

同濟大學

XXXX 课程报告

XXXXXX 进展调研

学院 电子信息工程学院

专业 数据科学与大数据技术

学号 ×××××××

姓名 XXX

2024 年 12 月 21 日

目录

1	模板说明	3
2	一些插入功能	3
2.1	插入公式	3
2.2	插入图片	3
2.3	插入文本框	4
2.4	插入表格	4
2.5	插入代码	4
2.6	定理环境	5
2.7	插入伪代码	6
2.8	插入参考文献	6
3	写在最后	7
3.1	主要参考	7

1 模板说明

本模板主要适用于一些课程的平时论文以及期末论文，默认页边距为 2.5cm，中文宋体，英文 Times New Roman，字号为 12pt（小四）。

编译方式：xelatex -> bibtex -> xelatex*2

默认模板文件由以下四部分组成：

- main.tex 主文件
- reference.bib 参考文献，使用 bibtex
- TongjiReport.sty 文档格式控制，包括一些基础的设置，如页眉、标题、学院、学号、姓名等
- figures 放置图片的文件夹

第一次使用时需前往 TongjiReport.sty 对标题、姓名、学号、院所、页眉等进行设置，设置完后即可一劳永逸，封面 LOGO 亦可替换。

默认带有封面页以及目录页，页码从目录页开始。

2 一些插入功能

2.1 插入公式

行内公式 $v - \varepsilon + \phi = 2$ 。

插入行间公式如式 1：

$$v - \varepsilon + \phi = 2 \tag{1}$$

2.2 插入图片

Tongji 校徽如图 1 所示，注意这里使用了 `\autoref{}` 命令，也就是会自动生成“图”“式”等前缀，无需手动输入。

此外，模版同时提供了校徽，如图 2 所示，请根据实际需求使用。



图 1: 同济大学

插入上面图片的代码：



(a) 左子图



(b) 右子图

图 2: 校徽

```
\begin{figure}[!htbp]
  \centering
  \includegraphics[width =.4\textwidth]{figures/tongji_logo.pdf}
  \caption{同济大学}
  \label{Tongji}
\end{figure}
```

2.3 插入文本框

本模板定义了一个圆角灰底的文本框，使用简化命令`\tbox{}`即可，如果你不喜欢，可以前往 `TongjiReport.sty` 对其进行修改。

这是一个圆角灰底的文本框

2.4 插入表格

本模板文件如[表 1](#)所示。

2.5 插入代码

本模板有一种较为粗糙的代码高亮方式，使用`\begin{lstlisting}`模块来使用，以 C++ 为例，一下程序显示模块的参数，选择语言 `language` 为 C++，使用 `caption` 来

文件名	说明
main.tex	主文件
reference.bib	参考文献
TongjiReport.sty	文档格式控制
figures	图片文件夹

表 1: 本模板文件组成

指定代码块标题，具体为[language=C++, caption=My Code, label=lst:code]，C++ 代码简单高亮如下：

Code Listing 1: My Code

```

1      #include<iostream>
2      using namespace std;
3
4      // This is a sample struct
5      struct node
6      {
7          int x;
8          int y;
9      };
10     int main()
11     {
12         struct node p;
13         cout << "Enter x and y: ";
14         cin >> p.x >> p.y;
15         return 0;
16     }
```

2.6 定理环境

本模板提供了一些较常见的定理环境，如2.1、2.2、2.3、2.4、2.5、2.6、2.6等。使用方法为\begin{Theorem}、\begin{Lemma}、\begin{Corollary}、\begin{Proposition}、\begin{Definition}、\begin{Example}、\begin{proof}。

定理 2.1. *Test*

引理 2.2. *Test*

推论 2.3. *Test*

命题 2.4. *Test*

定义 2.5. *Test*

例 2.6. *Test*

证明. *Test*

□

2.7 插入伪代码

本模板还提供伪代码的插入, 使用`\begin{algorithm}`和`\end{algorithm}`模块, 具体使用方法请参考[LaTeX/Algorithms](#), 以下是 Dijkstra 算法的伪代码¹:

Algorithm 1 Dijkstra's Algorithm

Require: Graph $G = (V, E)$ with non-negative weights, source vertex s

Ensure: Shortest path distances $d[v]$ from source s to each vertex $v \in V$

```

1: Initialize distance array  $d[]$ , for each vertex  $v \in V$ , set  $d[v] = \infty$ 
2: Set  $d[s] = 0$ 
3: Initialize priority queue  $Q$ , insert all vertices with their distances
4: while  $Q$  is not empty do
5:   Extract vertex  $v$  with minimum distance from  $Q$ 
6:   for each neighbor  $u$  of  $v$  do
7:     Calculate alternative path distance  $\text{alt} = d[v] + \text{weight}(v, u)$ 
8:     if  $\text{alt} < d[u]$  then
9:       Update  $d[u] = \text{alt}$ 
10:      Decrease key of  $u$  in  $Q$  to  $\text{alt}$ 
11:    end if
12:  end for
13: end while
```

2.8 插入参考文献

直接使用`\cite{}`即可。

例如:

此处引用了文献 [\[1\]](#)。此处引用了文献 [\[2\]](#)

引用过的文献 [\[3\]](#) 会自动出现在参考文献中 [\[4\]](#)。

3 写在最后

3.1 主要参考

- Github: https://github.com/Jiazhen-Lei/SJTU_Course_Template_Latex
- Overleaf: <https://www.overleaf.com/latex/>

参考文献

- [1] Renjie Zhao, Fengyuan Zhu, Yuda Feng, Siyuan Peng, Xiaohua Tian, Hui Yu, and Xinbing Wang. Ofdma-enabled wi-fi backscatter. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2019.
- [2] Fengyuan Zhu, Yuda Feng, Qianru Li, Xiaohua Tian, and Xinbing Wang. Digiscatter: efficiently prototyping large-scale ofdma backscatter networks. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pages 42–53, 2020.
- [3] Xilong Wang, Jiaodi Zhou, Kaijun Su, Jinzhou Du, Longtao Wei, Xing Li, Juan Du, and Dongliang Lu. Atmospheric radioactive nuclide deposition on the coast of the maowei sea, northern beibu gulf, china. *Frontiers in Marine Science*, 10, June 2023.
- [4] Deng-Ping Fan, Ge-Peng Ji, Ming-Ming Cheng, and Ling Shao. Concealed object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6024–6042, October 2022.