

# Inscriptis - A Python-based HTML to text conversion library optimized for knowledge extraction from the Web

Albert Weichselbraun<sup>1</sup>

<sup>1</sup> Swiss Institute for Information Science, University of Applied Sciences of the Grisons, Pulvermühlestrasse 57, Chur, Switzerland

DOI: [10.21105/joss.03557](https://doi.org/10.21105/joss.03557)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

---

**Editor:** [Sebastian Benthall](#) ↗

## Reviewers:

- [@reality](#)
- [@rlskoester](#)

**Submitted:** 12 July 2021

**Published:** 16 October 2021

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

Inscriptis provides a library, command line client and Web service for converting HTML to plain text.

Its development has been triggered by the need to obtain accurate text representations for knowledge extraction tasks that preserve the spatial alignment of text without drawing upon heavyweight, browser-based solutions such as Selenium ([Huggins et al., 2021](#)). In contrast to existing software packages such as HTML2text ([Swartz, 2021](#)), jusText ([Belica, 2021](#)) and Lynx ([Dickey, 2021](#)), Inscriptis

1. provides a layout-aware conversion of HTML that more closely resembles the rendering obtained from standard Web browsers and, therefore, better preserves the spatial arrangement of text elements. Inscriptis excels in terms of conversion quality, since it correctly converts complex HTML constructs such as nested tables and also interprets a subset of HTML (e.g., `align`, `valign`) and CSS (e.g., `display`, `white-space`, `margin-top`, `vertical-align`, etc.) attributes that determine the text alignment.
2. supports annotation rules, i.e., user-provided mappings that allow for annotating the extracted text based on structural and semantic information encoded in HTML tags and attributes used for controlling structure and layout in the original HTML document.

These unique features ensure that downstream knowledge extraction components can operate on accurate text representations, and may even use information on the semantics and structure of the original HTML document, if annotation support has been enabled.

## Statement of need

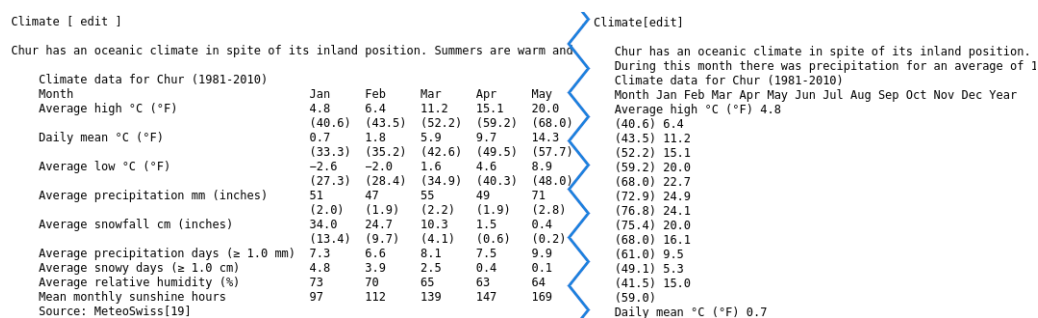
Research in a growing number of scientific disciplines relies upon Web content. [Li et al. \(2014\)](#), for instance, studied the impact of company-specific News coverage on stock prices, in medicine and pharmacovigilance social media listening plays an important role in gathering insights into patient needs and the monitoring of adverse drug effects ([Convertino et al., 2018](#)), and communication sciences analyze media coverage to obtain information on the perception and framing of issues as well as on the rise and fall of topics within News and social media ([Scharl et al., 2017](#); [Weichselbraun et al., 2021](#)).

Computer science focuses on analyzing content by applying knowledge extraction techniques such as entity recognition ([Fu et al., 2021](#)) to automatically identify entities (e.g., persons, organizations, locations, products, etc.) within text documents, entity linking ([Ding et al., 2021](#)) to link these entities to knowledge bases such as Wikidata and DBpedia, and sentiment

analysis to automatically assess sentiment polarity (i.e., positive versus negative coverage) and emotions expressed towards these entities (Wang et al., 2020).

Most knowledge extraction methods operate on text and, therefore, require an accurate conversion of HTML content which also preserves the spatial alignment between text elements. This is particularly true for methods drawing upon algorithms which directly or indirectly leverage information on the proximity between terms, such as word embeddings (Mikolov et al., 2013; Pennington et al., 2014), language models (Reis et al., 2021), sentiment analysis which often also considers the distance between target and sentiment terms, and automatic keyword and phrase extraction techniques.

Despite this need from within the research community, many standard HTML to text conversion techniques are not layout aware, yielding text representations that fail to preserve the text's spatial properties, as illustrated below:



**Figure 1:** Text representation of a table from Wikipedia computed by Inscriptis (left) and Lynx (right). Lynx fails to correctly interpret the table and, therefore, does not properly align the temperature values.

Consequently, even popular resources extensively used in literature suffer from such shortcomings. The text representations provided with the Common Crawl corpus<sup>1</sup>, for instance, have been generated with a custom utility (Kreymer et al., 2021) which at the time of writing did not consider any layout information. Datasets such as CCAIined (El-Kishky et al., 2020), multilingual C4 which has been used for training the mT5 language model (Xue et al., 2021), and OSCAR (Suárez et al., 2019) are based on subsets of the Common Crawl corpus (Caswell et al., 2021).

Even worse, some tutorials suggest the use of software libraries such as Beautiful Soup (Richardson, 2021), lxml (Behnel et al., 2021) and Cheerio (Mueller et al., 2021) for converting HTML. Since these libraries have been designed with a different use case in mind, they are only well-suited for scraping textual content. Once they encounter HTML constructs such as lists and tables, these libraries are likely to return artifacts (e.g., concatenated words), since they do not interpret HTML semantics. The creators of the Cheerio library even warn their users, by explicitly stating that it is not well-suited for emulating Web browsers.

Specialized conversion tools such as HTML2Text perform considerably better but often fail for more complex Web pages. Researchers sometimes even draw upon text-based Web browsers such as Lynx to obtain more accurate representations of HTML pages. These tools are complemented by content extraction software such as jusText (Belica, 2021), dragnet (Peters & Lecocq, 2013), TextSweeper (Lang et al., 2012) and boilerpy3 (Riebold, 2021) which do not consider the page layout but rather aim at extracting the relevant content only, and approaches that are optimized for certain kinds of Web pages like Harvest (Weichselbraun et al., 2020) for Web forums.

Inscriptis, in contrast, not only correctly renders more complex websites but also offers the option to preserve parts of the original HTML document's semantics (e.g., information on

<sup>1</sup><https://commoncrawl.org/>

headings, emphasized text, tables, etc.) by complementing the extracted text with annotations obtained from the document. Figure 2 provides an example of annotations extracted from a Wikipedia page. These annotations can be useful for

- providing downstream knowledge extraction components with additional information that may be leveraged to improve their respective performance. Text summarization techniques, for instance, can put a stronger emphasis on paragraphs that contain bold and italic text, and sentiment analysis may consider this information in addition to textual clues such as uppercase text.
- assisting manual document annotation processes (e.g., for qualitative analysis or gold standard creation). Inscriptis supports multiple export formats such as XML, annotated HTML and the JSONL format that is used by the open source annotation tool doccano<sup>2</sup> (Nakayama et al., 2018). Support for further annotation formats can be easily added by implementing custom annotation post-processors.
- enabling the use of Inscriptis for tasks such as content extraction (i.e., extract task-specific relevant content from a Web page) which rely on information on the HTML document's structure.

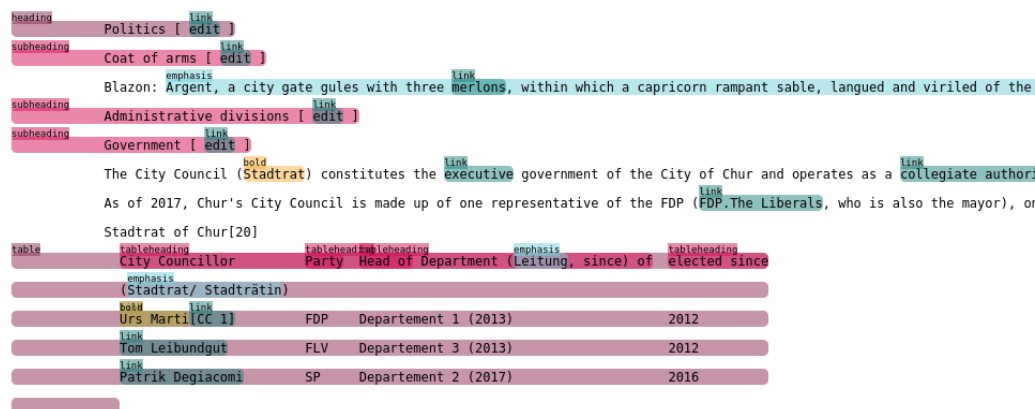


Figure 2: Annotations extracted from the Wikipedia entry for Chur that have been exported to HTML using the `--postprocessor html` command line option.

In conclusion, Inscriptis provides knowledge extraction components with high quality text representations of HTML documents. Since its first public release in March 2016, Inscriptis has been downloaded over 135,000 times from the Python Package Index (PyPI)<sup>3</sup>, has proven its capabilities in national and European research projects, and has been integrated into commercial products such as the webLyzard Web Intelligence and Visual Analytics Platform.

## Mentions

The following research projects use Inscriptis within their knowledge extraction pipelines:

- **CareerCoach**: “Automatic Knowledge Extraction and Recommender Systems for Personalized Re- and Upskilling suggestions” funded by Innosuisse.
- **Job Cockpit**: “Web analytics, data enrichment and predictive analysis for improved recruitment and career management processes” funded by Innosuisse

<sup>2</sup>Please note that doccano currently does not support overlapping annotations and, therefore, cannot import files containing overlapping annotations.

<sup>3</sup>Source: <https://pepy.tech/project/inscriptis>

- [EPOCH project](#) funded by the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility and Technology (BMK) via the ICT of the Future Program.
- [MedMon](#): “Monitoring of Internet Resources for Pharmaceutical Research and Development” funded by Innosuisse.
- [ReTV project](#) funded by the European Union’s Horizon 2020 Research and Innovation Programme.

## Acknowledgements

Work on Inscriptis has been conducted within the MedMon, Job Cockpit and CareerCoach projects funded by Innosuisse.

## References

- Behnel, S., Faassen, M., Bicking, I., Joukl, H., Sapin, S., Parent, M.-A., Grisel, O., Buchcik, K., Wagner, F., Kroymann, E., Everitt, P., Ng, V., Kern, R., Pakulat, A., Sankel, D., Kasperski, M., Silva, S. da, & Oberndörfer, P. (2021). *Lxml - processing XML and HTML with python*. lxml Project. <https://lxml.de/>
- Belica, M. (2021). jusText - heuristic based boilerplate removal tool. In *GitHub repository*. GitHub. <https://github.com/miso-belica/jusText>
- Caswell, I., Kreutzer, J., Wang, L., Wahab, A., Esch, D. van, Ulzii-Orshikh, N., Tapo, A., Subramani, N., Sokolov, A., Sikasote, C., Setyawan, M., Sarin, S., Samb, S., Sagot, B., Rivera, C., Rios, A., Papadimitriou, I., Osei, S., Suárez, P. J. O., ... Adeyemi, M. (2021). *Quality at a Glance: An Audit of Web-Crawled Multilingual Datasets*. <http://arxiv.org/abs/2103.12028>
- Convertino, I., Ferraro, S., Blandizzi, C., & Tuccori, M. (2018). The usefulness of listening social media for pharmacovigilance purposes: A systematic review. *Expert Opinion on Drug Safety*, 17(11), 1081–1093. <https://doi.org/10.1080/14740338.2018.1531847>
- Dickey, T. E. (2021). *Lynx - the text web-browser*. Lynx Home Page. <https://lynx.invisible-island.net>
- Ding, W., Chaudhri, V. K., Chittar, N., & Konakanchi, K. (2021). JEL: Applying End-to-End Neural Entity Linking in JPMorgan Chase. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17), 15301–15308. <https://ojs.aaai.org/index.php/AAAI/article/view/17796>
- El-Kishky, A., Chaudhary, V., Guzmán, F., & Koehn, P. (2020). CCAIghed: A Massive Collection of Cross-Lingual Web-Document Pairs. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5960–5969. <https://doi.org/10.18653/v1/2020.emnlp-main.480>
- Fu, J., Huang, X., & Liu, P. (2021). SpanNER: Named Entity Re-/Recognition as Span Prediction. *arXiv:2106.00641 [cs]*. <https://doi.org/10.18653/v1/2021.acl-long.558>
- Huggins, J., Gross, P., & Wang, J. T. (2021). *jusText - heuristic based boilerplate removal tool*. The Selenium Project. <https://www.selenium.dev>
- Kreymer, I., Nagel, S., Jackson, A., & Levitt, N. (2021). IIPC web archive commons - utility code for OpenWayBack and other projects. In *GitHub repository*. GitHub. <https://github.com/commoncrawl/ia-web-commons>

- Lang, H.-P., Wohlgenannt, G., & Weichselbraun, A. (2012). TextSweeper - A System for Content Extraction and Overview Page Detection. *International Conference on Information Resources Management (Conf-IRM)*. <https://aisel.aisnet.org/confirm2012/17/>
- Li, Q., Wang, T., Li, P., Liu, L., Gong, Q., & Chen, Y. (2014). The effect of news and public mood on stock movements. *Information Sciences*, 278, 826–840. <https://doi.org/10.1016/j.ins.2014.03.096>
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a Meeting Held December 5-8, 2013, Lake Tahoe, Nevada, United States*, 3111–3119. <http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality>
- Mueller, M., Böhm, F., Mike, J., & Chambers, D. (2021). Cheerio - fast, flexible, and lean implementation of core jQuery designed specifically for the server. In *GitHub repository*. GitHub. <https://github.com/cheeriojs/cheerio>
- Nakayama, H., Kubo, T., Kamura, J., Taniguchi, Y., & Liang, X. (2018). doccano: Text annotation tool for human. In *GitHub repository*. GitHub. <https://github.com/doccano/doccano>
- Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. <https://doi.org/10.3115/v1/D14-1162>
- Peters, M. E., & Lécocq, D. (2013). *Content extraction using diverse feature sets*. 89–90. <https://doi.org/10.1145/2487788.2487828>
- Reis, E. S. D., Costa, C. A. D., Silveira, D. E. D., Bavaresco, R. S., Righi, R. D. R., Barbosa, J. L. V., Antunes, R. S., Gomes, M. M., & Federizzi, G. (2021). Transformers aftermath: Current research and rising trends. *Communications of the ACM*, 64(4), 154–163. <https://doi.org/10.1145/3430937>
- Richardson, L. (2021). Beautiful soup - a library that makes it easy to scrape information from web pages. In *PyPI repository*. Python Package Index. <https://pypi.org/project/beautifulsoup4/>
- Riebold, J. (2021). BoilerPy3 - python port of boilerpipe library. In *GitHub repository*. GitHub. <https://github.com/jmriebold/BoilerPy3>
- Scharl, A., Herring, D., Rafelsberger, W., Hubmann-Haidvogel, A., Kamolov, R., Fischl, D., Föls, M., & Weichselbraun, A. (2017). Semantic Systems and Visual Tools to Support Environmental Communication. *IEEE Systems Journal*, 11(2), 762–771. <https://doi.org/10.1109/JSYST.2015.2466439>
- Suárez, P. J. O., Sagot, B., & Romary, L. (2019, July). *Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures*. <https://doi.org/10.14618/IDS-PUB-9021>
- Swartz, A. (2021). html2text - a python script that converts a page of HTML into clean, easy-to-read plain ASCII text. In *GitHub repository*. GitHub. <https://github.com/Alir3z4/html2text>
- Wang, Z., Ho, S.-B., & Cambria, E. (2020). A review of emotion sensing: Categorization models and algorithms. *Multimedia Tools and Applications*. <https://doi.org/10.1007/s11042-019-08328-z>
- Weichselbraun, A., Brasoveanu, A. M. P., Waldvogel, R., & Odoni, F. (2020). Harvest - An Open Source Toolkit for Extracting Posts and Post Metadata from Web Forums. 2020

- IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 438–444. <https://doi.org/10.1109/WIIAT50758.2020.00065>
- Weichselbraun, A., Kuntschik, P., Fancolino, V., Saner, M., & Wyss, V. (2021). Adapting Data-Driven Research to the Fields of Social Sciences and the Humanities. *Future Internet*, 13(3). <https://doi.org/10.3390/fi13030059>
- Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., & Raffel, C. (2021). mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 483–498. <https://doi.org/10.18653/v1/2021.naacl-main.41>