# ~Resume as dictionary~

A \*\*resume\*\* is to \*\*People Data Information Systems\*\* the same as \*\*Chuck Norris\*\* is to … ?

You got it, the most likely answer is: \*\*MOVIES\*\*. Predicting such responses has become quite commonplace, especially in the era of AI, hasn't it? 😂

Now, envisioning the resume of someone like Chuck Norris might also come naturally. But have you ever wondered how it appears when encoded within a program? :no\_mouth:

This tutorial release delves into Python's widely-used datatype: the Dictionary. Let's explore what it entails and witness a live demonstration of how a resume looks when represented as a dictionary in code below ↓↓↓

Did you notice the key-value pair "name": "Chuck Norris" in the provided code? Congratulations, you've just taken your first step towards comprehending data structures! :clap:

## Dictionary (datatype)

First thing first, what are datatypes? In programming languages, like Python, data types represent the kind of value that a variable can hold. Some common data types in Python include integers (whole numbers), floats (decimal numbers), strings (sequences of characters), lists (ordered collections of items), tuples (immutable ordered collections), dictionaries (unordered key-value pairs), and sets (unordered collections of unique items). Each data type has its own characteristics and behaviours, allowing for versatile programming and manipulation of data in Python.

Now, the main characteristics of the \*\*dictionary data type\*\* in Python include its ability to store data in key-value pairs, where each key is unique within the dictionary. Dictionaries are mutable, meaning they can be modified after creation. They provide fast access to values based on their associated keys, making them efficient for data retrieval. They can contain various data types as both keys and values, offering flexibility in organizing and manipulating data structures within Python programs.

## Best Practices (a.k.a. Lessons Learned)

- Knowing datatypes is the basic of knowing data structures; dictionary might become your best friend for its versatility and performance.

## Resources

- https://docs.python.org/3/tutorial/datastructures.html

# ~Run Streamlit locally~

Are you prepared to code and deploy your own \*\*People Data Information System\*\* in just 10 minutes?

Expanding upon the groundwork laid out in previous steps (refer to my prior posts or the app page), you're now ready to trial run the simplest Streamlit web application possible.

The objective here is merely to confirm the setup of your Desktop Developer Kit - nothing more!

Saying that, all that's required is to create your project name, activate Python, install Streamlit, and execute your application locally. For further details, read below↓↓↓:

https://advanced-project-template.streamlit.app/

## Name project

Begin by creating a folder on your computer and giving it a name of your choice. For instance, you could use ‘pdis’ as an abbreviation for your \*\*People Data Information System\*\*. Open VSCode and navigate to this specific folder within it.

This folder now serves as the primary location for your app project. P.S. At this moment, you need not deal with the Streamlit Advanced Project Template on GitHub, nor setup git at all.

## Activate Python

Once you've opened your project folder in VS Code, set Python as your programming language. This is done through the 'select interpreter' option, where you choose the Python version you installed earlier when creating your Conda environment in Anaconda.

From this point forward, VS Code will recognize the Python version and developer environment to use for your project folder.

## Install Streamlit

Within VS Code, via the terminal (usually PowerShell by default), install the Streamlit package using the \*\*‘pip install streamlit’\*\* command into your conda environment associated with your project. Afterward, you can type \*\*‘pip list’\*\* to view all Python packages installed for your specified environment.

Keep in mind that packages are installed for the designated conda environment, allowing you to create future projects that utilize the same Python setup associated with that conda environment.

## Run application

Within your project folder, create a new file—let's say ‘home.py’. Open this file and input: \*\*st.title(‘People Data Information System’)\*\*, then save. Congratulations, you've just wrote your first program!

Now, execute it via the terminal using the command \*\*‘streamlit run home.py’\*\*. If successful, your browser will launch, displaying your web app in action on your local machine (localhost).

## Best Practices (a.k.a. Lessons Learned)

- There are multiple ways to achieve the same outcome, and different online instructions are often based on differing assumptions.

- Perhaps therefore software installations (almost) never proceed as smoothly as described in tutorials (including this one :wink:).

- Take small steps and ensure that each one works; otherwise, unresolved technical issues will surface later on, commonly referred to as ‘technical debt’.

# ~101: Desktop Developer Kit~

So, lets’ starting building your \*\*People Data Information System\*\* with \*\*Streamlit Advanced Project Template\*\* together.

Before diving in, it's essential to ensure you have your minimal developer kit in place. Here's what you need to do:

- Begin by installing Anaconda, a Python software package manager.

- Next, install Visual Studio Code, a code editing software.

That's all it takes! For further details, please refer to the information below:

https://advanced-project-template.streamlit.app/

P.S. the demo-app under provided link might need you to wake it up first 😃.

## Minimal tech stack

There are some general assumptions to bear in mind. This tutorial and the provided template are based on my experience with Windows desktop. Different operating systems may necessitate different steps. Additionally, all the software utilized and provided is open-source, meaning there are no licensing costs involved. Furthermore, there are various other ways and tools available to set up your minimal developer kit that you can explore. With that said, I personally recommend the following:

### Anaconda

Anaconda serves as a distribution management system for Python (and R) programming languages. Simply go online, download it, and proceed with the installation. Create a new environment (known as a 'conda' environment), name it and opt for the installation of Python 3.11 (the latest version available). From now on, you can install your Python software packages using either the Anaconda desktop application or, more conveniently, via a command-line terminal. It's worth noting that Streamlit also supports Python 3.12, but I have yet to transition to it with my template and tutorial.

### Visual Studio Code

Visual Studio Code, often abbreviated as VS Code, is a feature-rich code editor supporting various functionalities such as debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded version control with Git, among others. To get started, head online and install it. Since you've already installed Python with Anaconda, there's no need to install it again with the editor. However, during the installation process, you may want to select the "Add path" option, which ensures that the default terminal knows about python packages downloaded later on.

## Best Practices (a.k.a. Lessons Learned)

- Avoid upgrading your software stack unless it's necessary or you intentionally wish to do so.

- Refrain from experimenting with alternative options until you have a solid understanding of the chosen approach.

## Resources

- <https://anaconda.org/>

- https://code.visualstudio.com/

# ~Streamlit Advanced Project Template~

\*\*People Data Information System\*\* + \*\*Streamlit Advanced Project Template\*\* = :hearts::hearts::hearts:

Recently I delved into learning how to build and deploy a simple full-stack web data application. To my surprise, the process wasn't quite as simple as online tutorials and influencers like to claim😃.

Despite the challenges, the positive reactions from others and my continual note-taking effort inspired me to compile a practical curriculum.

Stay tuned! I'll soon be releasing it as an open-source Streamlit template codebase, accompanied by a step-by-step online tutorial.

Feel free to take it apart, enhance it, and surpass my efforts 💪. I'll be delighted if it aids you in advancing your projects or acquiring useful knowledge.

## Focus

My primary goal with this initiative is to deliver step-by-step a performing and reliable application-independent codebase, incorporating essential custom components, all under the umbrella of what I'm calling the \*\*Streamlit Advanced Project Template\*\*.

Additionally, I aim to pass on general principles of data modelling and programming, which I recommend adopting early on. To complement this, I plan to include a mix of best practices for general project setup and configuration based on my experiences and research.

To set expectations, I won't delve much into component APIs or other well-documented technical topics. Similarly, while I won't talk about every background detail, I'll provide pointers for further independent research allowing you to modifying an existing template codebase without to much struggle :thumbsup:.

## Application

* 1. The code template and tutorial is going to be communicated around an imaginary \*\*People Data Information System\*\*. In short, about a system that allows users in different roles to share information and make decisions. Understand it as a demo topic that is comprehensive for board group of people while the application specifics will be irrelevant for the codebase template and tutorial itself.

## Audience

This tutorial is designed for individuals with a background in data analysis and interest in management, possessing basic programming skills but are new to Python and web development practices. Streamlit Advanced Project Template is tailored for those eager to prototype ideas or create simple, yet solid, data-orientend web application for sharing with others.

## Resources

- https://streamlit.io/

- https://www.python.org/

- https://code.visualstudio.com/

Comment any specific topics you like me to consider and follow for updates to come!