**Simple File Reader**

Reading is an essential skill for a developer. It helps you improve your knowledge as well as stay up to date with happenings around you. However, the hustle and bustle of everyday living can result in not having time to read. To solve this challenge, the *file\_reader* program was developed.

**Table of Contents**

* [Description](https://chat.openai.com/c/a3cb2979-5e38-4773-8d80-c268b31f4170#description)
* [Installation](https://chat.openai.com/c/a3cb2979-5e38-4773-8d80-c268b31f4170#installation)
* [Usage](https://chat.openai.com/c/a3cb2979-5e38-4773-8d80-c268b31f4170#usage)
* [Supported File Types](https://chat.openai.com/c/a3cb2979-5e38-4773-8d80-c268b31f4170#supported-file-types)
* [Controls](https://chat.openai.com/c/a3cb2979-5e38-4773-8d80-c268b31f4170#controls)
* [Contributing](https://chat.openai.com/c/a3cb2979-5e38-4773-8d80-c268b31f4170#contributing)
* [License](https://chat.openai.com/c/a3cb2979-5e38-4773-8d80-c268b31f4170#license)

**Description**

This is a simple command-line program that can read PDF, text, and Word document files. It utilizes the Python libraries **pypdf**, **docx**, and **pyttsx3** for reading PDFs, Word documents, and converting text to speech, respectively. The main program can be found in the **file\_reader.py** file. Helper functions such as reader controls that allow the user to adjust the volume, choose the speed rate of the reader as well as the voice to use, can be found in the **controls.py** file. Tests for the robustness and correctness of the program can be found in the **test\_reader.py** file. Dependencies required for the program to run successfully can be found in the **requirements.txt** file.

In choosing the text-to-speech (TTS) engine to use, the python module pyttsx3 was chosen due to its simplicity of use and availability for offline use. Thus users can use the program with or without connecting to the internet.

Going forward, additional functionality will be added to enhance the reading experience of the user such as OCR recognition, customizable hotkeys, text summarization, text-to-speech markup support, automatic language detection together with a nice user interface (UI).

**Installation**

To use the *file\_reader* program, follow these steps:

1. Clone this repository to your local machine:

git clone https://github.com/your-username/file-reader.git

1. Install the required dependencies:

pip install –m requirements.txt

**Usage**

After installation, you can run the program using the following command:

python file\_reader.py <file\_path> <options>

Replace **<file\_path>** with the path to the file you want to read. Additionally, you can specify optional parameters:

* **volume**: Adjust the volume of the speech (0-100). Negative values cannot be provided.
* **rate**: Adjust the speech rate (0-100).
* **voice**: Select a voice for speech synthesis (currently supports only two voices: male[id=0] and female[id=1].

For example:

python file\_reader.py example.pdf volume=70 rate=80 voice=1

The program will run successfully even without specifying any additional options. The default settings will be used in this case

`For example:`

`python file\_reader.py <file\_path>`

**Supported File Types**

The program supports the following file types:

* **PDF**: Use **\*.pdf** files.
* **Text**: Use **\*.txt** files.
* **Word Document**: Use **\*.docx** files.

**Controls**

The **controls.py** file provides additional functionality to control the reading experience. You can adjust the volume, speech rate, and select different voices for speech synthesis.

**Contributing**

Contributions are welcome! If you'd like to contribute to this project, please follow these steps:

1. Fork the repository.
2. Create a new branch (**git checkout -b feature/new-feature**).
3. Make your changes.
4. Commit your changes (**git commit -am 'Add new feature'**).
5. Push to the branch (**git push origin feature/new-feature**).
6. Create a new Pull Request.

**Testing**

Several tests were written to check the robustness, correctness and reproducibility of this program.

These tests can be found here [\*\*Tests\*\*](test\_filereader.py).