

# Day 2, part 2. Building from templates

Digital Skills for Research

11 May 2022

## Contents

<b>1</b>	<b>Templates and big projects</b>	<b>1</b>
1.1	Principles of working on a multi-part book (like a thesis) . . . . .	1
<b>Task 2-4.</b>	<b>Create a document based on a template</b>	<b>3</b>

## 1 Templates and big projects

- Many publication venues provide their own  $\text{\LaTeX}$  templates. It makes sense to keep the original source file (.tex) in the project folder for reference.
- Notice lack of extensions in imports.

A zipped  $\text{\LaTeX}$  template may include the following files (Look into the source code to see another method to produce columns and control their size and positioning):

Computational Linguistics by MIT Press Journals

**clv3.cls** class file, used in `documentclass{clv3}`

**alocal.sty** style file (a patch for ArabTeX package) to be imported with `\usepackage{alocal}` if necessary

**compling.bst** bibliography style file, used as `\bibliographystyle{compling}`

**compling\_style.bib** example file with bibliography records

**COLI-manual3.tex** main source code

RANLP conference

- ranlp2021.sty
- acl\_natbib.bst
- anthology.bib
- ranlp2021.tex

### 1.1 Principles of working on a multi-part book (like a thesis)

- Adapting an existing template maybe easier than setting up your own from scratch.
- Using a template implies familiarity with the packages and commands.

**Specificity of a multi-part book-like project: What is different?**

- parts are typeset in separate files which are imported by the main code which defines overall parameters – e.g. thesis.tex importing

```
\include{contents}
\include{tables}
\include{chapters/dedication}
\include{chapters/1_intro}
```

- `\documentclass[12pt,a4paper,twoside,openright]{report}`

- additional elements in text:

– `\usepackage{epigraph} ... \epigraph{}{}`

– headers-footers:

```
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhead{}
% RO/LE: position of text on odd/even pages
\fancyhead[RO,LE]{Thesis Title}
```

– `\usepackage{todonotes} ... \todo[inline]{...}`

– a glossary to collect all abbreviations and acronyms

This comment is generated by a `\todo[inline]` command: Do I need a glossary?

```
\usepackage[acronym,automake]{glossaries}
\makeglossaries
```

– code listings using verbatim `\usepackage{listings}`

```
\lstinputlisting[language=Python]{args.py}
```

Listing 1: Types of arguments for a Python script from a file

```
import argparse
```

```
parser = argparse.ArgumentParser()
parser.add_argument('--raw', help="Path_to_input_folder", required=True)
parser.add_argument('--outto', '-o', help="Where_to_store_outputs?", require
parser.add_argument('--model', type=str, default='gsd', choices=['ancora',
parser.add_argument('--lempos', action="store_true", help="Boolean_flag")
parser.add_argument('--thres', type=float, default=0.5)
parser.add_argument('--langs', nargs='+', default=['ru', 'en'], help='Exam
```

```
args = parser.parse_args()
lang = args.lang
print(langs)
print(args.lempos)
```

Listing 2: Looping thru all folders under 'root'

```
for path, dirs, files in os.walk(args.root):
    last_folder = os.path.abspath(path).split('/')[−1]
    for i, file in enumerate(files):
        filepath = path + os.sep + file
```

## Listings

1	Types of arguments for a Python script from a file	2
2	Looping thru all folders under 'root'	2

## Task 2-4. Create a document based on a template

You can complete any of the following tasks. Each involves selecting a template and exploring the commands it uses by modifying it. Acknowledge the original template (add a link to it) in your source code

- Produce a CV using one of the templates  
<https://www.latextemplates.com/cat/curricula-vitae>
- Prepare a mock submission to RANLP using their [Overleaf template](#)
- Explore and set up a multi-chapter phd/master thesis template based on what is [provided by the University of Wolverhampton](#) and (most recently, in Oct 2021) adapted by [Tharindu Ranasinghe](#)