

## Root Directory

It is preferred to keep the root directory as the workspace directory. This is consistent with the running python package `src/my_package` out of the debugger. In this the code in `notebooks` and in `src` should all function the same, using the same relative paths.

```
import os
os.getcwd()
```

```
'/Users/michaelvolk/Documents/projects/Dendron-Template'
```

## Import Your My Package

```
from my_package.first_module import a_useful_function

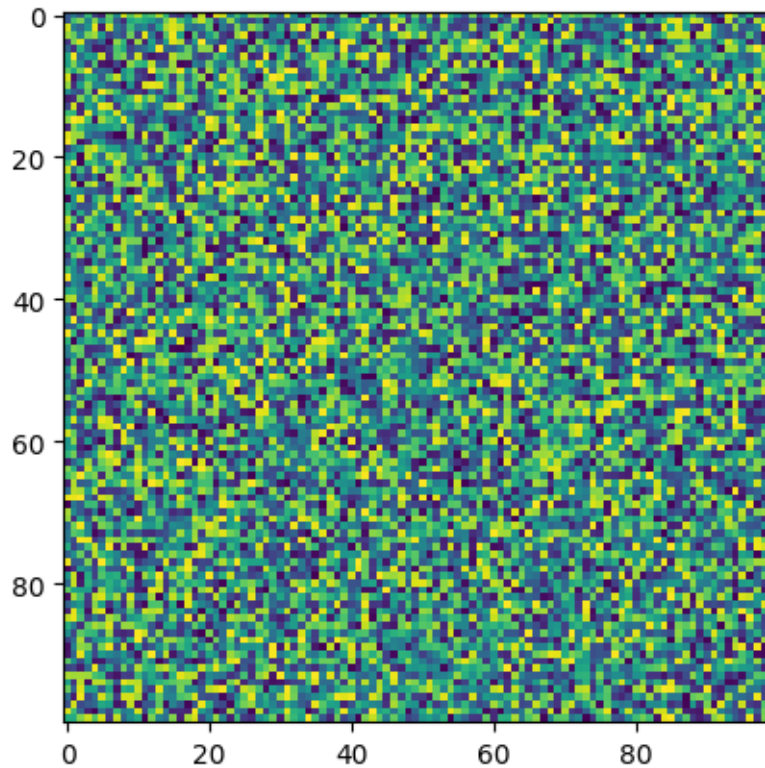
a_useful_function()
```

Hello world! `a_useful_function` rom `first_module.py`

## Saving Images

Jupyter notebooks are great for a reports, instruction, demonstrations etc. But they lack reproducibility and in my estimation, they make it difficult to build out layers of abstraction.

```
import matplotlib.pyplot as plt
import numpy as np
data = np.random.rand(100, 100)
plt.imshow(data, interpolation='nearest')
plt.show()
```



- If you want to display images in the notebooks, it is wise to save them to assets so they can be referenced at a later time. It is hard to tell when a previous analysis will be useful, and if notes are properly linked these results can be more easily found. For example if I want an image from notebook 1, and an image from notebook 2 to be brought into a final report I can go and save manually, but this would make it difficult to trace where the result came from. In the spirit of reproducibility, this is bad practice.
- We can see from this notebook that we have chance for making errors in over-writing images. If data is changed I will lose the previous image. If the `src/a_useful_function` is written well our functions that generate images will write them with unique names (e.g. time stamped).

```
plt.imshow('notes/assets/images/using-my-package-plot.png', data)
```