Import Pitfalls

There are some very common import pitfalls that programmers fall into. We’ll go over the two most common here:

> - Circular imports
> - Shadowed imports

Let’s start by looking at circular imports

Circular Imports

Circular imports happen when you create two modules that import each other. Let’s look at an example as that will make it quite clear what I’m referring to. Put the following code into a module called a.py

```
# a.py
import b

def a_test():
    print("in a_test")
    b.b_test()

a_test()
```

Then create another module in the same folder as the one above and name it b.py

```
# b.py
import a

def b_test():
    print('In test_b"')
    a.a_test()
```

```
b_test()
```

If you run either of these modules, you should receive an AttributeError. This happens because both modules are attempting to import each other. Basically what's happening here is that module a is trying to import module b, but it can't do that because module b is attempting to import module a which is already being executed. I've read about some hacky workarounds but in general you should just refactor your code to prevent this kind of thing from happening

Shadowed Imports

Shadow imports (AKA name masking) happen when the programmer creates a module with the same name as a Python module. Let's create a contrived example! In this case, create a file named math.py and put the following code inside it:

```
import math

def square_root(number):
    return math.sqrt(number)

square_root(72)
```

Now open a terminal and try running this code. When I tried this, I got the following traceback:

```
Traceback (most recent call last):
  File "math.py", line 1, in <module>
    import math
  File "/Users/michael/Desktop/math.py", line 6, in <module>
    square_root(72)
  File "/Users/michael/Desktop/math.py", line 4, in square_root
    return math.sqrt(number)
AttributeError: module 'math' has no attribute 'sqrt'
```

What happened here? Well when you run this code, the first place Python looks for a module called "math" is in the currently running script's folder. In this case, it finds the module we're running and tries to use that. But our module doesn't have a function or attribute called sqrt, so an AttributeError is raised.

Wrapping Up

We’ve covered a lot of ground in this article and there’s still a lot more to learn about Python’s importing system. There’s PEP 302 which covers import hooks and allows you to do some really cool things, like import directly from github. There’s also Python’s importlib which is well worth taking a look at. Get out there and start digging in the source code to learn about even more neat tricks. Happy coding!