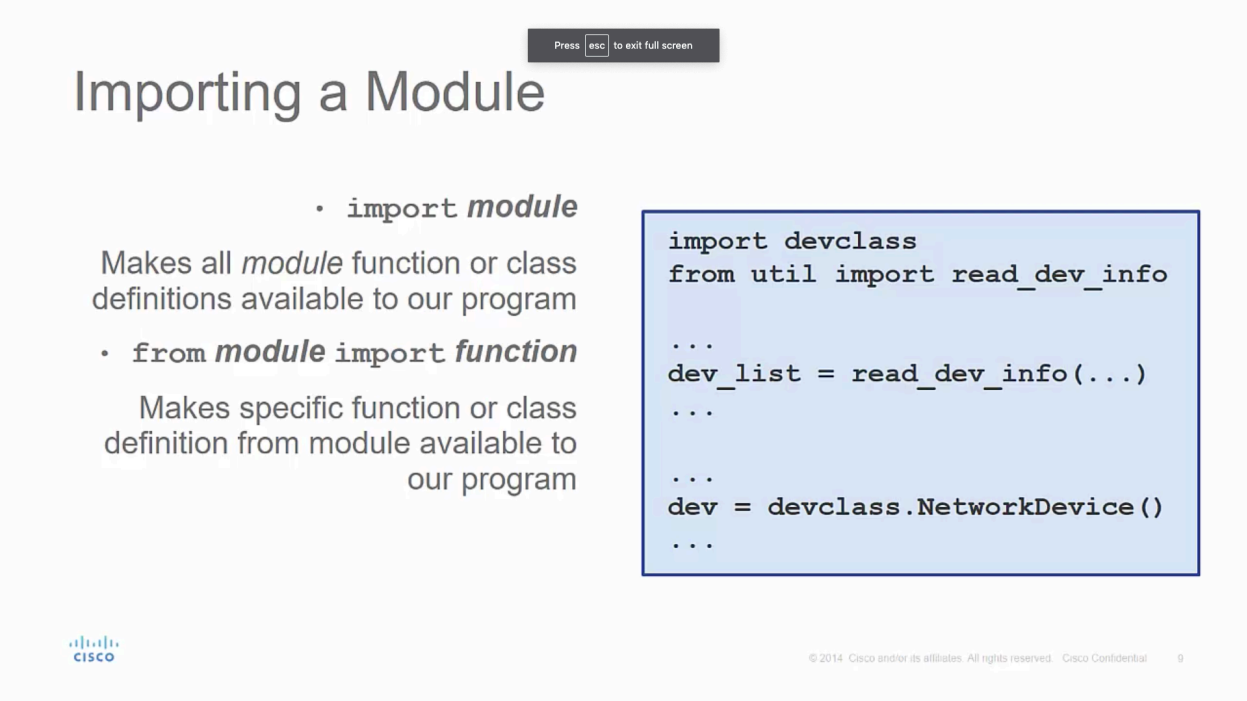

I can import the module and all its functions or specific functions from the module.

If I import the entire module, I need to refernce the module before calling the function.
e.g- devclass.NetworkDevice()

If I import the specific function from the module, I can just call the function.
e.g – read_dev_info(…)

Your application must inform Python about the module you intend to use, in order to actually call a function, or reference a class within that module.

Consider the following sample code:

```
import devclass
from util import read_dev_info
...
   dev_list = read_dev_info('devices')
   ...
   for device in devices_list:
      dev = devclass.NetworkDevice(...)
   ...
```

The example above shows multiple ways to import a module:

- `import module`: In the simplest format, you only need to include an `import` statement, informing Python that you intend to reference functions or classes from that module within your code. Using the `import` statement makes all functions or classes within the module available to be called.

  When you import just using the name of the module, you must precede the function or class name with the name of the module in which it resides. In the example above, the code specifies only `import devclass`, which means that functions or classes that are referenced from that module must be preceded by the name of the module.

  Importing the entire module may be a convenient method of allowing access to functions or classes within a module. However, consider that Python must make all functions or classes available, which may be inefficient. If the module has many functions or classes, it may be more efficient to use the next method.

- `from module import name`: In this more precise format, you specify both the module, and the actual function within the module you wish to reference. If your code needs to reference more than one function within a module, you will need to specify each of them, in order for Python to know your intended usage.
- `import module as name`: If for some reason your code already contains a variable or function with the same name as the function you intend to call from another module, you can change the local name by which you will refer to your module name.