Importing Everything!

WE'LL COVER THE FOLLOWING

- Wrapping Up

Python provides a way to import **all** the functions and values from a module as well. This is actually a **bad** idea as it can contaminate your **namespace**. A namespace is where all your variables live during the life of the program. So let's say you have your own variable named **sqrt**, like this:

```
from math import sqrt
sqrt = 5
```

Now you have just changed the **sqrt** function into a variable that holds the value of 5. This is known as **shadowing**. This becomes especially tricky when you import everything from a module. Let's take a look:

```
from math import sqrt
sqrt = 5
print(sqrt(16))

# TypeError: 'int' object is not callable
```

To import everything, instead of specifying a list of items, we just use the "" *wildcard which means we want to import everything. If we don't know what's in themath** module, we won't realize that we've just clobbered one of the functions we imported. When we try to call the **sqrt** function after reassigning it to an integer, we find out that it no longer works.

Thus it is recommended that in most cases, you should import items from modules using one of the previous methods mentioned in this chapter. There are a few exceptions to this rule. Some modules are made to be imported using the “*” method. One prominent example is Tkinter, a toolkit included with Python that allows you to create desktop user interfaces. The reason that it is supposedly okay to import Tkinter in this way is that the modules are named so that it is unlikely you would reuse one yourself.

Wrapping Up

Now you know all about Python imports. There are dozens of modules included with Python that you can use to give extra functionality to your programs. You can use the builtin modules to query your OS, get information from the Windows Registry, set up logging utilities, parse XML, and much, much more. We will be covering a few of these modules in Part II of this book.

In the next chapter, we will be looking at building our own functions. I think you’ll find this next topic to be very helpful.