

# Project1

## 文档预览器 document previewer

### 1. 项目介绍

通过预览器进行一个 markdown 类型文件路径的接收，通过渲染生成相应的 HTML 文件，利用 webview 功能打开对应的网页。

### 2. 实现方式

Node 结点：

int type 存储相应的 markdown 语法类型用于解析

int flag 列表缩进标志

string content 存储解析出的语句

string address 存储图片和文件的地址（如果有的话）

Node\* next 链表结构的实现

Class markdown 定义一个 markdown 类

markdown(const string& filename) 自定义构造函数，传入选择的文件路径

void operate() 对 markdown 文件进行处理（此处主要负责范围较大的语法结构的解析，不考虑行内具体的内容。如标题、引用、代码块、列表、段落等），生成对应的链表

void read(string str) 在一个新的链表中对行内语法进行解析，如强调、行内代码、图片、链接等

以及针对两个链表中含有的不同语法结构进行转化的 transform 函数，生成相应的 HTML 文件

主要采取的数据结构为链表，本次共设计了两个链表，一个用来存储遍历呈现在行间的语法结构，并针对不同的语法类型考虑是否要再解析行内代码（比如代码块不需要解析，而列表则需要）。一个针对行内代码进行解析的新链表相当于是该链表在特定结点上的子链表，当然也可以在一个结构体中实现。通过对于两种类型的链表的遍历完成相应的转化

打开原 markdown 文件后逐行遍历，优先存储行间语法，生成相应的链表，再针对特定的语法类型调用 read 函数，生成解析行内语法的子链表。分别调用对应的 transform 函数将文本转换为 HTML 格式，加上前缀后缀等。最后生成一个完整的 HTML 文件

### 3. Gui 实现

选用的软件为 QT

界面中通过选择需要解析的文件的地址，将路径和原文件的内容显示在文本框中，点击转换的按钮生成相应的 HTML 文件同时在浏览器中打开