Python- create an importable package for our utils.py module so it's functions, constants and classes can be shared by applications in different folders

Ref: https://towardsdatascience.com/understanding-python-imports-init-py-and-pythonpath-once-and-for-all-4c5249ab6355

Task:
You have a collection of utilities which you would like to use in other programs.

To start, you created a module, utils.py, and placed it in the same directory as the application that's using it like so;

```
(data301) >tree
.
├── 32_categorical_variables.ipynb
└── utils.py
```

but things stop working as soon as you try to access utils from outside the current directory. Like so;

```
(data301) >tree
.
├── week_3
│   ├── 32_categorical_variables.ipynb
│   └── utils.py
└── week_4
    └── 41_wants_to_use_utils.ipynb
```

The problem is that utils.py is a module, it must be made into a package, which you then import. The following is 1 way to do this;

Solution:
1.Create a separate folder for utils.py. I will call it utils, but it can be whatever you want.
2.In the utils folder, add a file called __init__.py.

```
.
├── utils
│   ├── __init__.py
│   └── utils.py
├── week_3
│   └── 32_categorical_variables.ipynb
└── week_4
    └── 41_wants_to_use_utils.ipynb
```

__init__.py defines the objects exported from the modules in the directory (caveat: it can also define exports from modules in other directories)

3. Assuming utils.py has the following function that you would like to export:

```python
def generate_tshirt_order(numb_small=100, numb_medium=100, numb_large=100):
    '''
```

4. Add the following line to __init__.py if you want to import everything from the utils module

from utils.utils import *

#or you can import selective bits like this
#from utils.utils import generate_tshirt_order
#from utils.utils import PROCESSED_DATA

5. You now have a package called utils. To import it, the python interpreter must be able to find it. No worries, add your packages parent directory to the system path.

```
# I need to add the parent directory of utils in order to find it
#it happens to be up 1 directory.
import os
import sys
PROJECT_ROOT = os.path.abspath(os.path.join(
        os.getcwd(),
        os.pardir)
)

#only add it once
if (PROJECT_ROOT not in sys.path):
    sys.path.append(PROJECT_ROOT)
```