Cover Page

This is a cover page.

TEX101: Introduction to FAST Template

SiriusAhu
ID: 18359xxxxx

October 8, 2025

Abstract

This template demonstrates the baseline structure for FAST – the Coursework LaTeX toolkit focused on being Fast, Accessible, Stylish, and equipped as a Toolkit. Replace this text with your assignment abstract or a short summary of the work.

Contents

1 Introduction

This coursework template provides a clean starting point for LaTeX-based assignments. It demonstrates a modular structure where metadata, configuration, and content live in clearly separated files.

> *Adapt the structure to match your module requirements and remove demonstration sections you do not need.*

This is an example subsection to demonstrate the document structure and test the Table of Contents.

1.1 Subsection Example

TODO: It's a placeholder.

1.1.1 Subsubsection Example

TODO: It's another placeholder.

2 Mathematics

Equation (**??**) showcases how to typeset numbered equations alongside inline mathematics such as $\nabla \cdot \vec{F} = 0$.

$$\sum_{i=1}^{n} \vec{F}_i = m \cdot \vec{a} \tag{1}$$

For multi-line derivations, use the `align` environment:

$$E = mc^2, \tag{2}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_V \rho \, \mathrm{d}V = - \int_S \rho \vec{v} \cdot \mathrm{d}\vec{S}. \tag{3}$$

3 Figures

Use the `graphicx` package to insert figures (images). Store assets under `assets/images/` to keep the project tidy.

2 customized commands are provided for convenience.

First, `\picHere` wraps a full `figure` environment with caption and label support. (Figure **??** is an example.)

The basic usage pattern is:

- `\picHere{path}{width}{caption}{label}` inserts a centred figure with a caption and a cleveref-compatible label.

Here is an example of how to use it:

Listing 1: Using the figure helpers in a listing block.

```
\picHere{assets/images/github-icon.png}{0.7\textwidth}{Example figure included
    from external asset.}{fig:example-figure}
```

Figure 1: Example figure included from external asset.

While `\picHereSimple` is a minimal drop-in for decorative images that do not require referencing. (No caption or label, just the image.)

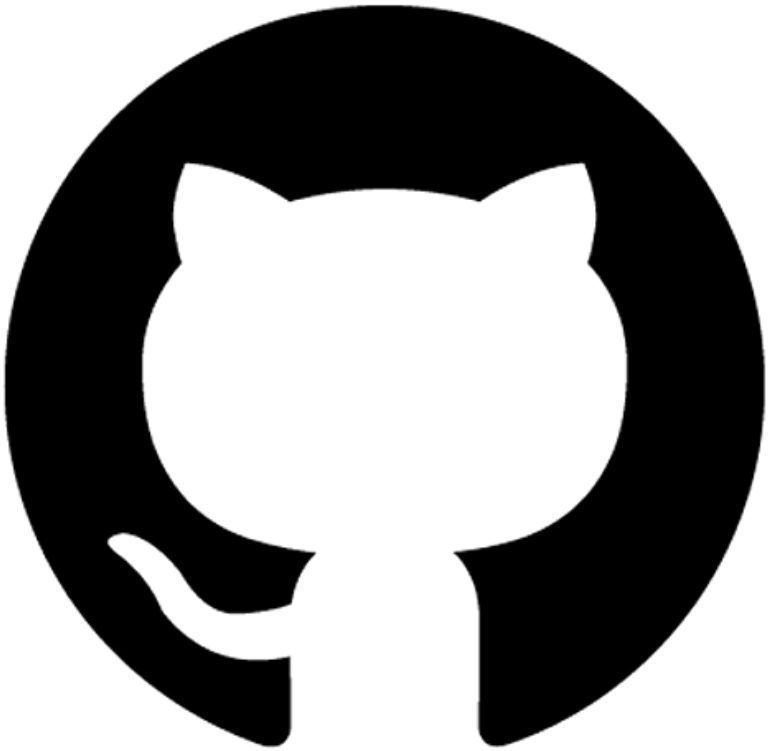
The usage pattern is:

- `\picHereSimple{path}{width}` gives you the same layout without caption or label when the image is purely decorative.

  Here is an example of how to use it:

Listing 2: Using the figure helpers in a listing block.

```
\picHereSimple{assets/images/github-icon.png}{0.7\textwidth}
```



4 Tables

Refer to Table ?? for a simple `booktabs` example.

Table 1: Summary of template goals.

Goal	Description
Lean	Fast compilation cycle with PDFLaTeX.
Easy to use	Small learning curve for newcomers.
Adaptable	Modular structure for course-specific tweaks.
Polished	Professional visual styling out of the box.

5 Code Listings

Source code is rendered using the `listings` package with the default style defined in `config/style.tex`. Adjust `\TemplateCodeListingStyle` in `options.tex` to point to a different style.

?? demonstrates an inline listing embedded directly in the document for short examples. The more extensive Python and C++ modules now live in Appendix ??, keeping the main narrative focused while still providing full source listings for reference.

Listing 3: Running total helper implemented in modern JavaScript.

```
export function runningTotal(values) {
  let total = 0;
  return values.map((value) => {
    total += value;
    return total;
  });
}

console.log(runningTotal([4, 8, 15, 16, 23, 42]));
```

6 Referencing

Manage bibliography entries in `bib/references.bib`. The template uses `biblatex` with the `biber` backend for flexible citation styles.

A Appendix Example

Appendices are input after `\appendix` is declared in `main.tex`. Use this space for supplementary derivations, raw data, or extended proofs that support the main text.

A.1 Supplementary Code Listings

The full Python and C++ utilities referenced in **??** are provided here for completeness.

Listing 4: Python helper for computing descriptive statistics.

```python
from statistics import median


def describe(values: list[float]) -> tuple[float, float]:
    mean = sum(values) / len(values)
    return mean, median(values)


if __name__ == "__main__":
    mean, med = describe([4, 8, 15, 16, 23, 42])
    print(f"mean={mean:.2f}, median={med}")
```

Listing 5: C++ program computing the arithmetic mean of a sample.

```cpp
#include <iostream>
#include <numeric>
#include <vector>

double mean(const std::vector<int>& data) {
    return std::accumulate(data.begin(), data.end(), 0.0) / data.size();
}

int main() {
    const std::vector<int> samples{4, 8, 15, 16, 23, 42};
    std::cout << "mean=" << mean(samples) << '\n';
}
```

A.2 Just some text

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor

in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.