

Python In-class Programming Assignment Importing Your Own Custom Libraries

As we've seen, Python is a very modular language. We can import libraries to add functionality as we need it. For example, if we need a function to generate a random integer between 1 and 10, we can import the appropriate library to gain that capability.

```
import random

n = random.randint(1,10)
print(n)
```

Would this technique work on custom libraries that we write? YES! The easiest way to demonstrate this is with an example.

First, make sure your repo is up-to-date. Next, use the command below in a terminal window to copy a directory from your repo to a location of your choosing:

```
cp -R ~/repo201/tools/utls . <- In this case, I'm copying it to my current directory
```

This template directory contains two things:

1. A file called `mymath.py` with a single function in it: `isPrime(n)`.
2. A file called `__init__.py`, which Python uses as an indicator that the `utls` directory contains code which is designed to be imported into another program. There's nothing contained in this file (though for advanced uses, there could be), but it's required so Python knows this directory contains importable code.

To use the `isPrime(n)` function in a separate program we write, we just need to ensure that our program and the `utls` directory are both in the same directory. We then can import and use `isPrime(n)` using syntax like this:

```
from utls import mymath

candidate = 17
print(mymath.isPrime(candidate)) <- prints True
```

A few key points:

1. I named the directory `utls`, but you can name it anything you want.
2. I named the file `mymath.py`, but you can name it anything you want.
3. After import, to use a custom external function the syntax is:
`<filename>.<function name>`
4. The first time you run a custom function, Python will create a `__pycache__` directory within the `utls` directory. It's used by Python for housekeeping and you can safely ignore it.