# The General Idea:

1. Make a preview file in Zoom, and download it as a .csv file. I would suggest renaming it something short and/or easy to remember.
2. You will need to put the preview file into the folder for the project.
3. Run it through SteUp.py, making sure you type in the file name correctly. (You do not need to include “.csv”)
4. It will run through it fairly quickly, but I would suggest setting it in the morning, and then going to grab your coffee or whatever.
5. Once it is done you should see another .csv file that has the same name, just with “Build” in front of it.
6. You will want to upload the file into a gsheet if you want to work from that. You also definitely need the file to remain in the folder.
   1. If you come across a contact that is now coming up with no contact info, Make sure you put “None” or a space, so the “full name” cell is not empty. When you start the next day, the program looks for empty cells to start from.
7. Run the file through “Builder.py”. You will need to input your Zoom credentials, and will potentially need to
8. Assuming that you are building out of G Sheets, make sure you download it as a .csv file at the end of the day to replace the original file in the program’s folder.
   1. If you start the program again and it is starting from the middle of the list, make sure you don’t have any empty name cells until the line that you want to start from.
9. Either at the end of the build or the end of each day of building, run the lines you have built through the Finisher.py.

# Set Up: Puts Headers in Order & Finds Company Websites

## Program name in Drive:

ZoomBuilder - SetUp.py

## What you need:

A Zoom preview file in a .csv format. You can move the columns and delete rows, but do not delete any columns, the program will remove any that are not normally needed.

## What it does:

1. The program will ask what file to read, you don’t need to include .csv, but make sure your spelling and capitalization match the file completely.
2. You will then be prompted to input if you want to replace any existing URLs that you might have. If you have no URLs this shouldn’t matter. It needs to be Y, y, N, or n.
3. The program then opens a chrome window and searches for the company websites.

## What it makes:

A new .csv file titled “Build\_[original file name].csv”

## Notes:

* You will probably want your files to have short, easy to type names. You can also rename the files in between programs, but I like leaving them alone since the added file names can help keep track of where the file is in the process.
* Make sure that you sort by company, reorganize, and delete any extra accounts as needed *before* the next step

# Builder: Automates Building from a Preview File

## Program name in Drive:

ZoomBuilder - Builder.py

## What you need:

* The “Build\_[original file name].csv” from the SetUp program
  + Required column labels:
    - Comments, Company, Full Name, First Name, Last Name, Title, Phone 1, Phone 2, Email, Person City, Person State, Website, Industry, Employee Size, Revenue, LinkedIn URL, HQ Address, HQ City, HQ State, Zip, Country
* The login credentials for the zoom account you will be using
* Access to the text message or email that Zoom sends the verification code to
* If you prefer building in GSheets, you need to also upload “Build\_[original file name].csv” in to a google sheet to build from. Otherwise, do not open the .csv file to build into it until after the program is logged in.

## What it does:

1. The program finds the first blank “Full Name” and starts from that row.
2. You will be prompted to input your Zoom login information. If you type something wrong here you will be able to correct the mistake later.
3. A chrome window will open, and the program will log in. You might have to complete a reCaptcha and tell it to log in a second time.
4. Zoom will then send an authorization code to the phone number or email associated with the account. You will be prompted by the program to input the code into PyCharm, but if you put it in the browser, simply input “done” for the program.
5. The program will now take the Zoom ID for the first individual and load a page that only contains the individual, so load times are greatly reduced.
6. Once you are done collecting the information for that individual, click the home button for Zoom. The program should detect the change within five seconds and move on to the next person automatically.

## What it makes:

The program does not create a file, all of the progress is you building into the sheet

## Notes:

* Double check the companies and titles as you work, the more days there have been since the original pull, the more likely it is for contacts to have changed.
* If you have a contact that is no longer valid or nothing pulls up for them, either delete the entire row or put something into the “full name” cell for the row. If you leave an empty cell the program will try to start there the next time you start it.
* Make sure you also collect the company information for the first contact per company. This is one reason to sort by company, you can now copy and paste the information to everyone else at the company.
* If you are building out of GSheets make sure you download the build at the end of the day and replace the file you were building out of. (Alternatively you can name them “Day1” “Day2” etc.)

# Finisher: Checks Emails & Finds/Verifies LinkedIn’s

## Program name in Drive:

ZoomBuilder - Finisher.py

## What you need:

* The finished part of the build from the builder step
* Your LinkedIn login credentials

## What it does:

1. First, the program will prompt you for the name of the file to read from, and verify that all column headers that are needed are there. The order and capitalization do not matter, but the headers all need to be present.
2. The program will then ask for your LinkedIn email and password, as well as how many times you want to run the emails through the bouncer. (Recommended 3-5 times)
3. First the email bouncer is run, checking those emails that bounced for however many times you told it to run.
4. The program will then open a chrome browser, log into the provided LinkedIn account, and search for the LinkedIn accounts for people.
5. Next the program will open another chrome browser to verify the LinkedIn accounts. This is the most time consuming portion.
6. Finally the program writes the output file with the results.

## What it makes:

The reformatted build file under the name “Finished\_[input file name].csv”. It should contain all of the same headers as the builder file, with two added columns for the email bouncer and LinkedIn verifier results.

## Notes:

* Required headers (order and capitalization do not matter): comments, company, full name, first name, last name, title, phone 1, phone 2, email, person city, person state, website, industry, employee size, revenue, linkedin url, hq address, hq city, hq state, zip, country
* LinkedIn verification is the slowest part of this program, I would definitely start this one before leaving at the end of the day and letting it run overnight.