

# Speech-to-Text Converter using Python

NTHU General Education Logical Thinking of Informatics

Group members:

109006234	邱慧莉 Anastasia Riana
109006238	楊玉蘋 Annabella Putri Dirgo
109006278	白亦喬 Mercy Angelina

# Table of Contents

Group Member Introduction .....	2
I. Background .....	3
II. Project Purpose.....	4
III. Feature 1 (Speech-to-Text).....	5
IV. Feature 2 (Translate) .....	7
V. User Interface .....	8
VI. Conclusion.....	9
References.....	10

## **Group Member Introduction**

### **Anastasia Riana** – International Bachelor Degree Program Group B EECS

My name is Anastasia Riana from Jakarta, Indonesia and I am currently enrolled at NTHU International Bachelor Degree Program majoring in Electrical Engineering and Computer Science. I chose to enroll in this class not just to fulfill my General Education credits but also to get more familiar with multiple programming languages. Whenever I am not studying, I often keep myself busy with learning multiple skills online such as foreign language, programming soft-skills, etc. I also enjoy my spare time watching movies and TV shows. By joining this class, I hope to have a deeper understanding about Python and be able to utilize its potential in the future as a professional working in the tech industry.

### **Annabella Putri Dirgo** – International Bachelor Degree Program Group B EECS

My name is Annabella Putri Dirgo, an International Bachelor Program majoring in EECS undergraduate student and I'm from Indonesia. I like to play the guitar and watch movies. I also enjoy reading books, especially classic one. Jane Austen being my favorite novelist.

I pride myself on being detail-oriented, analytical, and visionary. Thus, I'm happy to step into college life, which provides a range of opportunities to reach my dream. Post-college, I aspire to work in the technology field especially in machine learning and mobile app development.

### **Mercy Angelina** – International Bachelor Degree Program Group B Business Management

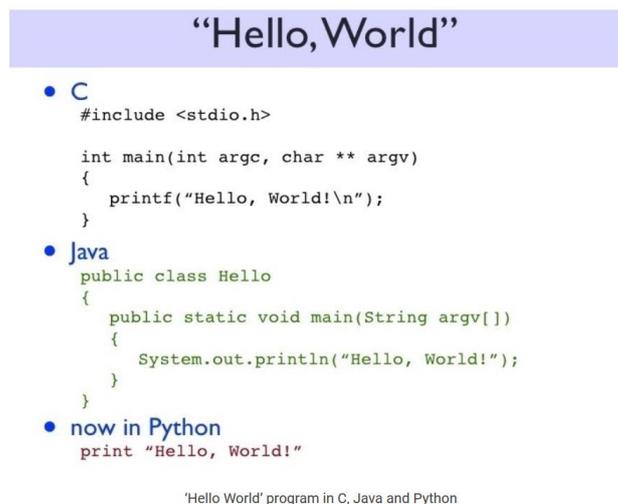
My name is Mercy Angelina, and I came from Surabaya, Indonesia. I am an International Bachelor Degree Program student majoring in Business Management. I am personally an organized, innovative, and dedicated person. In my leisure time, making DIYs, baking, watching movies, and learning new soft skills are the things that I usually do. I find some contentment in turning my hobbies and interests into a business.

I am particularly interested in supply chain management and information system. Choosing Business Management as my major is the first step that will lead me into my desired goal and dream career. I enrolled in this GE course with the hope that I can learn the basics of programming, which will be beneficial for my future career.

## I. Background

Most people's journey towards learning how to program starts with choosing the first programming language that suits them. When we knew that there was a General Education Course that teaches the basics of programming in Python, we joined the class almost immediately due to its increasing popularity. In terms of overall popularity, according to Hughes (2020) <sup>1</sup>, Python now ranks as the third most widely-used programming language worldwide with a rating of 12.21%. In second place is Java, which has a rating of 12.53%, and C, which remains the world's most popular programming language with a rating of 16.48%. Other than that, Python is well known to be incredibly easy to learn for beginners as it is a simplified syntax.

For example, if we want to write a simple program called 'Hello World', different languages require a certain amount of code lines to run the program. However, for Python, we can simply write one line of code to display 'Hello World' in the terminal. Here is the comparison for the same 'Hello World' program but in C language and Java.



```
“Hello, World”
```

- C

```
#include <stdio.h>

int main(int argc, char ** argv)
{
    printf("Hello, World!\n");
}
```
- Java

```
public class Hello
{
    public static void main(String argv[])
    {
        System.out.println("Hello, World!");
    }
}
```
- now in Python

```
print "Hello, World!"
```

'Hello World' program in C, Java and Python

Other than its simple nature, Python has so much more to offer. Because of the tremendous support from the Python community, we do not need to make our own libraries, like in other programming language such as C++, where we have to develop our own library. In Python, we can simply call the desired library or modules, which can save so much time.

---

<sup>1</sup> Hughes, O. (2020, December 4). *Python on track to be crowned 'programming language of the year'*. Retrieved from TechRepublic: <https://www.techrepublic.com/article/python-on-track-to-be-crowned-programming-language-of-the-year/>

## **II. Project Purpose**

As students, things that become our daily problems are the assignments that we need to finish on time and exams that we need to prepare for. Besides that, we might also have other things to do, for instance, doing the chores, doing other homework, etc. When we have many things to finish in such a short period, multitasking might be the best solution. Since most of the students' assignments are making essays or reports, our group tried to find a tool to make it easier for people to type what they think.

Therefore, after discussing, our group came up with the idea of making a speech-to-text recognition program with Python. By using this program, we can simply just say the words that we want to type and save them as a text file. When ideas suddenly come out at an unexpected time, we can directly make a note by converting our speech into text. This program will be handy if we want to type something while doing other things.

In addition to that, since our group members are all foreigners, we always have to do our assignments in other languages, such as English. Thus, we thought that adding a translation function would be very helpful. If we want to write essays, we can type with our native language first then use the translation function to change it to other languages.

### III. Feature 1 (Speech-to-Text)

Before running the code, we need to make sure that 'pip SpeechRecognition', 'pip deep-translator', and 'pip PyAudio' have been installed.

After a lot of research, we decided to use a library called PyAudio. This library is built to help programmers to play and record any type of audio we want.<sup>2</sup> We utilize PyAudio to catch the user's speech and store it into a variable within the code.

Python has multiple modules that we can utilize to recognize the user's speech. But there is one package that caught our attention that is: SpeechRecognition. Recognizing speech requires audio input which is provided by the PyAudio module, and SpeechRecognition makes retrieving this input really easy. Instead of having to build scripts for accessing microphones and processing audio files from scratch, SpeechRecognition will have you up and running in just a few minutes. The SpeechRecognition library acts as a wrapper for several popular speech APIs and is thus extremely flexible. One of these—the Google Web Speech API—supports a default API key that is hard-coded into the SpeechRecognition library.

In the base code, we make the 'record' function to record the voice through the microphone. When the 'Record' button is clicked, the microphone will be enabled. Then it will adjust the ambient noise before it starts recording the voice. We use a built-in function called record and set it to record the voice for 10 seconds. After that we use another built-in function called 'recognize\_google' to recognize the recorded voice. In our case, we set our program to recognize the voice in English. When the program successfully records the speech, it will let the user know by displaying 'Successfully recorded!' on the window.

---

<sup>2</sup> *pyaudio*. (n.d.). Retrieved from RIP Tutorial: <https://riptutorial.com/python/topic/10627/pyaudio>

```

def rec():
    r = sr.Recognizer()
    while True:
        with sr.Microphone() as source:
            r.adjust_for_ambient_noise(source)
            audio = r.record(source, duration=10)
        try:
            global txt
            txt = r.recognize_google(audio, language='en-US')
            msg.configure(text=txt)
            print(txt)
            Label(root, text='Successfully recorded!').grid(row=3, column=0, sticky='W')
        except Exception as e:
            print(e)
            break

```

After recording the voice, the recorded message can be viewed to make sure that the microphone caught the message that we desire. To save the recorded voice, we set the program to write a file called 'result.txt' to store the recorded message in the txt file. Whenever the microphone does not catch everything that the user say, they can edit the message in the left textbox and click the 'Save' button to rewrite it.

```

def view():
    x = str(txt)
    T.insert(END, x)

```

```

def save():
    with open(file='result.txt', mode='w') as fh:
        x = T.get("1.0",END)
        fh.write(x)

```

## IV. Feature 2 (Translate)

For this part of the program, we need to utilize a module called deep-translator module. There are a lot of modules that can translate, but one module that caught our attention is deep-translator. This module has multi language support and has automatic language detection. It is also easy to use and extendable. Its main feature is that its support for Google Translate comes in handy for our project which is the main reason why we chose to use this model.

For the translate feature, when the 'Translate' button is pressed, the program will create a file to save the translation result called 'trans\_result.txt'. Then the program will read the first line until the end of the text in the left textbox. The user can type the language that they want the message to be translated in the top right entry box. The program can translate the text to almost every language, but not the languages that are not alphabet-based, such as Mandarin, Korean, and Japanese. That is because the function 'fh.write' cannot be used to write the character in 'trans\_result.txt' that have been created from the 'with' command. After the message is successfully translated, the result will be stored in the 'trans\_result.txt' file. Every time the 'Translate' button is clicked, the program will automatically translate the message and then save the result on this file. Therefore, if the user translate the message more than once, the text stored in the 'trans\_result.txt' file will be the latest one.

```
def trans():
    with open(file='trans_result.txt', mode='w') as fh:
        x = T.get("1.0",END)
        y = str(transbox.get())
        trans = GoogleTranslator(source='auto', target=y).translate(x)
        fh.write(trans)
        print(trans)
        T1.insert(END,trans)
```

Whenever the user want to switch to another language to translate, they need to click the 'clear' button first then retype the language they want.

```
def clear():
    T1.delete('1.0', END)

root.mainloop()
```

## V. User Interface

According to Team (2021)<sup>3</sup>, user interface (UI) is a point in which human users interact with a computer with a goal to make the user's experience easy and intuitive. UI includes input devices like keyboard and mouse. Python provides a standard Python interface which is called tkinter (“Tk interface”). Tkinter is available on both Windows and Unix systems. Some methods that are available and applied in this program are grid, title, label, entry, and button.

Grid functionality is to put the widgets in a 2-dimensional table. Title is to make a name that will appear on top of the window. The label widget is to make display boxes in which programmers can place an image or text. The entry widget is used for an input. The button widget is used to add buttons in Python applications which are designed for the user to interact with.

```
import speech_recognition as sr
from deep_translator import GoogleTranslator
from tkinter import *

root = Tk()
root.geometry('500x200')
root.title('Speech-to-Text Recognition Program')

T = Text(root,height=5, width=30)
T.grid(row=1,column=0)

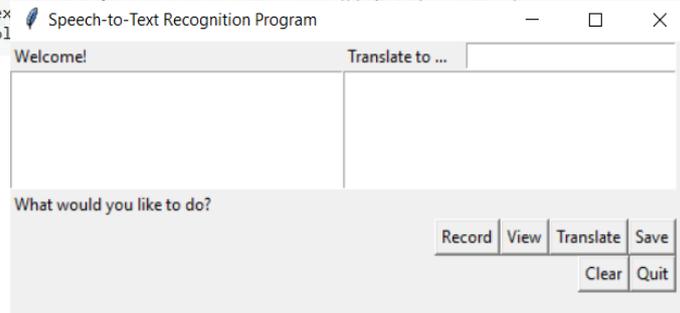
T1 = Text(root,height=5, width=30)
T1.grid(row=1,column=1)

msg = Label()
label = Label(root, text='Welcome!').grid(row=0, column=0,sticky='W')
label1 = Label(root, text='What would you like to do?').grid(row=2, column=0, sticky='W')

label2 = Label(root, text='Translate to ...').grid(row=0, column=1, sticky='W')
transbox=Entry(root,width=25)
transbox.grid(row=0, column=1, sticky='E')

gridframe = Frame(root)
for j in range(1):
    Button(gridframe, text='Record', command= lambda: rec()).pack(side=LEFT)
    Button(gridframe, text='View', command= lambda: view()).pack(side=LEFT)
    Button(gridframe, text='Translate', command= lambda: trans()).pack(side=LEFT)
    Button(gridframe, text='Save', command= lambda: save()).pack(side=LEFT)
gridframe.grid(row=5, column=1, sticky='E')

gridframe = Frame(root)
for j in range(1):
    Button(gridframe, text='Clear', command= lambda: clear()).pack(side=LEFT)
    Button(gridframe, text='Quit', command= lambda: quit()).pack(side=LEFT)
gridframe.grid(row=6, col
```



<sup>3</sup> Team, I. E. (2021, March 30). *What Is a User Interface? (Definition, Types and Examples)*. Retrieved from indeed: <https://www.indeed.com/career-advice/career-development/user-interface>

## **VI. Conclusion**

From this group project, we of course learn more about programming in Python, not to mention how to optimize the module that Python initially has to make a simple and useful program. Moreover, we also learned to make the user interface for the program. By doing this project, we realized that we actually do not need to write a difficult algorithm for the program. We discovered that Python is a really unique language compared to the others because we can use a lot of libraries and modules to improve our program, on top of that we can create our own modules and we do not need to declare all the variables that we use. Since the syntax and the structure of Python is really simple, we do not need multiple lines of code to do certain things.

In summary, our group hope that this program can bring benefits to a lot of people, especially for those who are very busy with their tasks and expecting to finish them all within a short time. By using this program, not only can we multitask, but we can also translate the speech that had been converted into text, to another language.

## References

- Amos, D. (n.d.). *The Ultimate Guide To Speech Recognition With Python*. Retrieved from Real Python: <https://realpython.com/python-speech-recognition/>
- Hughes, O. (2020, December 4). *Python on track to be crowned 'programming language of the year'*. Retrieved from TechRepublic: <https://www.techrepublic.com/article/python-on-track-to-be-crowned-programming-language-of-the-year/>
- LIVEWIRE. (2019, April 9). *Advantages of Python | Python Certification Course | Livewire Ambarnath, Mumbai*. Retrieved from Live Wire: <https://www.livewireindia.com/blog/python-certification-course-livewire-ambarnath/>
- pyaudio*. (n.d.). Retrieved from RIP Tutorial: <https://riptutorial.com/python/topic/10627/pyaudio>
- Rockikz, A. (2020, October). *How to Convert Speech to Text in Python*. Retrieved from The PythonCode: <https://www.thepythoncode.com/article/using-speech-recognition-to-convert-speech-to-text-python>
- Subramanian, D. (2020, April 7). *Easy Speech-to-Text with Python*. Retrieved from towards data science: <https://towardsdatascience.com/easy-speech-to-text-with-python-3df0d973b426>
- Team, I. E. (2021, March 30). *What Is a User Interface? (Definition, Types and Examples)*. Retrieved from indeed: <https://www.indeed.com/career-advice/career-development/user-interface>