```
Here you finally learn how to write realistic whole programs in Python.

You’ll write your own modules and learn how to use others from Python’s
standard library and other sources.
```

Modules and the import Statement

```
A module is just a file of any Python code.

You don’t need to do anything special—any Python code can be used as
a module by others.

We refer to code of other modules by using the Python import statement.

This makes the code and variables in the imported module available to your program.
```

Import a Module

```
The simplest use of the import statement is import module, where module
is the name of another Python file, without the .py extension.
```

```
%%writefile fast.py
from random import choice
places = ["McDonalds", "KFC", "Burger King", "Taco Bell","Wendys", "Arbys", "Pizza Hut"]
def pick():
  return choice(places)
```

```
    Overwriting fast.py
```

```
#!python fast.py
```

!CAUTION WHILE WRITING THE .py file

1. At first write the python code
2. Then add the statement **%%writefile fast.py** at the beginning of the code

```
import fast
place = fast.pick()
print("Let's go to", place)
```

```
    Let's go to Taco Bell
```

```
%%writefile lunch.py
import fast
place = fast.pick()
print("Let's go to", place)
```

```
    Writing lunch.py
```

```
!python lunch.py
```

```
    Let's go to Taco Bell
```

We could have written fast.py, as shown below, importing random within the pick() function instead of at the top of the file.

```
%%writefile fast2.py
places = ["McDonalds", "KFC", "Burger King", "Taco Bell", "Wendys", "Arbys", "Pizza Hut"]
def pick():
  import random
  return random.choice(places)
```

```
Writing fast2.py
```

```
%%writefile lunch2.py
import fast2
place = fast2.pick()
print("Let's go to", place)
```

```
Writing lunch2.py
```

```
!python lunch2.py
```

```
Let's go to McDonalds
```

Import a Module with Another Name

```
%%writefile fast3.py
import fast2 as f
place = f.pick()
print("Let's go to", place)
```

```
Writing fast3.py
```

```
!python fast3.py
```

```
Let's go to Burger King
```

Import Only What You Want from a Module

```
%%writefile fast4.py
from fast2 import pick
place = pick()
print("Let's go to", place)
```

```
Writing fast4.py
```

```
!python fast4.py
```

```
Let's go to Burger King
```

Another example

```
%%writefile ap/ap1.py
def myname():
   print('APURBA')

def mySubject():
   print("Machine Learning")
```

```
Overwriting ap/ap1.py
```

```
%%writefile ap/ap2.py
def myCollege():
   print('JISCE')
```

```
Writing ap/ap2.py
```

```
%%writefile apurba.py
from ap import ap1,ap2
ap1.myname()
ap1.mySubject()
ap2.myCollege()
```

```
Overwriting apurba.py
```

```
!python apurba.py
```

```
APURBA
Machine Learning
JISCE
```

Packages

```
A package is just a subdirectory that contains .py files.

And you can go more than one level deep, with directories inside those.

We just wrote a module that chooses a fast-food place.

Let’s add a similar module to dispense life advice.

We’ll make one new main program called questions.py in our current directory.

Now make a subdirectory named choices and put two modules in it —fast.py and advice.py.

Each module has a function that returns a string.

The main program (questions.py) has an extra import and line.
```

```
%%writefile choices/fast.py
from random import choice
places = ["McDonalds", "KFC", "Burger King", "Taco Bell","Wendys", "Arbys", "Pizza Hut"]
def pick():
   """Return random fast food place"""
   return choice(places)
```

```
Writing choices/fast.py
```

```
%%writefile choices/advice.py
from random import choice
answers = ["Yes!", "No!", "Reply hazy", "Sorry, what?"]
def give():
   """Return random advice"""
   return choice(answers)
```

```
Writing choices/advice.py
```

```
%%writefile questions.py
from choices import fast, advice
print("Let's go to", fast.pick())
print("Should we take out?", advice.give())
```

```
Overwriting questions.py
```

```
!python questions.py
```

```
Let's go to McDonalds
Should we take out? No!
```

The Module Search Path

```
To see all the places that your Python interpreter looks, import the standard sys module and use its path list.

This is a list of directory names and ZIP archive files that Python searches in order to find modules to import.
```

```
import sys
for place in sys.path:
  print(place)
```

```
/content
/env/python
/usr/lib/python39.zip
/usr/lib/python3.9
/usr/lib/python3.9/lib-dynload

/usr/local/lib/python3.9/dist-packages
/usr/lib/python3/dist-packages
/usr/local/lib/python3.9/dist-packages/IPython/extensions
/root/.ipython
```

You can modify the search path within your code.

Let’s say you want Python to look in the /choices/apu directory before any other:

```
import sys
sys.path.insert(0, "/choices/apu")
```

```
dir(print())
```

```
['__bool__',
 '__class__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__gt__',
 '__hash__',
 '__init__',
 '__init_subclass__',
 '__le__',
 '__lt__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__setattr__',
 '__sizeof__',
 '__str__',
 '__subclasshook__']
```