All documents – DOCX, ODT, JATS, LATEX, PDF, EPUB and HTML – were generated successfully, which demonstrates the cross-platform compatibility of the workflow.

# Perspective

Following the trend to peer production, the formatting of scientific content must become more efficient. Markdown/ pandoc has the potential to play a key role in the transition from proprietary to community-driven academic production. Important research tools, such as the statistical computing and graphics language R (R Core Team 2014) and the Jupyter notebook project (Kluyver et al. 2016) have already adopted the MD syntax (e.g. <http://rmarkdown.rstudio.com/>). The software for writing manuscripts in MD is mature enough to be used by academic writers. Therefore, publishers also should consider implementing the MD format into their editorial platforms.

# Conclusions

Authoring scientific manuscripts in markdown (MD) format is straight-forward, and manual formatting is reduced to a minimum. The simple syntax of MD facilitates document editing and collaborative writing. The rapid conversion of MD to multiple formats such as DOCX, LATEX, PDF, EPUB and HTML can be done easily using pandoc, and templates enable the automated generation of documents according to specific journal styles.

The additional features we implemented facilitate the correct indexing of meta information of journal articles according to the ‘semantic web’ philosophy.

Altogether, the MD format supports the agile writing and fast production of scientific literature. The associated time and cost reduction especially favours community-driven publication strategies.

# Acknowledgments

We cordially thank Dr. Gerd Neugebauer for his help in creating a subset of a bibtex data base using BibTool, as well as Dr. Ricardo A. Chávez Montes, Prof. Magnus Palmblad and Martin Fenner for comments on the manuscript. Warm thanks also go to Anubhav Kumar and Jennifer König for proofreading.

# Software and code availability

The relevant software for creating this manuscript used is cited according to (Smith, Katz, and Niemeyer 2016) and listed in **Tab. 3**. Since unique identifiers are missing for most software projects, we only refer to the project homepages or software repositories:

**Table 3.** Relevant software used for this article.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Software** | **Use** | **Authors** | **Version** | **Release** | **Homepage/ repository** |
| pandoc | universal markup converter | John MacFarlane | 1.16.0.2 | 16/01/13 | [http://www.pandoc.org](http://www.pandoc.org/) |
| pandoc-citeproc | library for CSL citations with pandoc | John MacFarlane, Andrea Rossato | 0.9.1 | 16/03/19 | <https://github.com/jgm/pandoc-citeproc> |
| pandoc-jats | creation of JATS files with pandoc | Martin Fenner | 0.9 | 15/04/26 | <https://github.com/mfenner/pandoc-jats> |
| ownCloud | personal cloud software | ownCloud GmbH, Community | 9.1.1 | 16/09/20 | <https://owncloud.org/> |
| Markdown Editor | plugin for ownCloud | Robin Appelman | 0.1 | 16/03/08 | <https://github.com/icewind1991/files_markdown> |
| BibTool | Bibtex database tool | Gerd Neugebauer | 2.63 | 16/01/16 | <https://github.com/ge-ne/bibtool> |

The software created as part of this article, *pandoc-scholar*, is suitable for general use and has been published at <https://github.com/pandoc-scholar/pandoc-scholar>, DOI: [10.5281/zenodo.376761](https://doi.org/10.5281/zenodo.376761). The source code of this manuscript, as well as the templates and pandoc Makefile, have been deposited to <https://github.com/robert-winkler/scientific-articles-markdown/>.

Drawings for document types, devices and applications have been adopted from Calibre <http://calibre-ebook.com/>, openclipart <https://openclipart.org/> and the GNOME Theme Faenza <https://code.google.com/archive/p/faenza-icon-theme/>.

# Bibliography

Benkler, Yochai. 2006. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. New Haven, CT, USA: Yale University Press.

Berners-Lee, Tim, and James Hendler. 2001. “Publishing on the Semantic Web.” *Nature* 410 (6832): 1023–4. doi:[10.1038/35074206](https://doi.org/10.1038/35074206).

Bourne, Philip. 2005. “Will a Biological Database Be Different from a Biological Journal?” *PLOS Computational Biology* 1 (3): e34. doi:[10.1371/journal.pcbi.0010034](https://doi.org/10.1371/journal.pcbi.0010034).

Brauer, Michael, Patrick Durusau, Gary Edwards, David Faure, Tom Magliery, and Daniel Vogelheim. 2005. “Open Document Format for Office Applications (OpenDocument) V1.0.” OASIS.

Brown, Cecelia. 2001. “The E-Volution of Preprints in the Scholarly Communication of Physicists and Astronomers.” *J. Am. Soc. Inf. Sci.* 52 (3): 187–200. doi:[10.1002/1097-4571(2000)9999:9999<::AID-ASI1586>3.0.CO;2-D](https://doi.org/10.1002/1097-4571(2000)9999:9999<::AID-ASI1586>3.0.CO;2-D).

———. 2003. “The Role of Electronic Preprints in Chemical Communication: Analysis of Citation, Usage, and Acceptance in the Journal Literature.” *J. Am. Soc. Inf. Sci.* 54 (5): 362–71. doi:[10.1002/asi.10223](https://doi.org/10.1002/asi.10223).

Brown, Patrick O, Michael B Eisen, and Harold E Varmus. 2003. “Why PLoS Became a Publisher.” *PLoS Biol* 1 (1). doi:[10.1371/journal.pbio.0000036](https://doi.org/10.1371/journal.pbio.0000036).

Butler, Declan. 2001. “Los Alamos Loses Physics Archive as Preprint Pioneer Heads East.” *Nature* 412 (6842): 3–4. doi:[10.1038/35083708](https://doi.org/10.1038/35083708).

Callaway, Ewen. 2013. “Preprints Come to Life.” *Nature News* 503 (7475): 180. doi:[10.1038/503180a](https://doi.org/10.1038/503180a).

Corbí, Alberto, and Daniel Burgos. 2015. “Semi-Automated Correction Tools for Mathematics-Based Exercises in MOOC Environments.” *International Journal of Interactive Multimedia and Artificial Intelligence* 3 (3): 89–95. doi:[10.9781/ijimai.2015.3312](https://doi.org/10.9781/ijimai.2015.3312).

Dominici, Massimiliano. 2014. “An Overview of Pandoc.” *TUGboat* 35 (1): 44–50.

DPT Collective. 2015. “From Print to Ebooks: A Hybrid Publishing Toolkit for the Arts.” In, edited by Joe Monk, Miriam Rasch, Florian Cramer, and Amy Wu. Institute of Network Cultures.

Eikebrokk, Trude, Tor Arne Dahl, and Siri Kessel. 2014. “EPUB as Publication Format in Open Access Journals: Tools and Workflow.” *Code4Lib*, no. 24 (April).

Eisen, Michael. 2003. “Publish and Be Praised.” *The Guardian*, October.

Fecher, Benedikt, and Sascha Friesike. 2014. “Open Science: One Term, Five Schools of Thought.” In *Opening Science*, edited by Sönke Bartling and Sascha Friesike, 17–47. Springer International Publishing.

Ginsparg, Paul. 1994. “First Steps Towards Electronic Research Communication.” *Computers in Physics* 8 (4): 390–96. doi:[10.1063/1.4823313](https://doi.org/10.1063/1.4823313).

Haak, Laurel L., Martin Fenner, Laura Paglione, Ed Pentz, and Howard Ratner. 2012. “ORCID: A System to Uniquely Identify Researchers.” *Learned Publishing* 25 (4): 259–64. doi:[10.1087/20120404](https://doi.org/10.1087/20120404).

Hickson, Ian, Robin Berjon, Steve Faulkner, Travis Leithead, Erika Doyle Navara, Edward O’Connor, Silvia Pfeiffer, et al. 2014. “HTML5.” W3C Recommendation. W3C.

Houghton, John, Bruce Rasmussen, Peter Sheehan, Charles Oppenheim, Anne Morris, Claire Creaser, Helen Greenwood, Mark Summers, and Adrian Gourlay. 2009. “Economic Implications of Alternative Scholarly Publishing Models: Exploring the Costs and Benefits.” [http://www.webarchive.org.uk/wayback/archive/20140614041628/http://www.jisc.ac.uk/publications/reports/2009/economicpublishingmodelsfinalreport.aspx#downloads](http://www.webarchive.org.uk/wayback/archive/20140614041628/http://www.jisc.ac.uk/publications/reports/2009/economicpublishingmodelsfinalreport.aspx" \l "downloads).

International Organization for Standardization. 2013. “ISO 32000-1:2008 - Document Management – Portable Document Format – Part 1: PDF 1.7.” *ISO*. <http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=51502>.

———. 2014. “ISO/IEC 10646:2014 - Information Technology – Universal Coded Character Set (UCS).” *ISO*. <http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=63182>.

Kielhorn, Axel. 2011. “Multi-Target Publishing-Generating EPub, PDF, and More, from Markdown Using Pandoc.” *TUGboat-TeX Users Group* 32 (3): 272.

Kluyver, Thomas, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, et al. 2016. “Jupyter Notebooks—a Publishing Format for Reproducible Computational Workflows.” In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 87–90. doi:[10.3233/978-1-61499-649-1-87](https://doi.org/10.3233/978-1-61499-649-1-87).

Lamport, Leslie. 1994. *LaTeX: A Document Preparation System*. 2 edition. Reading, Mass: Addison-Wesley Professional.

Lanthaler, Markus, and Christian Gütl. 2012. “On Using JSON-LD to Create Evolvable RESTful Services.” In *Proceedings of the Third International Workshop on RESTful Design*, 25–32. ACM.

Leonard, Sean. 2016. “Guidance on Markdown: Design Philosophies, Stability Strategies, and Select Registrations.” RFC. RFC Editor; Internet Request for Comments.

National Information Standards Organization. 2012. “JATS: Journal Article Tag Suite.” ANSI/NISO Z39.96.

Ngo, Tom. 2006. “OFFICE OPEN XML OVERVIEW ECMA TC45.” Ecma International.

Ovadia, Steven. 2014. “Markdown for Librarians and Academics.” *Behavioral & Social Sciences Librarian* 33 (2): 120–24. doi:[10.1080/01639269.2014.904696](https://doi.org/10.1080/01639269.2014.904696).

R Core Team. 2014. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. [https://www.R-project.org](https://www.R-project.org/).

Raggett, Dave, Arnaud Le Hors, Ian Jacobs, Arnaud Le Hors, Dave Raggett, and Ian Jacobs. 1999. “HTML 4.01 Specification.” W3C Recommendation. W3C.

Shotton, David. 2010. “CiTO, the Citation Typing Ontology.” *Journal of Biomedical Semantics* 1 (1): S6. doi:[10.1186/2041-1480-1-S1-S6](https://doi.org/10.1186/2041-1480-1-S1-S6).

Simonsen, K. 1992. “Character Mnemonics & Character Sets.” RFC. Rationel Almen Planlaegning; Internet Request for Comments.

Smith, Arfon M., Daniel S. Katz, and Kyle E. Niemeyer. 2016. “Software Citation Principles.” Edited by Silvio Peroni. *PeerJ Computer Science* 2 (September): e86. doi:[10.7717/peerj-cs.86](https://doi.org/10.7717/peerj-cs.86).

Solomon, David, and Bo-Christer Björk. 2016. “Article Processing Charges for Open Access Publicationthe Situation for Research Intensive Universities in the USA and Canada.” *PeerJ* 4 (July): e2264. doi:[10.7717/peerj.2264](https://doi.org/10.7717/peerj.2264).

Suber, Peter. 2012. *Open Access*. Cambridge, Mass: The MIT Press.

Towbin, H., T. Staehelin, and J. Gordon. 1979. “Electrophoretic Transfer of Proteins from Polyacrylamide Gels to Nitrocellulose Sheets: Procedure and Some Applications.” *Proceedings of the National Academy of Sciences* 76 (9): 4350–4. <http://www.pnas.org/content/76/9/4350>.

Van Noorden, Richard. 2012. “Journal Offers Flat Fee for ‘All You Can Publish’.” *Nature News* 486 (7402): 166. doi:[10.1038/486166a](https://doi.org/10.1038/486166a).

———. 2013. “Open Access: The True Cost of Science Publishing.” *Nature* 495 (7442): 426–29. doi:[10.1038/495426a](https://doi.org/10.1038/495426a).

———. 2014. “The ArXiv Preprint Server Hits 1 Million Articles.” *Nature News*. doi:[10.1038/nature.2014.16643](https://doi.org/10.1038/nature.2014.16643).

Volmer, Dietrich A., and Caroline S. Stokes. 2016. “How to Prepare a Manuscript Fit-for-Purpose for Submission and Avoid Getting a ‘Desk-Reject’.” *Rapid Commun. Mass Spectrom.*, January, n/a–n/a. doi:[10.1002/rcm.7746](https://doi.org/10.1002/rcm.7746).

Willinsky, John. 2005. “The Unacknowledged Convergence of Open Source, Open Access, and Open Science.” *First Monday* 10 (8). doi:[10.5210/fm.v10i8.1265](https://doi.org/10.5210/fm.v10i8.1265).

Woelfle, Michael, Piero Olliaro, and Matthew H. Todd. 2011. “Open Science Is a Research Accelerator.” *Nat Chem* 3 (10): 745–48. doi:[10.1038/nchem.1149](https://doi.org/10.1038/nchem.1149).

Yergeau, F. 2003. “UTF-8, a Transformation Format of ISO 10646.” RFC. Alis Technologies.

Youngen, Gregory K. 1998. “Citation Patterns to Traditional and Electronic Preprints in the Published Literature.” *Coll. Res. Libr.* 59 (5): 448–56. doi:[10.5860/crl.59.5.448](https://doi.org/10.5860/crl.59.5.448).