How to set up an Anaconda environment for the Phyllis Voice programs (Windows)

Chris Venour, October 2020

Introduction

This document describes how to create an Anaconda environment on your Windows machine in which the Phyllis Voice programs will run. The Phyllis Voice programs are Python programs which depend on various Python modules, such as the `google-cloud-speech` and the `pyaudio` modules, in order to work. Therefore, you need to install these Python modules on your Windows machine before the Phyllis code can execute properly. To avoid breaking other Python programs on your Windows machine, the modules should be installed in an isolated Anaconda environment rather than directly within the Windows Command Prompt.

Why use Anaconda?

If you were to install the Phyllis Voice modules directly within the Windows Command Prompt, this could break other Python programs on your machine. This section attempts to explain why this could happen and how creating separate Anaconda environments avoids this problem.

Let's say that a year ago you wrote a Python program - let's call it program A - that uses version 1.15 of Google's TensorFlow library. However, today you want to write a different Python program - let's call it program B - that uses version 2.3 of TensorFlow.

If you open a Windows Command Prompt window and directly install version 2.3 of TensorFlow, it's possible, maybe even likely, that program A will no longer run on your machine because the older version of TensorFlow it was using (version 1.15) has been overwritten by the newer version (version 2.3).

Creating an Anaconda environment for program A and a separate Anaconda environment for program B will avoid this problem. The Anaconda environment built for program A will have TensorFlow 1.15 installed in it whereas the Anaconda environment for program B will have TensorFlow 2.3 installed in it and both programs will execute without error within their respective environments.

Install Anaconda

You need to install the Anaconda software on your Windows machine if you haven't already done so. Here are the steps to take to determine whether or not Anaconda is already on your machine:

1. Open a Windows Command Prompt.

2. Type in `conda`

If a page of text is output to the screen describing conda's command line arguments, then Anaconda is already installed on your machine and you can skip to the next section. However, if you get an error message saying "conda is not recognized as internal or external command", then Anaconda needs to be installed on your machine.

To install Anaconda on your machine, perform the following sequence of steps:

1. Go to the webpage [Anaconda Products](#)
2. Scroll down the page until you see the area shown in Figure 0-1.
3. In the Windows column, click "64-Bit Graphical Installer". This downloads an executable file.
4. Launch the installer.
5. Accept the defaults by clicking the 'Next' button but then STOP when you see the window shown in Figure 0-2.
6. Do not accept the default settings at this point. Instead, select the checkbox labelled "Add Anaconda3 to my PATH environment variable". After doing so, a warning message in red lettering will appear explaining that selecting this checkbox is "Not recommended". You can ignore this warning.
7. You can keep the other checkbox selected if you like (i.e. the checkbox with the label "Register Anaconda3 as my default Python 3.8").
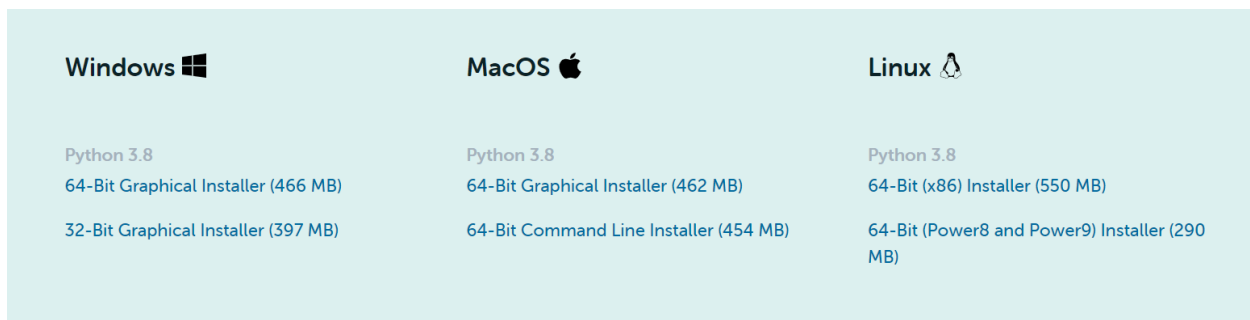8. Click the 'Install' button.

Windows ⊞	MacOS	Linux 🐧
Python 3.8	Python 3.8	Python 3.8
64-Bit Graphical Installer (466 MB)	64-Bit Graphical Installer (462 MB)	64-Bit (x86) Installer (550 MB)
32-Bit Graphical Installer (397 MB)	64-Bit Command Line Installer (454 MB)	64-Bit (Power8 and Power9) Installer (290 MB)

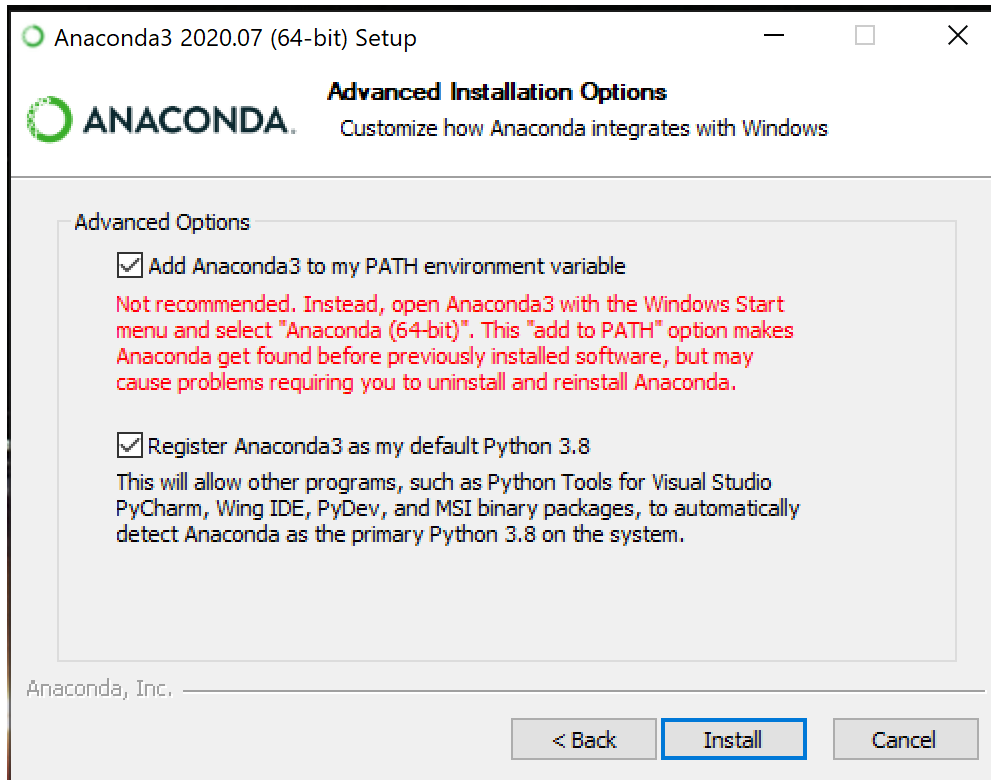*Figure 0-1 Downloading the Anaconda Windows installer*

Figure 0-2 Click on the "Add Anaconda3 to my PATH environment variable" option, in spite of the warning.

To make sure the installation was successful, open a Windows Command Prompt window and type:

```
conda
```

A page of text should appear on your screen describing conda's command line arguments.

Create an environment for the Phyllis code

Now that Anaconda is installed on your machine, you can use it to create an environment in which the Phyllis code can run. To create an Anaconda environment called 'phyllisenv', enter the following command in a Windows Command Prompt window:

```
conda create --name phyllisenv python=3.7
```

This command creates an environment, gives it the name 'phyllisenv' and installs Python 3.7 inside that environment. The Phyllis code uses the Google Cloud Speech API and this API supports Python version 3.5 or later, so installing Python 3.7 is a good choice.

Now enter the new Anaconda environment called 'phyllisenv' by typing the following in the Windows Command Prompt:

```
conda activate phyllisenv
```

The text "(phyllisenv)" should now appear in front of your Windows Command Prompt cursor. This indicates that you are inside the 'phyllisenv' environment.

Now that you're inside the phyllisenv environment, you can install the Python modules that the Phyllis code requires by entering the following commands:

```
pip install "google-cloud-speech==1.3.2"
pip install pydub
pip install pipwin
pipwin install pyaudio
pip install regex
pip install playsound
pip install "google-cloud-texttospeech==1.0.1"
```

Note that the version numbers that appear after the 'google-cloud-speech' and 'google-cloud-texttospeech' modules are important. If you don't include those version numbers when pip installing these modules, the latest versions of 'google-cloud-speech' and 'google-cloud-texttospeech' will install and the Phyllis code will not work. This is because the Phyllis code was implemented using older versions of these modules, so it's important to include the version numbers given above.

That's it! An Anaconda environment in which the Phyllis code can run has been created.

Install and initialize Google Cloud SDK

To use Google's Speech To Text and Text To Speech online services, Google's documentation explains that you need to (1) set up a Cloud Console project and (2) download a private license key. I have already done this for Seamatica, and so all you need to do is the following:

1. Download the Cloud SDK Installer
2. Launch the installer and follow the prompts.
3. After installation has completed, the installer presents several options:
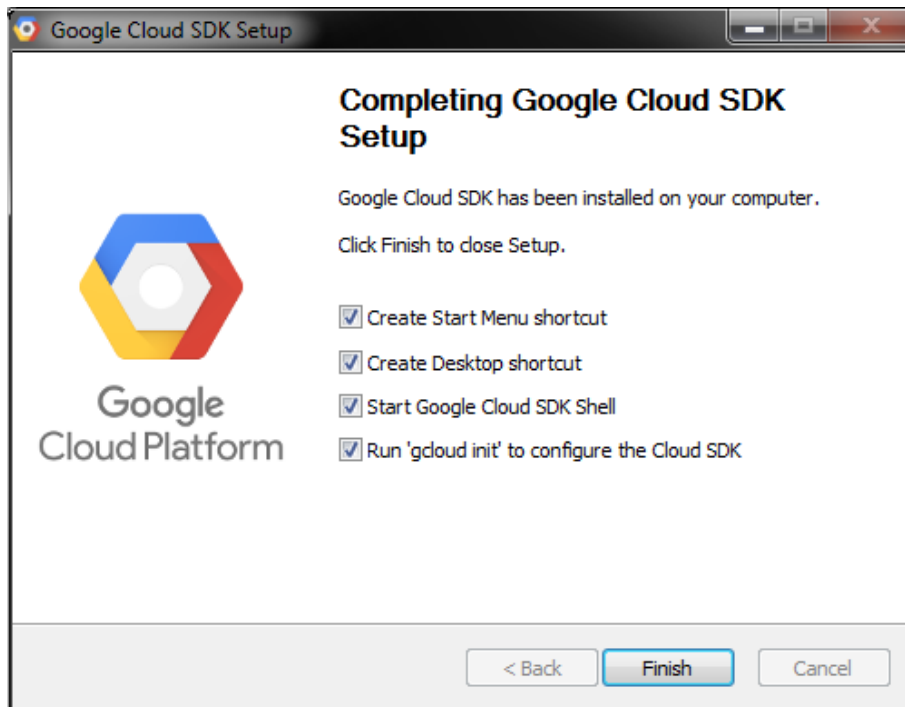
Make sure that the following are selected:

- Start Google Cloud SDK Shell
- Run 'gcloud init'

Test whether the Phyllis code works in 'phyllisenv'

To test whether the 'phyllisenv' environment works properly, follow this sequence of steps:

1. Open a Windows Command prompt.

2. Type the following command:
   ```
   conda activate phyllisenv
   ```

3. Change directories (cd) to the folder containing the Phyllis code.

4. Run a unit test of the NLU module by typing the following:

   ```
   python -m unittest unit_test_nlu_only.py
   ```