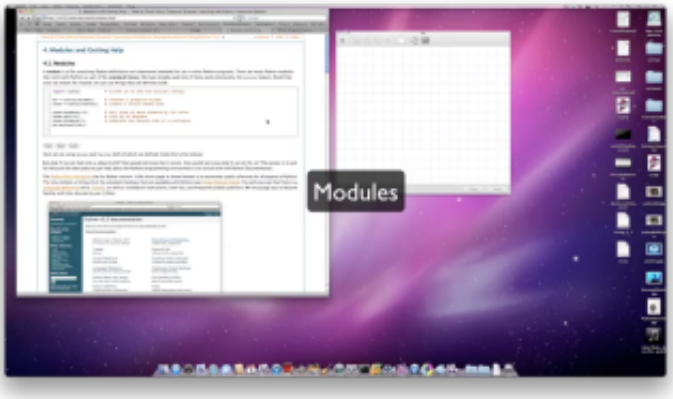
Modules and Getting Help

Modules

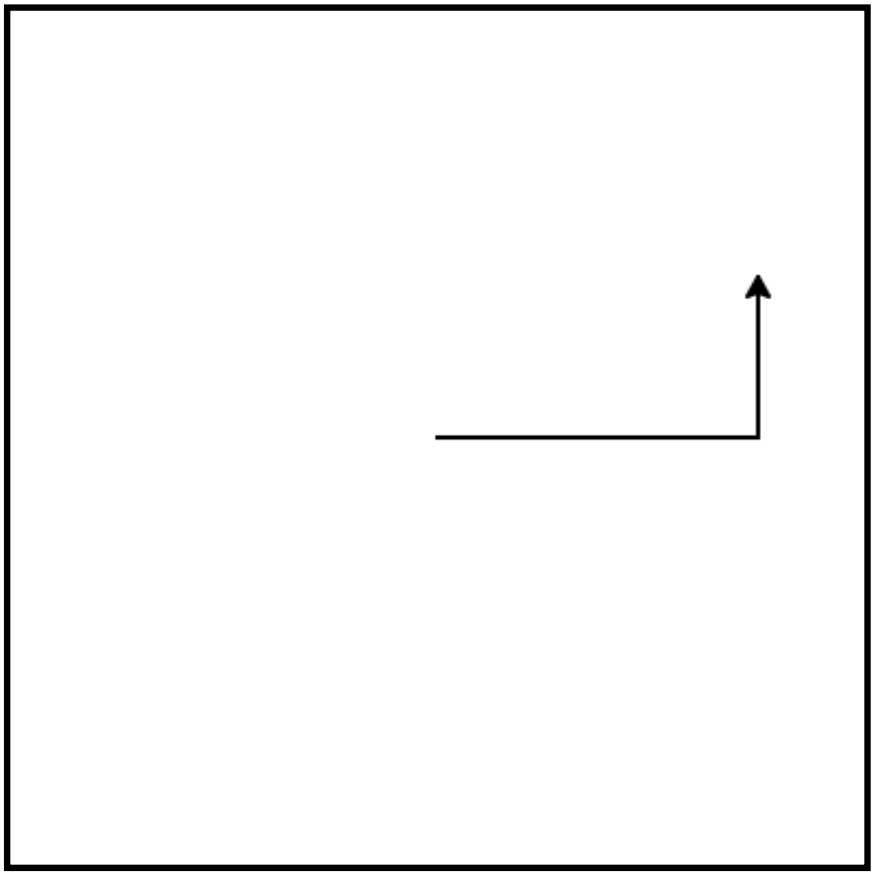


A **module** is a file containing Python definitions and statements intended for use in other Python programs. There are many Python modules that come with Python as part of the **standard library**. We have already used one of these quite extensively, the `turtle` module. Recall that once we import the module, we can use things that are defined inside.

```
1  import turtle            # allows us to use the turtles library
2
3  wn = turtle.Screen()     # creates a graphics window
4  alex = turtle.Turtle()   # create a turtle named alex
5
6  alex.forward(150)        # tell alex to move forward by 150 units
7  alex.left(90)            # turn by 90 degrees
8  alex.forward(75)         # complete the second side of a rectangle
9  wn.exitonclick()
10
```

ActiveCode: 1 (chmod_01)

Run

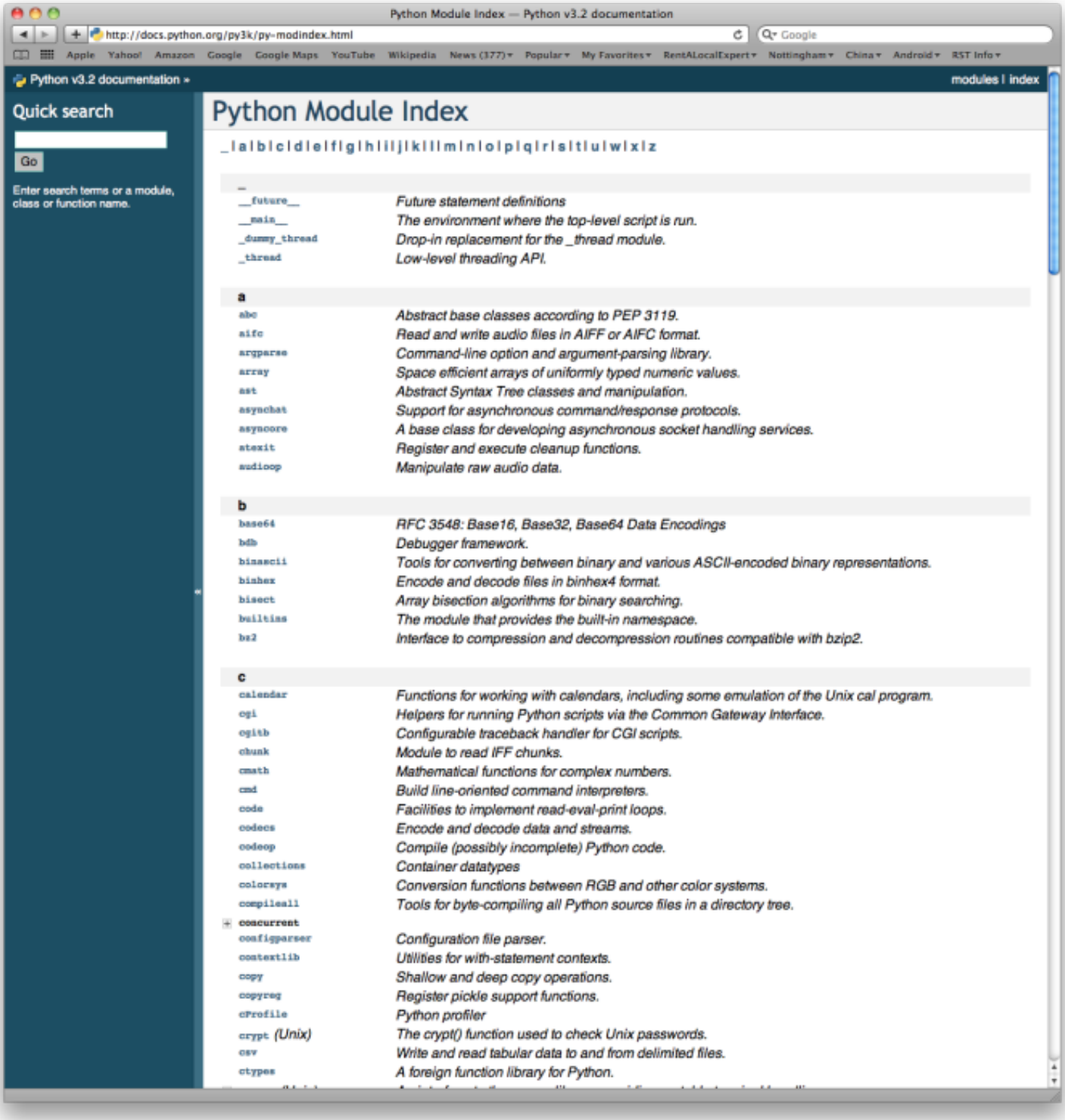


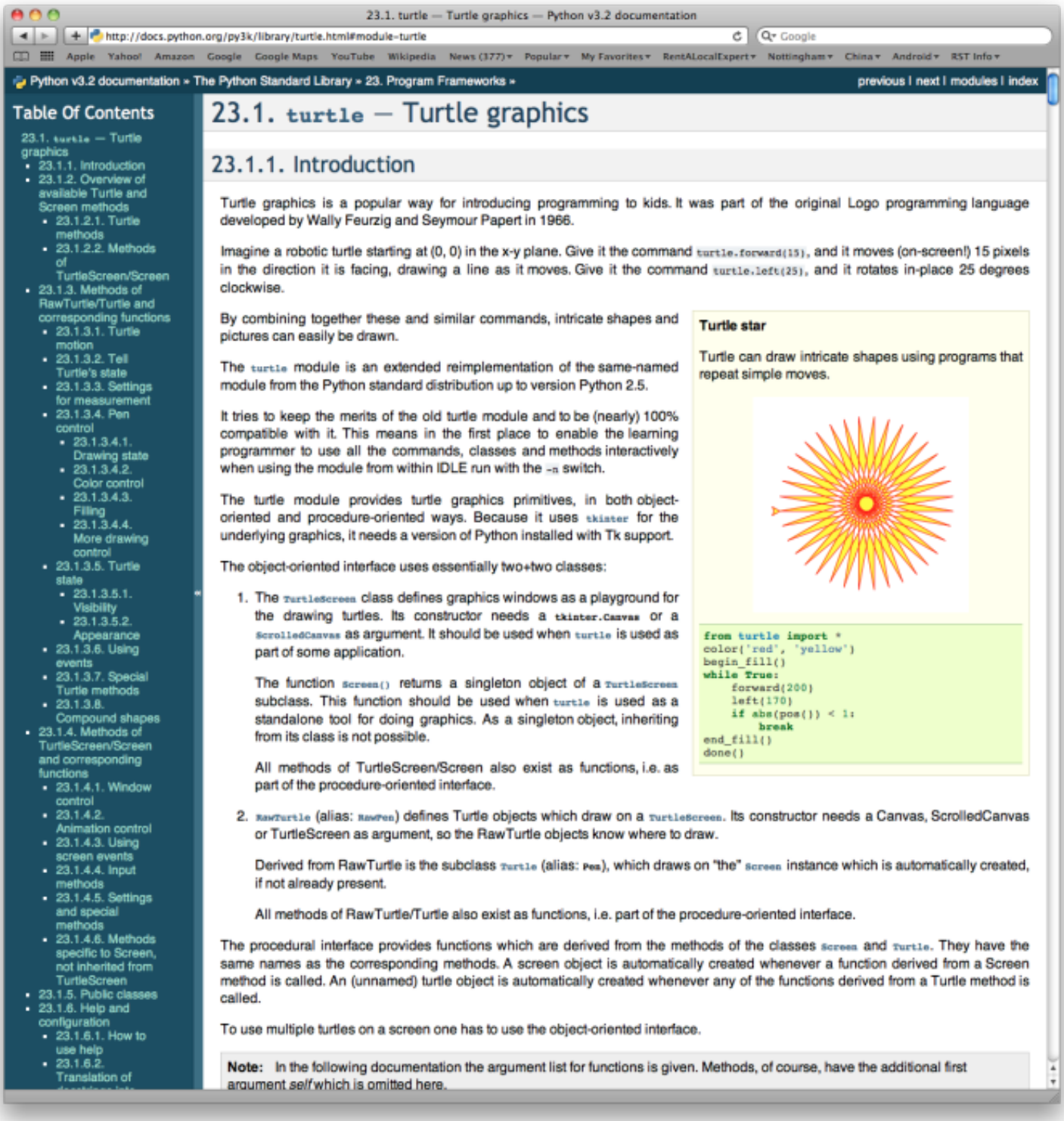
Here we are using `Screen` and `Turtle`, both of which are defined inside the turtle module.

But what if no one had told us about turtle? How would we know that it exists. How would we know what it can do for us? The answer is to ask for help and the best place to get help about the Python programming environment is to consult with the Python Documentation.

The Python Documentation (http://docs.python.org/py3k/) site for Python version 3 (the home page is shown below) is an extremely useful reference for all aspects of Python. The site contains a listing of all the standard modules that are available with Python (see Global Module Index (http://docs.python.org/py3k/py-modindex.html)). You will also see that there is a Language Reference (http://docs.python.org/py3k/reference/index.html) and a Tutorial (http://docs.python.org/py3k/tutorial/index.html), as well as installation instructions, how-tos, and frequently asked questions. We encourage you to become familiar with this site and to use it often.

If you have not done so already, take a look at the Global Module Index. Here you will see an alphabetical listing of all the modules that are available as part of the standard library. Find the turtle module.


Python Module Index — Python v3.2 documentation
http://docs.python.org/py3k/py-modindex.html
Google
Apple Yahoo! Amazon Google Google Maps YouTube Wikipedia News (377) Popular My Favorites RentALocalExpert Nottingham China Android RSS Info
Python v3.2 documentation »
modules | index
Quick search
Go
Enter search terms or a module, class or function name.
Python Module Index
_ | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | w | x | z
_
__future__ Future statement definitions
__main__ The environment where the top-level script is run.
_dummy_thread Drop-in replacement for the _thread module.
_thread Low-level threading API.
a
abc Abstract base classes according to PEP 3119.
aifc Read and write audio files in AIFF or AIFC format.
argparse Command-line option and argument-parsing library.
array Space efficient arrays of uniformly typed numeric values.
ast Abstract Syntax Tree classes and manipulation.
asynchat Support for asynchronous command/response protocols.
asyncore A base class for developing asynchronous socket handling services.
atexit Register and execute cleanup functions.
audioop Manipulate raw audio data.
b
base64 RFC 3548: Base16, Base32, Base64 Data Encodings
bdb Debugger framework.
binascii Tools for converting between binary and various ASCII-encoded binary representations.
binhex Encode and decode files in binhex4 format.
bisect Array bisection algorithms for binary searching.
builtins The module that provides the built-in namespace.
bz2 Interface to compression and decompression routines compatible with bzip2.
c
calendar Functions for working with calendars, including some emulation of the Unix cal program.
cgi Helpers for running Python scripts via the Common Gateway Interface.
cgitb Configurable traceback handler for CGI scripts.
chunk Module to read IFF chunks.
cmath Mathematical functions for complex numbers.
cmd Build line-oriented command interpreters.
code Facilities to implement read-eval-print loops.
codecs Encode and decode data and streams.
codeop Compile (possibly incomplete) Python code.
collections Container datatypes
colorsys Conversion functions between RGB and other color systems.
compileall Tools for byte-compiling all Python source files in a directory tree.
concurrent
configparser Configuration file parser.
contextlib Utilities for with-statement contexts.
copy Shallow and deep copy operations.
copyreg Register pickle support functions.
cProfile Python profiler
crypt (Unix) The crypt() function used to check Unix passwords.
csv Write and read tabular data to and from delimited files.
ctypes A foreign function library for Python.



You can see that all the turtle functionality that we have talked about is there. However, there is so much more. Take some time to read through and familiarize yourself with some of the other things that turtles can do.

Note: Python modules and limitations with activecode

Throughout these lectures, activecode windows allow you to practice the Python that you are learning. We mentioned in the first chapter that programming is normally done using some type of development environment and that the activecode used here was strictly to help us learn. It is not the way we write production programs.

To that end, it is necessary to mention that many of the modules available in standard Python will **not** work in the activecode environment. In fact, only turtle, math, and random have been ported at this point. If you wish to explore any additional modules, you will need to also explore using a more robust development environment.

Check your understanding

mod-1: In Python a module is:

- (•) a) A file containing Python definitions and statements intended for use in other Python programs.
- ( ) b) A separate block of code within a program.
- ( ) c) One line of code in a program.
- ( ) d) A file that contains documentation about functions in Python.

Check Me | Compare Me

Correct!! A module can be reused in different programs.

mod-2: To find out information on the standard modules available with Python you should:

- (•) a) Go to the Python Documentation site.
- ( ) b) Look at the import statements of the program you are working with or writing.
- ( ) c) Ask the professor
- ( ) d) Look in this lecture.

Check Me | Compare Me

Correct!! The site contains a listing of all the standard modules that are available with Python.

mod-3: True / False: All standard Python modules will work in activecode.

- a) True
- b) False

Check Me | Compare Me

Correct!! Only turtle, math, and random have been ported to work in activecode at this time.