

Using $\text{\LaTeX}$ (in linguistics)

András Bárány, University of Edinburgh
andras.barany@ed.ac.uk

24 June 2026
2026 SGSAAH summer school, Glasgow

1 Introduction

$\text{\LaTeX}$

$\text{\LaTeX}$ is a **typesetting system** with which you can create many different types of documents

- You can write **scientific papers** and **presentations** with it
- You can use it to write your **dissertations**
- But you can also write letters, reports, books, ...

Plan for today

Today, we'll cover the basics of using $\text{\LaTeX}$ as well as some more advanced topics.

- This session is intended to be **interactive** – ask questions!
- I will present a few topics and then you'll try things out yourself
- We will be using **Overleaf** (<https://www.overleaf.com/>)
- We'll take breaks every 50 minutes or so

Contents

1	Introduction	1
2	Advantages of $\text{\LaTeX}$	1
3	Basics	3
4	Bibliographies	7
5	Other useful concepts and packages	10
6	Complex documents	16
7	Slides and handouts	17
8	The rest	20
	References	21



Let's have a wee chat first!

2 Advantages of $\text{\LaTeX}$

What $\text{\LaTeX}$ is and isn't

$\text{\LaTeX}$ is **not** what you see is what you get (WYSIWYG), unlike LibreOffice or Word

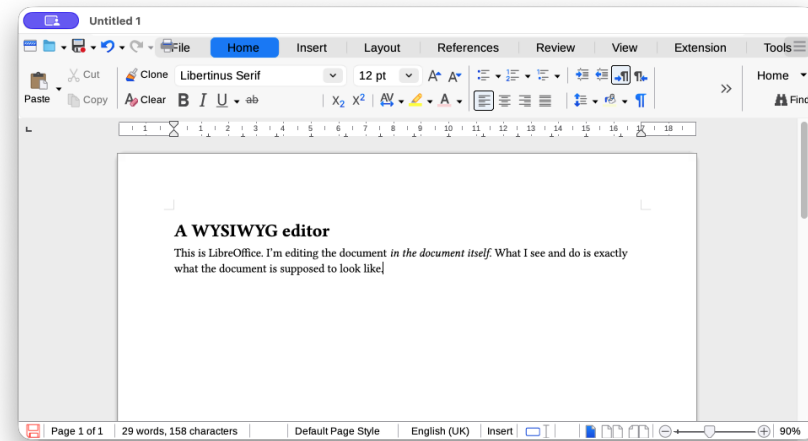


Figure 1: LibreOffice, a WYSIWYG editor

In $\text{\LaTeX}$, the ‘document’ is distributed across multiple files

- You edit a file with the suffix `.tex`
- This file is then **rendered** to give an **output file**
- The output is usually a **PDF** file
- You can include other files (images, bibliographies, etc.)

Why would we want this? Some **advantages** of this approach are ...

- Separating content from output gives you **flexibility**
- The same content can easily be rendered differently
- You can generate slides and handouts **from the same source**
- Editing a `.tex` file is **more precise**

What $\text{\LaTeX}$ can do

$\text{\LaTeX}$ is great for writing complex documents, such as a **PhD dissertation**

- Excellent support for bibliographies (**bib $\text{\LaTeX}$**)
- It easily handles **large documents** with many chapters
- Easy contents, list of figures, abbreviations, ...
- Unicode support (e.g. IPA and Hangul), numbered examples, glosses, diagrams, syntax trees, OT tableaux, ...
- It is **very stable** and **consistent** across versions

- (1) Korean (from <https://en.wiktionary.org/>)
안녕하세요 [an $\text{\LaTeX}$ has $\text{\LaTeX}$ ej $\text{\LaTeX}$ o] ‘hello’



Figure 2: A complex figure with **overlays** (changes from slide to slide)

(2) Hungarian

Ez egy komplex-ebb példa.
 DEM INDF.DET complex-CMPR example
 ‘This is a more complex example.’

List of abbreviations

ACC = accusative, CMPR = comparative, DEM = demonstrative, DET = determiner, INDF = indefinite, WYSIWYG = what you see is what you get.

3 Basics

A simple document

A tex-file consists minimally of a **preamble** and **document** environment

- The **preamble** specifies the form and properties of the output (lines 1–2 in Listing 1)
- The **document** environment specifies the **content** (lines 3–5 in Listing 1)

```
1 \documentclass{article}
2
3 \begin{document}
4 Hello, world.
5 % This is a comment.
6 \end{document}
```

Listing 1: A minimal document

Commands and environments

LaTeX syntax uses **commands** and **environments** to structure documents

- A command starts with a ‘\’ followed by a name and zero or more arguments

- For example, `\textbf{bold}` gives **bold**

An **environment** is surrounded by `\begin{...}` and `\end{...}`

- This **list** is an ‘`itemize`’ environment
- It is surrounded by `\begin{itemize}` and `\end{itemize}`
- A `tex` file’s content is in a document environment (Listing 1)

Lines starting with ‘`%`’ are **comments** — $\text{\LaTeX}$ ignores them

Simple formatting

Text is formatted using **commands** and **environments**

- You *emphasise* text with the command `\emph{...}`: `\emph{emphasise}` gives *emphasise*
- Use `\textbf{...}` for **bold text**
- Use `\textsc{...}` for SMALL CAPITALS
- Use `\texttt{...}` for typewritten text
- Use `\footnote{...}` for a footnote

```

1 \begin{itemize}
2   \item You \emph{emphasise} text with the command
3     \verb+\emph{...}+: \verb+\emph{emphasise}+ gives \emph{emphasise}
4   \item Use \verb+\textbf{...}+ for \textbf{bold text}
5   \item Use \verb+\textsc{...}+ for \textsc{small capitals}
6   \item Use \verb+\texttt{...}+ for \texttt{typewritten text}
7 \end{itemize}

```

Listing 2: Simple formatting

Titles and section headings

- Sections and subsections are numbered automatically
- The title consists of a document title, the author’s name and a date

Your turn!

You’ll edit your first document now!

- Create an account on <https://www.overleaf.com/>
- Then join this project: <https://edin.ac/4aijlGK>
- Make a copy of the project: **File** → **Make a copy**

```

1 \documentclass{article}
2
3 \title{A simple document}
4 \author{Your Name}
5 \date{\today}
6
7 \begin{document}
8
9 \maketitle
10
11 \section{A section header}
12
13 \subsection{A subsection header}
14
15 \end{document}

```

Listing 3: A simple document with a title and sections

— Your task —

1. Add a title, your name, and a date (you can use `\today`)
2. Add two sections and a few lines
3. Add a title (`\maketitle`) and a table of contents (`\tableofcontents`) (these are added where you enter the commands)
4. Add a unnumbered list (an `itemize` environment)
5. Add a numbered list (an `enumerate` environment)

Labels and references

Sections are numbered automatically — and we can refer to them easily

- A section command can be followed by `\label{...}`
- As labels, I use the type of command and helpful name
- For an introductory section, it could be `\label{sec.introduction}`
- The section can be referred to using the command `\ref{...}`
- `\ref{...}` takes the label of the section as its argument

— Your task —

1. Add a label to one of your sections using `\label{...}`

2. Refer to the section using `\ref{...}` somewhere else in your document
3. Consider the output and how you could improve it

Packages

What we have seen so far is useful but limited

- **Packages** extend the functionality of $\text{\LaTeX}$ in many ways
- Packages are added in the document's preamble with `\usepackage{...}`
- Listing 4 shows how to add the package `cleveref`

Your task

1. Add the package `cleveref`
2. Now replace your previous command `\ref{...}` with `\Cref{...}`
3. Try the commands `\cpageref{...}` and `\cref{...}` as well

```
1 \usepackage{cleveref}
```

Listing 4: Adding packages

Typefaces and languages

When writing about linguistics, you might present data from different languages

- Even if you only write about English, you might use the IPA
- Different typefaces have different support for distinct scripts
- I recommend open-source typefaces with good Unicode coverage
- Libertinus (Serif/Sans) or Fira Sans are great choices (they cover (most of) the IPA, lots of diacritics, support small capitals etc.)

Your task

1. Go to Wikipedia and search for 'International Phonetic Alphabet'
2. Copy some IPA characters (e.g. $\text{\textcircled{e}}$ or $\text{\textcircled{d}}$) into your `tex` file on Overleaf

3. Do the characters show up in the output?

Chances are that (some of) your IPA characters **did not appear** in the output

- This is because the default $\text{\LaTeX}$ typeface does not have those glyphs
- We will include a different typeface (as a package) to resolve this

Your task

1. Include the package `libertinus-otf` in your document (this sets the typeface ‘Libertinus Serif’ as the default)
2. Include the package `luatexko` to show the Korean characters

Document classes

In Listing 1, we set the `documentclass` property to ‘article’

- This is just one of several distinct classes $\text{\LaTeX}$ offers
- The `article` class is great for scientific articles
- Each class has particular properties for that type of document
- The `letter` class adds an addressee, a signature, folding marks, etc.
- For your dissertation, you can choose the `book` or `scrbook` class

Your task

1. Now change the document class to ‘scrbook’ and observe what changes
2. Change the document class to ‘scrartcl’ and observe what changes

4 Bibliographies

Bibliographies with *biblatex*

Academic writing requires referencing existing work

- The best way to do this in $\text{\LaTeX}$ is with `biblatex`
- This package provides ways of formatting citations and reference lists

- It supports a range of different citation and bibliography styles
- It will help you to use systematic references and avoid errors

To use biblatex, we need a **bibliography file** with the suffix `.bib`

- Listing 5 is an example of a reference of the type ‘book’
- The file `main.bib` in your Overleaf project shows a few others

```

1 @book{Spencer1991morphological,
2   author = {Spencer, Andrew},
3   title = {Morphological theory},
4   date = {1991},
5   location = {Oxford},
6   publisher = {Blackwell},
7 }
```

Listing 5: A biblatex entry (part of `main.bib`)

Then we need to add biblatex and refer to the bib file

- Line 1 in Listing 6 specifies an **optional argument**
- This optional argument selects a particular citation and referencing style
- Line 2 in Listing 6 tells biblatex where to look for data

```

1 \usepackage[style=unified]{biblatex}
2 \addbibresource{main.bib}
```

Listing 6: biblatex settings

Your task

1. Include biblatex in your tex document as in Listing 6

Now we can start **citing** work included in `main.bib`

- biblatex offers distinct citation commands
- `\textcite{Spencer1991morphological}` gives Spencer (1991)
- `\parencite[cf.][120]{Spencer1991morphological}` gives (cf. Spencer 1991: 120)
- `\cite[cf.][]{Spencer1991morphological}` gives cf. Spencer 1991

Citation commands can also take multiple **cite keys** (the labels of the entries)

Your task

1. Create a citation command for some entries in `main.bib`
2. Start by typing your citation command and ‘{’ and see what happens

Now that you’ve cited something, it’s time to print a list of references

- Line 1 in Listing 7 specifies sorting by name, year, title
- Line 2 includes the bibliography proper

```
1 \newrefcontext[sorting=nyt]
2 \printbibliography
```

Listing 7: Include a bibliography

Your task

1. Include the bibliography in your document
2. Change the style option to reading
3. Change the style option to numeric
4. Change it back to unified

Managing a bibliography

To use `biblatex`, you obviously need the right bibliographical entries

- You can edit your file manually (I do that; it works fine!)
- You can also use a [reference manager](#) such as `zotero`

There are many benefits to using a reference manager

- You minimise errors such as mismatches between citations and references
- Having one entry which you reference minimises typos etc.

Advanced features of `biblatex`

Another way to minimise errors is to use [cross-references](#) in `biblatex`

- Listing 8 shows an `incollection` entry
- It references another entry, a `collection`, in `main.bib`
- If you cite several chapters of one volume, they can refer to the volume
- You avoid repeating the same information in each entry

```

1 @incollection{KaanChun2018syntactic,
2   author = {Kaan, Edith and Chun, Eunjin},
3   title = {Syntactic adaptation},
4   date = {2018},
5   pages = {85--116},
6   crossref = {FedermeierWatson2018current},
7   doi = {10.1016/bs.plm.2018.08.003},
8 }

```

Listing 8: A biblatex entry of type incollection

```

1 @book{SikukuDiercks2026object,
2   author = {Sikuku, Justine and Diercks, Michael},
3   title = {Object marking in {Lubukusu}},
4   subtitle = {At the interface of pragmatics and syntax},
5   date = {2026},
6   location = {Berlin},
7   publisher = {Language Science Press},
8   pubstate = {inpress},
9 }

```

Listing 9: A biblatex entry with a particular pubstate

You can also specify particular states of publication (e.g. ‘in press’)

Your task

1. Cite SikukuDiercks2026object and check the reference list

5 Other useful concepts and packages

More on different languages: babel

You might be working in languages other than English, too

- We can specify a document’s language(s) with the babel package
- The package takes one or more languages as options

Your task

1. Add `\usepackage[british]{babel}` to your preamble
2. Now change the language to french or hungarian
3. Observe the magical changes!

4. Switch back to british

Quoting with *csquotes*

Changing the language with `babel` has other effects, too

- The package `csquotes` provides commands for quoting text

Your task

1. Load the package `csquotes`
2. Write a sentence surrounded by `\enquote{...}`
3. Observe the output and then change the language to french

Tables and figures

Other common environments include tables and figures

- These types of environments are called ‘floats’ in $\text{\LaTeX}$
- $\text{\LaTeX}$ decides on where to place them (this can take getting used to)
- Floats can have labels and captions (see Figure 1 and Listing 10)
- Including images as in Listing 10 requires the `includegraphics` package

```

1 \begin{figure}
2   \includegraphics[width=.75\textwidth]{img/libreoffice.png}
3   \caption{LibreOffice, a \gls{WYSIWYG} editor}\label{fig.libreoffice}
4 \end{figure}

```

Listing 10: The $\text{\LaTeX}$ code for Figure 1

The most modern package for including tables is `tabulararray`

- Creating tables can be tedious, but the result is worth it
- Listing 11 shows a table environment surrounding a `tblr`
- The specification ‘`{lcr}`’ indicates **three columns**

Column 1	Column 2 (this is wider)	Column 3
one	two	three
four	five	six

Table 1: This is a table

- The first is left-aligned (l), the second centred (c), the third right-aligned (r)
- Columns are separated by ampersands (&), rows end with \\

```

1 \begin{table}
2   \begin{tblr}{lcr}
3     \toprule
4     Column 1 & Column 2 (this is wider) & Column 3 \\
5     \midrule
6     one      & two                      & three \\
7     four     & five                     & six   \\
8     \bottomrule
9   \end{tblr}
10  \caption{This is a table}\label{tbl.tblr}
11 \end{table}

```

Listing 11: The $\text{\LaTeX}$ code for Table 1

Your task

1. Load the package tabularray
2. Load the booktabs library with `\UseTblrLibrary{booktabs}` in the preamble
3. Create a table environment
4. Create a table using the `tblr` environment
5. It should have at least two columns and lines
6. Add a caption and a label
7. Reference the label somewhere in your document
8. Add a list of tables using `\listoftables`



Where does the table appear? Is it where you want it to be?

Abbreviations and glosses

$\text{\LaTeX}$ can also handle acronyms and abbreviations for you

- We'll use three packages: `glossaries`, `glossaries-inline`, `leipzig`
- `leipzig` provides commands for common glossing abbreviations
- `\Acc{}` gives ACC, `\Cmpr{}` gives CMPR etc.

- The glossaries package provides [a list of abbreviations](#)!

You can also specify your own abbreviations (we'll see how in a second)

- `\gls{DOM}`, [on first use](#), gives 'differential object marking (DOM)'
- `\Gls{DOM}`, on first use, gives 'Differential object marking (DOM)'
- `\glsdesc{DOM}` gives 'differential object marking'
- When you use `\gls{DOM}` again, it will just give 'DOM'

Listing 12 shows how to load packages for glosses and abbreviations

- Line 5 shows how to define a new acronym
- The first argument of `\newacronym` defines the label, the second the output

```

1 \usepackage{glossaries}
2 \usepackage{glossary-inline}
3 \usepackage{leipzig}
4 \makeglossaries
5 \newacronym{DOM}{DOM}{differential object marking}

```

Listing 12: Loading packages for glossing and defining an acronym

Your task

1. Load the packages shown in Listing 12
2. Define a new acronym that you'd like to use

Wherever you want, you can include a list of abbreviations in your document

Your task

1. Refer to your acronym using `\gls{...}` in your document
2. Try out `\glsdesc{...}` or `\Gls{...}`
3. Include a list of abbreviations using `\printglossary`
4. Change the language to french and observe what changes
5. Add the option `[style=inline]` and observe what changes

Numbered examples

In some areas of linguistics, **numbered examples** are very common in papers

- These often include interlinear glosses, too, as in (2)

(2) Hungarian

Ez egy komplex-ebb példa.
 DEM INDF.DET complex-CMPR example
 ‘This is a more complex example.’

- There are various packages that handle numbered examples and glosses
- $\LaTeX$ takes care of continuously numbering examples
- You can give examples labels and refer to them

I use the package `expex-glossonly` with the `linguex` option

- In your Overleaf project, copy `expex-glossonly.tex` to load it

```

1 \ex.\label{ex.test}
2   This is a test
3
4 \ex.\label{ex.complex}
5   \a.\label{ex.subex}Another test
6   \b. Yet another test

```

Listing 13: Two examples

The output of Listing 13 is shown in (3) and (4), see also (4-a)

(3) This is a test

(4) a. Another test
 b. Yet another test

Each example or subexample can include glosses, see Listing 14

Your task

1. Create an example, with glosses or not
2. Create another one with at least three subexamples (you can re-use `\b.`)
3. Reference the example and one of the subexamples

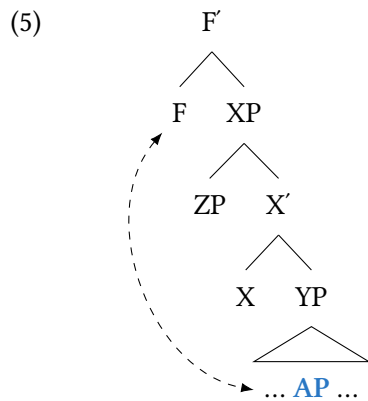
```

1 \ex.\label{ex.hungarian}
2   \begin{gls}
3     \gls{preamble} Hungarian//
4     \gls{a} Ez egy komplex-ebb példa.//
5     \gls{b} \Dem{} \Indf.\Det{} complex-\Cmpr{} example//
6     \gls{f} \enquote{This is a more complex example.}//
7   \end{gls}

```

Listing 14: A glossed example, the code for (2)

A tree using the *forest* package



```

1 \begin{forest}
2   [F']
3     [F,name=f ]
4     [XP
5       [ZP ]
6       [X'
7         [X ]
8         [YP
9           [ \dots{} \alert{AP} \dots{}
10            ,roof,name=yp ]
11         ]
12       ]
13     ]
14   ]
15   \draw [Latex-Latex, dashed, bend right=60]
16     (f) to (yp.west);
17 \end{forest}

```

Listing 15: Creating (5) using *forest*

For the double arrow in (5), you also need to load `\usetikzlibrary{arrows.meta}` in addition to the `forest` package.

LaTeX engines

Let's have a look at the settings on Overleaf (**Settings** → **Compiler**)

- The default setting on Overleaf is pdfLaTeX
- We have been using LuaLaTeX to render our output
- LuaLaTeX has great support for Unicode and modern typefaces

Error messages

There will be errors, there will be **many** errors

- Error messages can be difficult to interpret at first
- A common error is 'Undefined control sequence'

Your task

1. Add `\textbf{bold}` to your document and press 'Compile'
2. Click the red box on the top right
3. Fix the error in the line mentioned

6 Complex documents

Distributing content over various files

If you're working on a big project, you can split it into **distinct files**

- A common approach: a main file with the preamble and overall structure
- Each chapter is a separate file that is fed into the main file
- This can make rendering faster if you only include what you need

Your task

1. Include the file `additional-section.tex` using `\include{...}`
2. Now change the command to `\input{...}` and observe what changes

You can also use this to re-use your preamble in several documents

- This way, you can re-use the same settings over and over

- If you want to change something, you only need to do it in one file

— Your task —

1. Copy your preamble into a new file on Overleaf
2. Create a new tex file in which you include the preamble
3. Only use `include` and a document environment
4. Change which document to render (**Settings** → **Compiler**)

- If you have complex tables or figures, you can also save them separately
- You can include them using `\input{...}`

7 Slides and handouts

beamer

beamer is a document class for making presentations with slides

- In beamer documents each slide is a frame environment

```

1 \begin{frame}{This is the frame title}
2
3   This is the content of the slide.
4
5   In \texttt{beamer}, you can highlight using the \alert{alert} command.
6
7 \end{frame}
```

Listing 16: A beamer frame, creating a slide in a presentation

— Your task —

1. Open `beamer.tex` in your Overleaf project
2. Create a frame (a slide) in beamer

In slides, you can use **overlays**, meaning changes from slide to slide

- In a list, for example, you might not want each point to appear at once
- You can control this using the command `\pause`

Your task

1. Create a new frame in your beamer document
2. Insert a numbered or unnumbered list
3. Include a `\pause` command before one of the `\items`
4. Now add `<2>` to one of the `\items` — what happens?
5. Highlight something using `\alert<3>\{...\}`

beamer documents can be structured like others

Your task

1. Add a few sections to your presentation
2. Add a citation to one of your slides
3. Create a frame including just `\printbibliography`
4. Include a table of contents
5. Include the packages you need for abbreviations ...
6. ... then use one and print a list of abbreviations

Blocks

beamer provides a few environments not used in other classes

- These might look slightly different depending on the [theme](#)

This is an exampleblock

This is an **exampleblock** environment.

This is a block

This is a **block** environment.

This is an alertblock

This is an **alertblock** environment.

The syntax for blocks is similar to that of frames but blocks need a title

- Blocks can include other environments such as lists

```

1 \begin{exampleblock}{This is an exampleblock}
2   This is an \alert{\texttt{exampleblock}} environment.
3 \end{exampleblock}

```

Listing 17: A block

Your task

1. Add two different blocks to your presentation

Slides and handouts

Presentations are great, but you might also want a handout

- A nice way to do this is using `beamerarticle`, Listing 18
- This package turns your slides into ‘regular’ content
- You can also specify content for the slides or the handout only

```

1 %\documentclass[aspectratio=1610,t,ignorenonframetext]{beamer}
2 \documentclass{scrartcl}
3 \usepackage{beamerarticle}

```

Listing 18: Setting up `beamerarticle`**Content by mode**

With option `ignorenonframetext` `beamer` ignores content between frames

- This allows you to add content that you only see in the handout
- You can also use `\mode<beamer>\{...\}` and `\mode<article>\{...\}`
- The content of these commands only shows up in the **specified mode**

Your turn

1. Add some text between frames
2. Add a list in `\mode<beamer>\{...\}` and one in `\mode<article>\{...\}`
3. Change to `beamerarticle` to see what changes

- You can set up distinct preambles for `beamer` and `beamerarticle`
- I generated these slides and handout from the same content

8 The rest

Installing $\text{\LaTeX}$ on your computer

Using Overleaf is fine, especially when you collaborate with others

- But it is relatively slow and you need an internet connection

You can install $\text{\LaTeX}$ on your computer and work locally

- I'd recommend this for bigger projects such as a dissertation
- You can also synchronise with Overleaf

Installing $\text{\LaTeX}$

If you have a Mac, install from <https://tug.org/mactex/>. If you use Linux, check your package manager for the texlive package. If you're using Windows, go to <https://tug.org/texlive/windows.html>.

$\text{\LaTeX}$ editors

When using $\text{\LaTeX}$ on your computer, you'll need a [text editor](#)

- Any editor that can edit text files is fine, you may have a preference
- There are also specialised editors, for example:
 - TeXworks (<https://en.wikipedia.org/wiki/TeXworks>)
 - Texmaker (<https://en.wikipedia.org/wiki/Texmaker>)
 - TeXstudio (<https://en.wikipedia.org/wiki/TeXstudio>)

These editors provide all kinds of useful support when working with $\text{\LaTeX}$

- [Completion](#) for references, commands, environments, ...
- Syntax highlighting help you see where you are
- Error consoles help you fix what's wrong

Further resources

There are [lots](#) of great packages that extend what $\text{\LaTeX}$ can do

- You can browse these and their documentation at <https://ctan.org/>

There are [great online resources](#) for learning more!

- Overleaf has a 'learning centre' at <https://learn.overleaf.com/>

- Overleaf also provides templates for types of documents and some journals
- <https://www.learnlatex.org/> is a great resource
- Your university might offer a template and resources

It's normal to run into trouble...

- You can google your error messages ...
- ... and you'll often end up on <https://tex.stackexchange.com/>
- You can ask your friends and colleagues to help you
- Or you can ask your favourite GenAI agent to help you

Abbreviations

ACC = accusative, CMPR = comparative, DEM = demonstrative, DET = determiner, INDF = indefinite, WYSI-WYG = what you see is what you get.

References

Spencer, Andrew. 1991. *Morphological theory*. Oxford: Blackwell.