elsearticle class for Typst

v2.0.1 2026-02-10 MIT

Typst template for Elsevier article submissions

Mathieu Aucejo

elsearticle is a Typst template that aims to mimic the Elsevier article LaTeX class, a.k.a. elsarticle.cls, provided by Elsevier to format manuscript properly for submission to their journals.

Table of Contents

Part I

About

Elsearticle is a Typst template that aims to mimic the Elsevier article LaTeX class, a.k.a. elsarticle.cls, provided by Elsevier to format manuscript properly for submission to their journals.

elsarticle is designed to be as close as possible to the original class, whose specification can be found in the documentation[1]. The template is still in development and may not be fully compatible with all Elsevier journals.

This manual provides an overview of the features of the elsarticle template and how to use it.

To mimic the look and feel of the original elsarticle.cls, the following fonts must be used:
- XITS[2] and XITS Math for the best compatibility with the original elsarticle.cls;
- STIX Two Text[3] and STIX Two Math as a reasonable alternative;
- New Computer Modern and New Computer Modern Math are also defined to avoid compilation errors if the previous fonts are not available.

> The template is provided as is by the Typst community and is not affiliated with Elsevier.

---

[1]http://assets.ctfassets.net/o78em1y1w4i4/3ro3yQff1q67JHmLi1sAqV/1348e3852f277867230fc4b84a801734/elsdoc-1.pdf

[2]https://www.ctan.org/pkg/xits

[3]https://github.com/stipub/stixfonts

Part II

Usage

II.1 Using elsearticle

To use the elsearticle template, you need to include the following line at the beginning of your typ file:

```
#import "@preview/elsearticle:2.0.1": *
```

II.2 Initializing the template

After importing elsearticle, you have to initialize the template by a show rule with the #elsearticle command. This function takes an optional argument to specify the title of the document.

```
#show: elsearticle.with(
  ...
)
```

#elsearticle takes the following arguments:

```
#elsearticle(
  (title): none,
  (authors): (),
  (institutions): (),
  (abstract): none,
  (journal): none,
  (keywords): (),
  (format): "preprint",
  (numcol): 1,
  (line-numbering): false
)[body]
```

┌─ Argument ──┐
│ (title): none   `str` │
│   │
│   Title of the paper   │
└──┘

┌─ Argument ──┐
│ (authors): ()   `array` │
│   │
│   Array containing the list of authors of the article. Each author is defined as a dictionary │
│   with the following keys: │
└──┘

- `name`: Name of the author (required) `str` | `content`
- `affiliations`: List of affiliations of the author (required) `array`
- `corresponding`: Corresponding author (optional, default: false) `bool`
- `email`: Email of the corresponding author (optional) `str` | `content`

```
authors: (
    (
        name: "J. Doe",
        affiliations: ("a", "b"),
        corresponding: true,
        email: "jdoe@univ.edu",
    ),
    (
        name: "J. Smith",
        affiliations: ("b",),
    ),
)
```

--- Argument ---

`(affiliations): ()`   `dictionary`

Dictionary containing the list of affiliations of the article. Each affiliation is defined as a key-value pair, where the key is a `str` representing the affiliation ID and the value is a `str` | `content` giving the name and the address of the affiliation.

```
affiliations: (
    "a": [Institution A, City A, Country A],
    "b": [Instrution B, City B, Country B],
)
```

> If the paper has only one author, `institutions`must be:
> - `("",)` in the `authors` `array` of `dictionary`
> - `("": [Institution name])` in the `institutions` `dictionary`
>
> to conform with the Elsevier template requirements.

--- Argument ---

`(abstract): none`   `content`

Abstract of the paper

┌─ Argument ───┐
│ `(journal)`: `none`   `str` │
│   │
│   Name of the journal   │
└──┘

┌─ Argument ───┐
│ `(keywords)`: `()`   `array` │
│   │
│   List of the keywords of the paper   │
│   │
│   Each element of the `array` is a `str` representing a keyword   │
│   │
│   ┌──┐ │
│   │ keywords: ("Keyword 1", "Keyword 2")   │ │
│   └──┘ │
└──┘

┌─ Argument ───┐
│ `(format)`: `"review"`   `str` │
│   │
│   Format of the paper. Possible values are "preprint", "review", "1p", "3p" and "5p" │
└──┘

┌─ Argument ───┐
│ `(numcol)`: `1`   `number` │
│   │
│   Number of columns of the paper. Possible values are 1 and 2   │
│   │
│   According to the documentation of `elsearticle.cls` (see here[4]), the number of columns is related to the format of the paper: │
│   • 1p: Single column only   │
│   • 3p: Single or double column possible   │
│   • 5p: Double column only   │
│   │
│   To avoid unexpected behaviors, the value of the `numcol` argument is set to 1 by default and restricted to 1 or 2. │
└──┘

┌─ Argument ───┐
│ `(line-numbering)`: `false`   `bool` │
│   │
│   Enable line numbering in the document   │
└──┘

II.3 Additional features

The elsearticle template provides additional features to help you format your document properly.

II.3.1 Appendix

The template allows you to create appendices using the `#appendix` environment. The appendices are then numbered with capital letters (A, B, C, etc.). Figures, tables and equations are numbered accordingly, e.g. Eq. (A.1).

---

[4]https://assets.ctfassets.net/o78em1y1w4i4/3ro3yQff1q67JHmLi1sAqV/1348e3852f277867230fc4b84a801734/elsdoc-1.pdf

To activate the appendix environment, all you have to do is to place the following command in your document:

```
#show: appendix

// Appendix content here
```

II.3.2 Subfigures

Subfigures are not built-in features of Typst, but the elsearticle template provides a way to handle them. It is based on the subpar package that allows you to create subfigures and properly reference them.

To create a subfigure, you can use the following syntax:

```
#subfigure(
  figure(image("image1.png"), caption: []), <figa>,
  figure(image("image2.png"), caption: []), <figb>,
  columns: (1fr, 1fr),
  caption: [(a) Left image and (b) Right image],
  label: <fig>
)
```

The #subfigure function is a wrapper around the #subpar.grid function. The numbering is adapted to the context of the document (normal section or appendix).

II.3.3 Equations

The equations are numbered with the format "(1)", "(2)" in normal sections and with the format "(A.1)", "(A.2)" in appendices. In addition to these numbering patterns, the elsearticle template provides the #nonumeq to create unnumbered equations. The latter function can be used as follows:

```
#nonumeq[$
  y = f(x)
  $
]
```

II.4 Cross-referencing

The supplement of cross-references is set to "none" by default to avoid unexpected behavior when referencing more than one item. However, you can revert this setting by using the following syntax after the template definition:

```
#set ref(supplement: auto)
```

```
#set ref(supplement: auto)
```

Part III
Dependencies

The elsearticle template depends on the following packages:

- `equate:0.3.2`: For equation handling and numbering.
- `subpar:0.2.2`: For subfigure handling and numbering.