# Introduction

Data entry is an unavoidable task which most has grappled with at some point in their personal life or professional career. While data entry is necessary, and sometimes integral to certain business functions, there is no denying that such tasks are **mundane, time-consuming and disengaging**.Yet, these traits make data entry a prime candidate for automation. You can easily use Python to automate data entry and **improve your productivity**.

In this article, we demonstrate how one can programmatically populate sales transaction data into an online invoice builder. The end product is a Python script that can repeatedly generate all your customer invoices with just the click of a button.

# What is PyAutoGUI?

[PyAutoGUI](https://pyautogui.readthedocs.io/en/latest/) is a very powerful library that **allows your Python script to control the mouse and keyboard to automate interactions** with other applications. The API is designed to be as simple. PyAutoGUI works on Windows, macOS, and Linux, and runs on Python 2 and 3. Here are some basic and commonly-used PyAutoGUI functions you should know about:

**Mouse Functions**

* moveTo(x, y, duration=num\_seconds): moves the mouse cursor to the provided x and y integer coordinates. The duration is a optional parameter which indicates how many seconds you want this movement to take.
* click(x, y): simulates a single, left-button mouse click after moving the mouse curser to the provided x and y coordinates on your screen.
* doubleClick(): performs a double click of the left mouse button. It also has optional **x**, **y**, **interval**, and **button** keyword arguments.
* scroll(): simulates a mouse scroll wheel. The number of 'clicks' scrolled is determined by the value of integer which you pass to it. The amount of scrolling in a "click" varies across platforms.

**Keyboard Functions**

* write(): This function types out the characters in the input string provided to the method. To add a delay interval in between pressing each character key, pass an int or float to the interval keyword argument.
* press(): This function simulate pressing a key down and then releasing it up. It is primarily used to activate special keyboard keys such as "tab", "insert" etc.

**Screen Coordinates**

Locations on your screen are referred to by X and Y Cartesian coordinates. The X coordinate starts at 0 on the left side and increases going right and the Y coordinate starts at 0 on the top and increases going down.

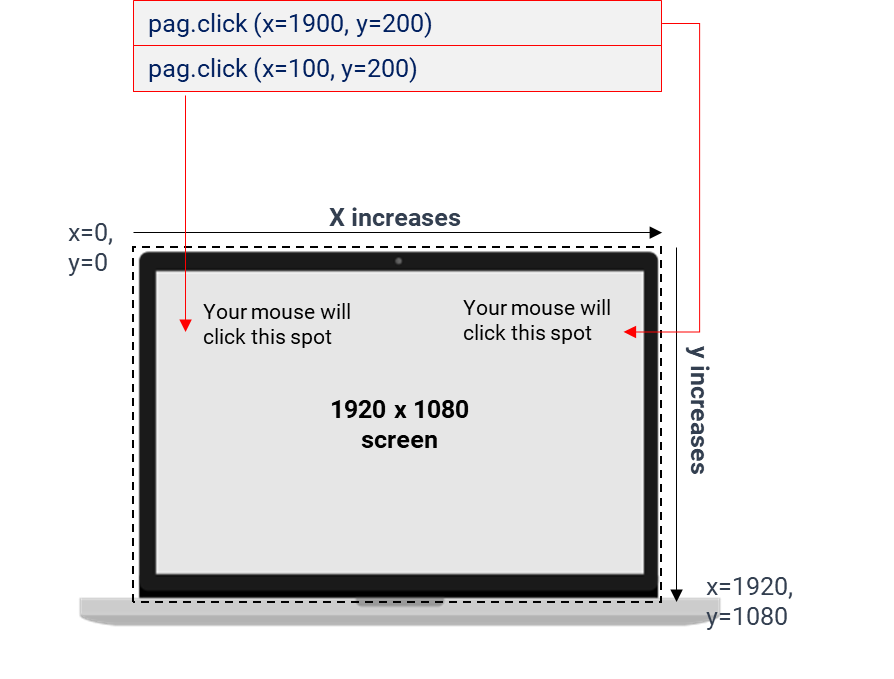


Figure 3: How to Pinpoint X and Y coordinates in PyAutoGUI

In summary, the **required x and y coordinates is highly dependent on your computer’s screen size**. For example, the specific X and Y coordinates to click a checkout button on Amazon would likely be different on a 1920 x 1080 screen versus a 1600 x 900 screen.

In the next 3 sections, we illustrate various PyAutoGUI use cases:

1. Filling in a form (Simple example)
2. Generate invoices
3. Other use cases of PyAutoGUI

# 1. Filling in a form (Simple example)

You would find that PyAutoGUI is an extremely easy-to-use library especially if you have already taken Heicoders Academy’s [**AI100: Python Programming & Data Visualisation**](https://heicodersacademy.com/AI100) course. Take a look at the sample code provided below, and you should have a good intuition of how PyAutoGUI can be used.

# 2. Automate Data Visualization

Most data analysis projects end with a presentation that contains lots of graphs. In my previous jobs, I’d update my Excel reporting and then manually make visualizations such as bar plots, piecharts, boxplots, and more.

This was time-consuming tough, so I looked for an alternative in Python. I found a couple of libraries that could help me automate plotting visualizations. I only had to read my updated Excel reporting with Python and by running a script all the visualizations would be created.

## How to solve this project

To automate data visualization, first, we need to write code that creates the graph we want (preferably in a Jupiter Notebook file). Then write down how the input data (Excel file) should look like, so you make sure the column names, the shape of the data, data type, and other important features remain the same in the next update.

Now you might be wondering “how can I make visualizations with Python?” Well, Python offers different options to make standard as well as interactive visualizations. [Here’s a guide to making visualizations with Matplotlib/Seaborn](https://towardsdatascience.com/a-simple-guide-to-beautiful-visualizations-in-python-f564e6b9d392) and [here’s another guide to making interactive visualizations with Pandas/Plotly](https://towardsdatascience.com/the-easiest-way-to-make-beautiful-interactive-visualizations-with-pandas-cdf6d5e91757).

That said, learning such libraries might take you a good number of hours. Here Mito can help too. [You can use Mito to make data visualization with a couple of clicks](https://docs.trymito.io/how-to/graphing) while generating the corresponding Python code. This code can be used to automate the creation of data visualization for future data.

# 3. Web Automation

One of the boring tasks I had to do as a junior data analyst was to upload files to a client’s website on a daily basis. The steps were quite simple: Go to website “X,” click on this button, select an option from a dropdown and click on upload.

Doing this could take 2 minutes or so, but I had to repeat these steps hundreds of times during the week. This is why I decided to automate this task with Selenium. This is a tool forcontrolling web browsers like Chrome through Python.

## How to solve this project

To automate any website with Python, first think of all the steps you would normally follow to do a task on a website. Your task might involve clicking on buttons, selecting elements within dropdowns, introducing text, scrolling up/down, logging in to pages, etc.

Once all the tasks are enumerated, replicate all the steps in Python using Selenium. Below is my [Python Selenium tutorial for beginners on YouTube](https://youtu.be/UOsRrxMKJYk) and [here’s a Selenium guide in Python](https://medium.com/swlh/web-scraping-basics-scraping-a-betting-site-in-10-minutes-8e0529509848). Both will help you create a bot in Selenium that will perform almost any task on a website as if you were controlling it yourself.