

An Introduction To Atom

Activity for the 2015 EarthScope Data Processing Workshop

Charles J. Ammon, Penn State

Follow along with this activity on your own.

Introduction

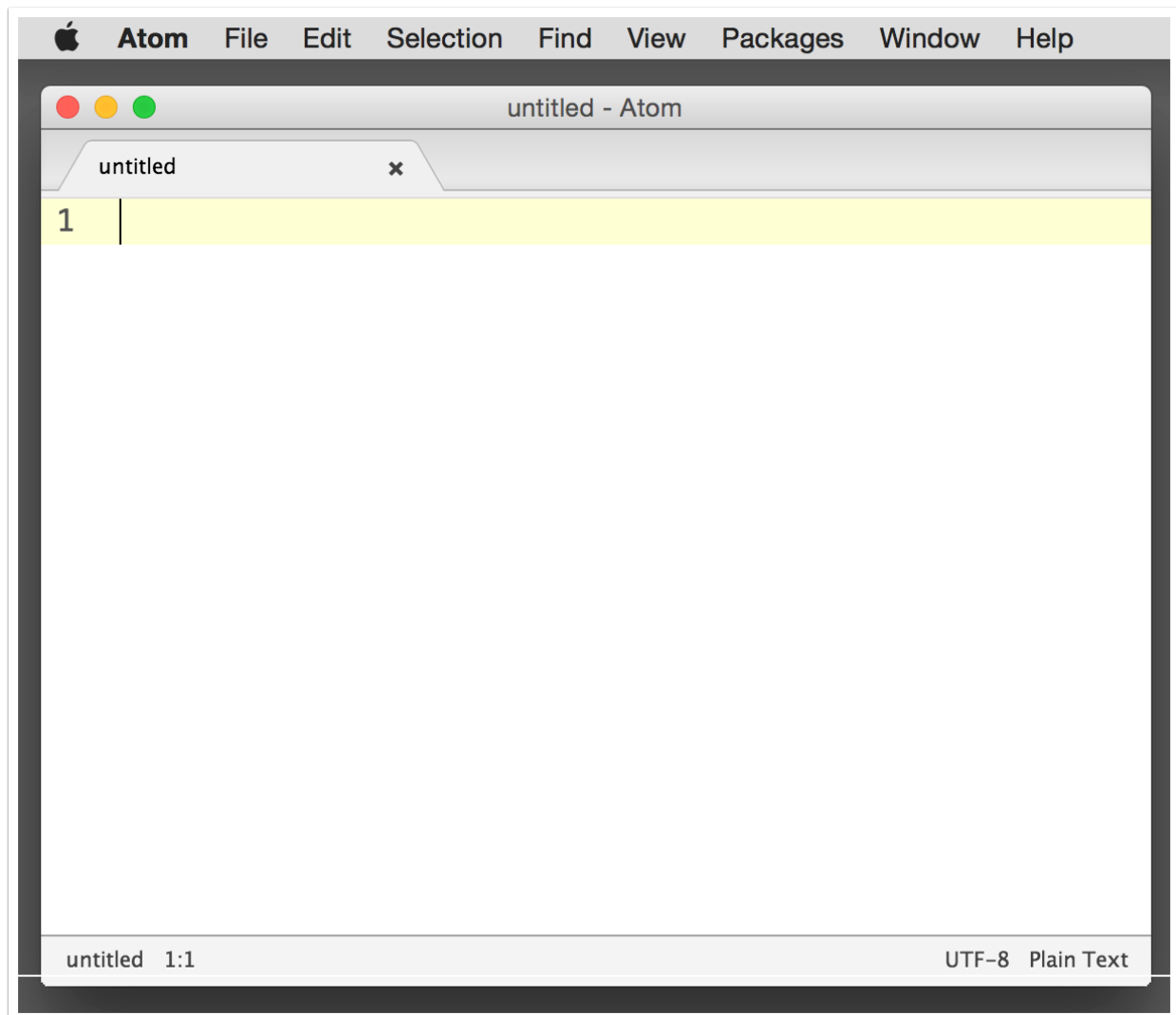
You absolutely should use a programming editor. The specific editor you choose may depend on the language or program that you use (for example, Matlab has it's own built in editor). I recommend that you learn both a GUI-based editor similar to the one we explore below, and a command-line editor such as vi, vim, nano, or emacs that you can use when you only have terminal access to a particular system.

We are going to use the Atom editor in this activity (and again later). The Atom editor is essentially a clone of a very good commercial editor, Sublime Text. Many of the features of Sublime are implemented in Atom, which is developed by the github group. Atom is a powerful customizable editor written in CoffeeScript, a flavor of javascript (so it runs on just about every platform).

When you launch Atom, you are presented with an empty "buffer" that you can fill by typing, or pasting in text that you may have copied from elsewhere (like a browser). We don't have time to go into many of Atom's features, the goal here is to show you a few items that can help motivate you to become familiar with Atom or a similarly powerful editor.

Atom implements the standard cut-copy-paste commands and also uses the standard open-file, save-file interfaces. So the basic editing activities should be familiar.

An Introduction To Atom

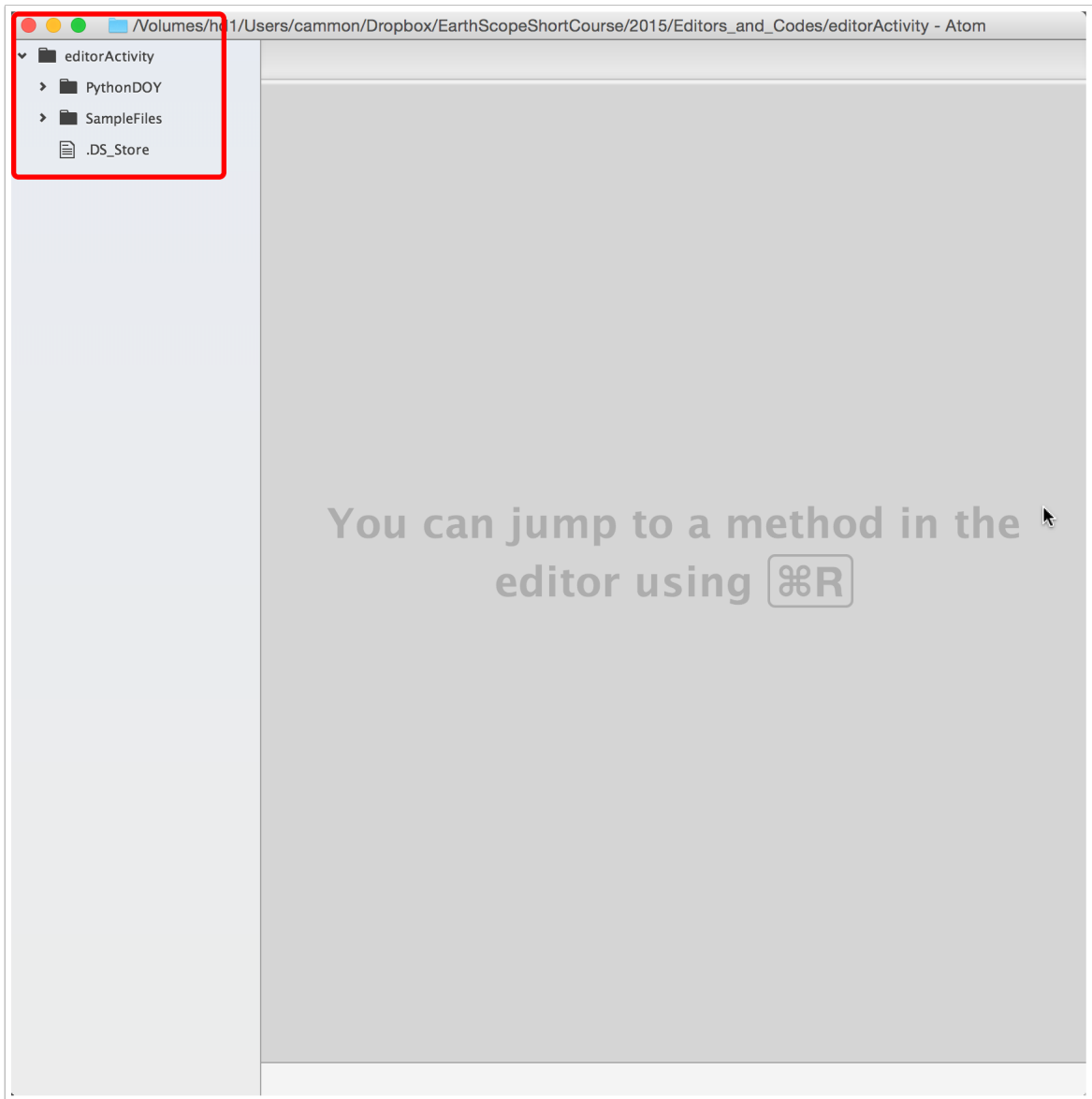


Opening Folders

A very important feature of any powerful editor is that they don't just open files, they let you work with folders - so you can manage all the files in a project at once. To open one of the files, you simply single-click it in the navigation bar to the left. The gray text

An Introduction To Atom

in the editor frame is a feature of Atom - it provides users with hints while the editor is idling.



An Introduction To Atom

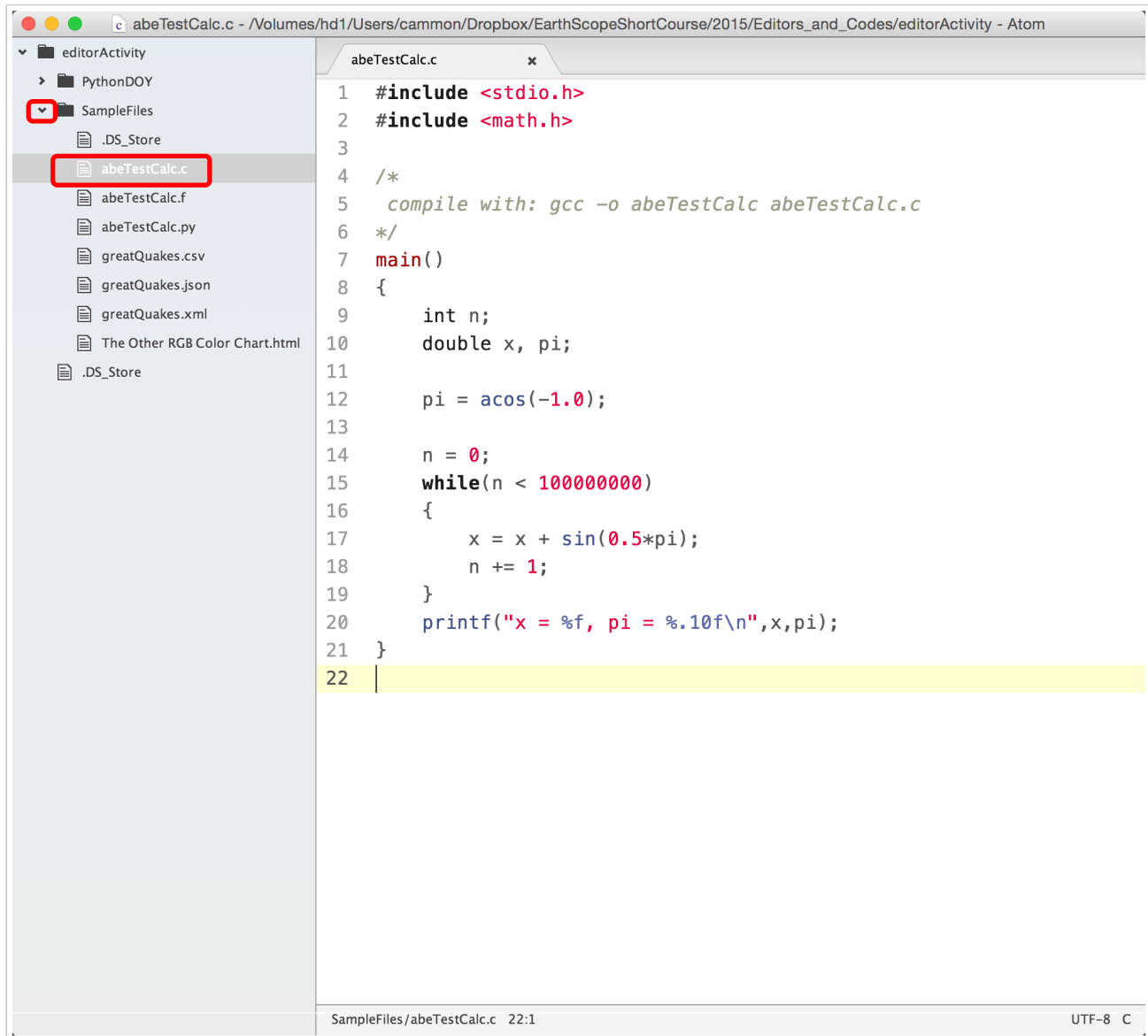
Syntax Highlighting

An essential feature for programmers is syntax highlighting. Atom knows about many common programming languages. Even if your language of choice is not one of the defaults included with Atom, you may be able to find a package that implements the highlighting for that language.

Open the SampleFiles folder in the activity files folder that you downloaded earlier. Then explore the syntax highlighting for the different languages.

Here's an example of syntax highlighting in C:

An Introduction To Atom

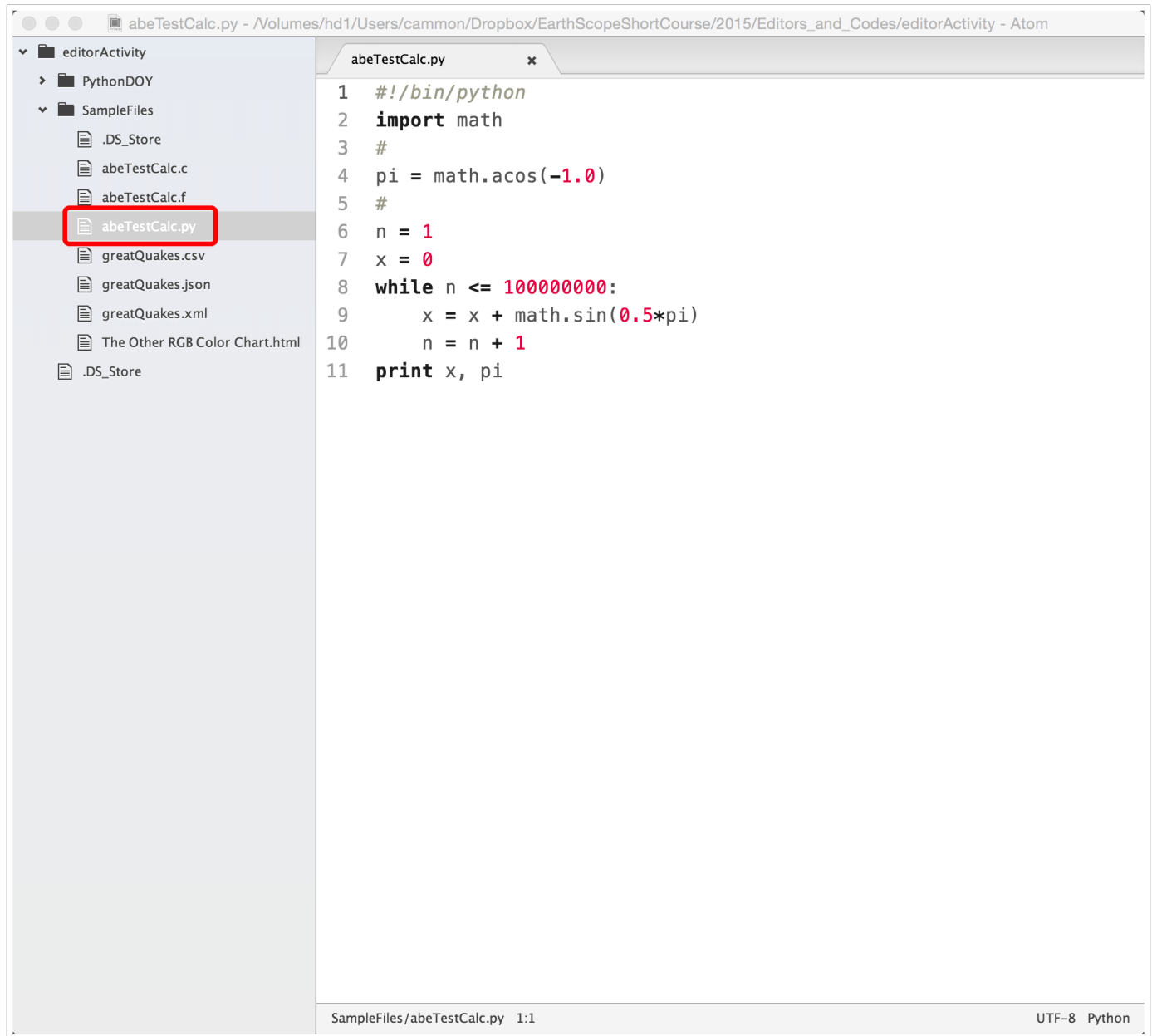


The screenshot shows the Atom text editor interface. On the left, a file explorer sidebar displays a project structure with folders 'editorActivity' and 'PythonDOY', and a file 'SampleFiles'. The 'SampleFiles' folder is expanded, showing a list of files including '.DS_Store', 'abeTestCalc.c', 'abeTestCalc.f', 'abeTestCalc.py', 'greatQuakes.csv', 'greatQuakes.json', 'greatQuakes.xml', 'The Other RGB Color Chart.html', and another '.DS_Store'. The file 'abeTestCalc.c' is selected and highlighted with a red rectangle. The main editor area displays the contents of 'abeTestCalc.c', which is a C program. The code includes headers for `<stdio.h>` and `<math.h>`, and contains a `main()` function. Inside `main()`, it declares `int n;` and `double x, pi;`, initializes `pi = acos(-1.0);`, sets `n = 0;`, and enters a `while(n < 100000000)` loop. The loop body calculates `x = x + sin(0.5*pi);` and increments `n += 1;`. After the loop, it prints the values of `x` and `pi` using `printf("x = %f, pi = %.10f\n", x, pi);`. The status bar at the bottom indicates the file path 'SampleFiles/abeTestCalc.c' and the current cursor position '22:1', along with the encoding 'UTF-8' and the character set 'C'.

```
1 #include <stdio.h>
2 #include <math.h>
3
4 /*
5  compile with: gcc -o abeTestCalc abeTestCalc.c
6 */
7 main()
8 {
9     int n;
10    double x, pi;
11
12    pi = acos(-1.0);
13
14    n = 0;
15    while(n < 100000000)
16    {
17        x = x + sin(0.5*pi);
18        n += 1;
19    }
20    printf("x = %f, pi = %.10f\n", x, pi);
21 }
22
```

An Introduction To Atom

Here is a python file:



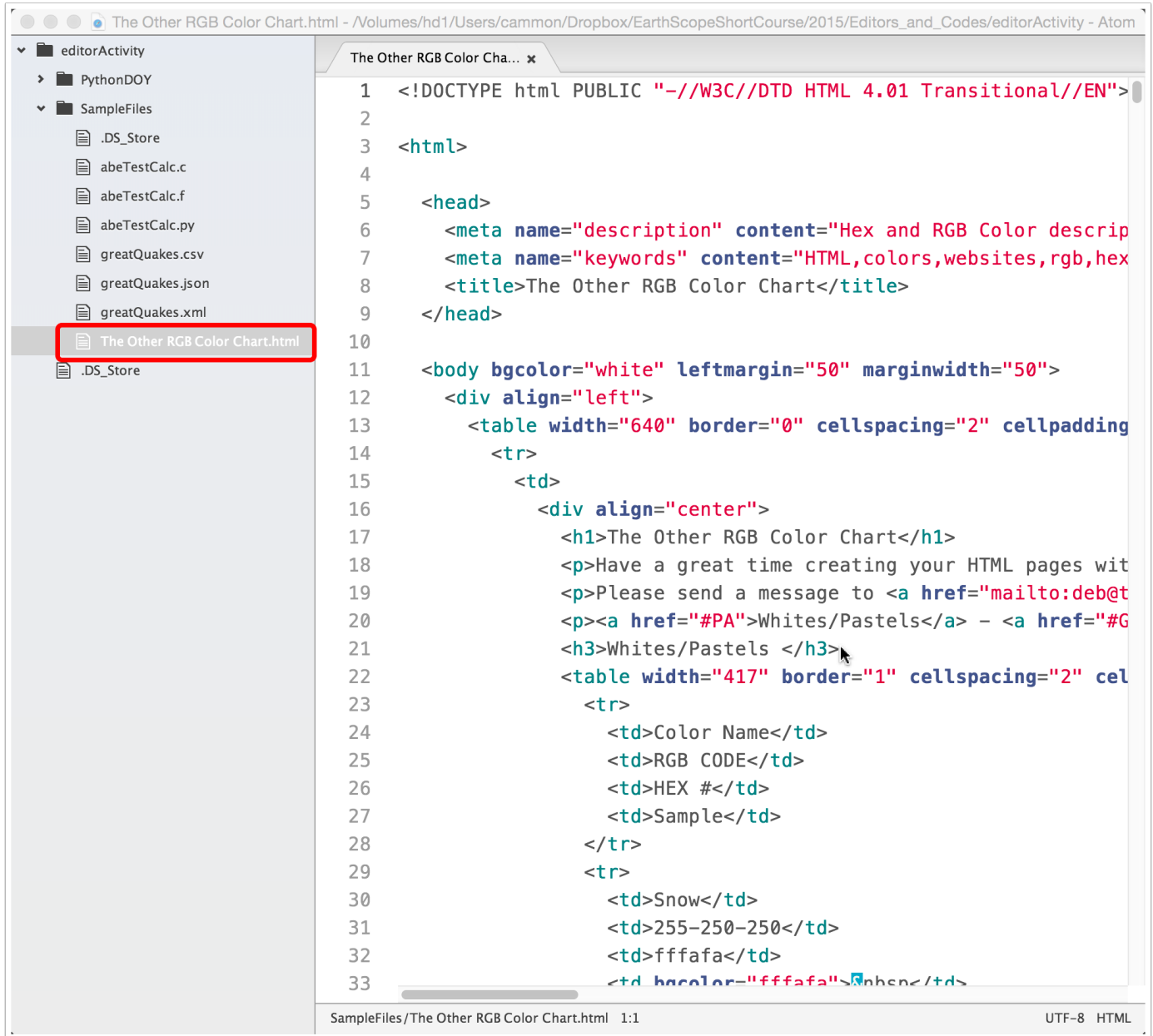
The screenshot shows the Atom text editor interface. On the left is a sidebar with a file explorer showing a project structure. The file 'abeTestCalc.py' is selected and highlighted with a red rectangle. The main editor pane on the right displays the contents of this file, which is a Python script. The script starts with a shebang line, imports the 'math' module, and calculates the value of pi using the arccos function. It then enters a while loop that iterates 100,000,000 times, adding the sine of half pi to a variable 'x' in each iteration. Finally, it prints the values of 'x' and 'pi'.

```
1  #!/bin/python
2  import math
3  #
4  pi = math.acos(-1.0)
5  #
6  n = 1
7  x = 0
8  while n <= 100000000:
9      x = x + math.sin(0.5*pi)
10     n = n + 1
11 print x, pi
```

At the bottom of the editor, a status bar shows the file path 'SampleFiles/abeTestCalc.py' and the line number '1:1'. On the far right, it indicates the encoding is 'UTF-8' and the language is 'Python'.

An Introduction To Atom

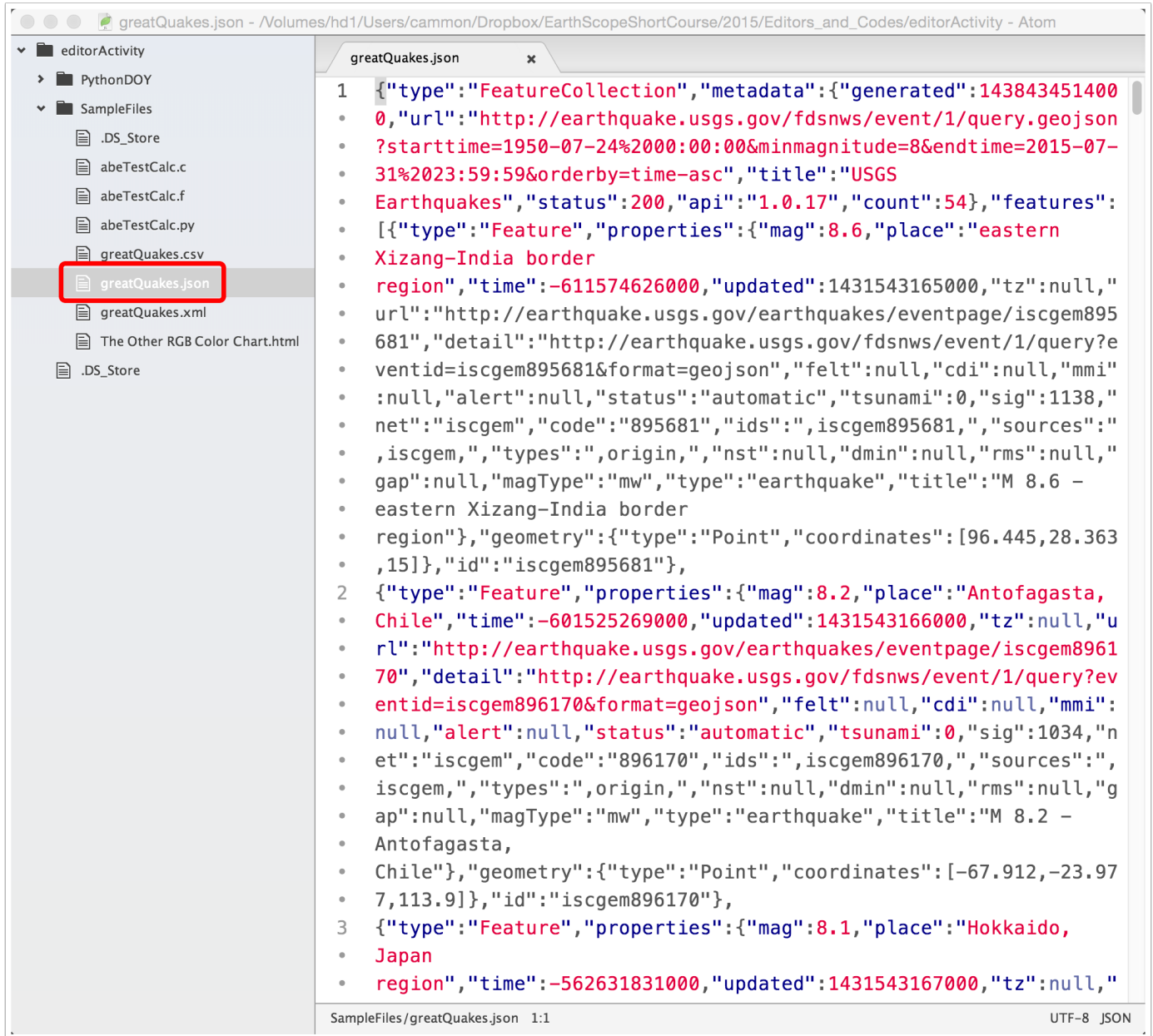
Here is an HTML file:



```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
2
3 <html>
4
5   <head>
6     <meta name="description" content="Hex and RGB Color descrip
7     <meta name="keywords" content="HTML,colors,websites,rgb,hex
8     <title>The Other RGB Color Chart</title>
9   </head>
10
11   <body bgcolor="white" leftmargin="50" marginwidth="50">
12     <div align="left">
13       <table width="640" border="0" cellpadding="2" cellspacing="2">
14         <tr>
15           <td>
16             <div align="center">
17               <h1>The Other RGB Color Chart</h1>
18               <p>Have a great time creating your HTML pages wit
19               <p>Please send a message to <a href="mailto:deb@t
20               <p><a href="#PA">Whites/Pastels</a> - <a href="#G
21               <h3>Whites/Pastels </h3>
22               <table width="417" border="1" cellpadding="2" cel
23                 <tr>
24                   <td>Color Name</td>
25                   <td>RGB CODE</td>
26                   <td>HEX #</td>
27                   <td>Sample</td>
28                 </tr>
29                 <tr>
30                   <td>Snow</td>
31                   <td>255-250-250</td>
32                   <td>fffafa</td>
33                   <td>Sample</td>
```

An Introduction To Atom

Here is a JSON file:



```
1 {"type": "FeatureCollection", "metadata": {"generated": 143843451400
  • 0, "url": "http://earthquake.usgs.gov/fdsnws/event/1/query.geojson
  • ?starttime=1950-07-24%2000:00:00&minmagnitude=8&endtime=2015-07-
  • 31%2023:59:59&orderby=time-asc", "title": "USGS
  • Earthquakes", "status": 200, "api": "1.0.17", "count": 54}, "features":
  • [{"type": "Feature", "properties": {"mag": 8.6, "place": "eastern
  • Xizang-India border
  • region", "time": -611574626000, "updated": 1431543165000, "tz": null, "
  • url": "http://earthquake.usgs.gov/earthquakes/eventpage/iscgem895
  • 681", "detail": "http://earthquake.usgs.gov/fdsnws/event/1/query?e
  • ventid=iscgem895681&format=geojson", "felt": null, "cdi": null, "mmi"
  • : null, "alert": null, "status": "automatic", "tsunami": 0, "sig": 1138, "
  • net": "iscgem", "code": "895681", "ids": "", "iscgem895681", "sources": "
  • , iscgem", "types": "", "origin": "", "nst": null, "dmin": null, "rms": null, "
  • gap": null, "magType": "mw", "type": "earthquake", "title": "M 8.6 -
  • eastern Xizang-India border
  • region"}, "geometry": {"type": "Point", "coordinates": [96.445, 28.363
  • , 15]}}, {"id": "iscgem895681"},
2 {"type": "Feature", "properties": {"mag": 8.2, "place": "Antofagasta,
  • Chile", "time": -601525269000, "updated": 1431543166000, "tz": null, "u
  • rl": "http://earthquake.usgs.gov/earthquakes/eventpage/iscgem8961
  • 70", "detail": "http://earthquake.usgs.gov/fdsnws/event/1/query?ev
  • entid=iscgem896170&format=geojson", "felt": null, "cdi": null, "mmi":
  • null, "alert": null, "status": "automatic", "tsunami": 0, "sig": 1034, "n
  • et": "iscgem", "code": "896170", "ids": "", "iscgem896170", "sources": "
  • , iscgem", "types": "", "origin": "", "nst": null, "dmin": null, "rms": null, "g
  • ap": null, "magType": "mw", "type": "earthquake", "title": "M 8.2 -
  • Antofagasta,
  • Chile"}, "geometry": {"type": "Point", "coordinates": [-67.912, -23.97
  • 7, 113.9]}}, {"id": "iscgem896170"},
3 {"type": "Feature", "properties": {"mag": 8.1, "place": "Hokkaido,
  • Japan
  • region", "time": -562631831000, "updated": 1431543167000, "tz": null, "
  • 8
```


An Introduction To Atom

Here is a Fortran File:

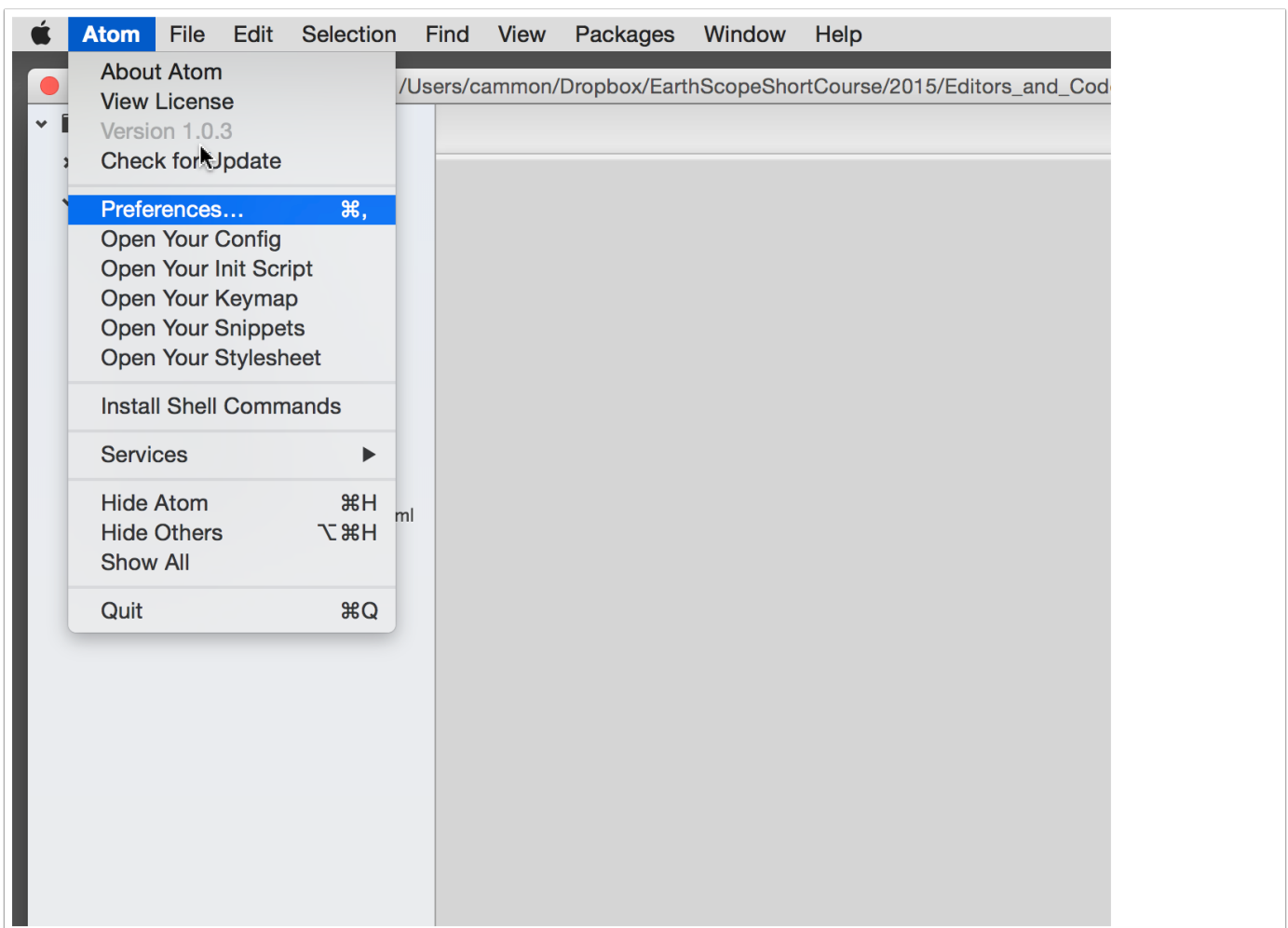
```
1  program abeTestCalc
2  c
3  c  compile with: gfortran -o abeTestCalc abeTestCalc.f
4  c  or
5  c  compile with gfortran -o abeTestCalc -O3 abeTestCalc.f
6  c
7  double precision x, pi
8  integer i
9
10 pi = acos(-1.0D0)
11 x = 0.0
12 do 1 i=1,100000000
13     x = x + dsin(0.5D0*pi)
14 1  continue
15
16 write(*,*) x, pi
17
18 stop
19 end
```

No Fortran highlights? That's awful.

An Introduction To Atom

Extending Atom by Installing Packages

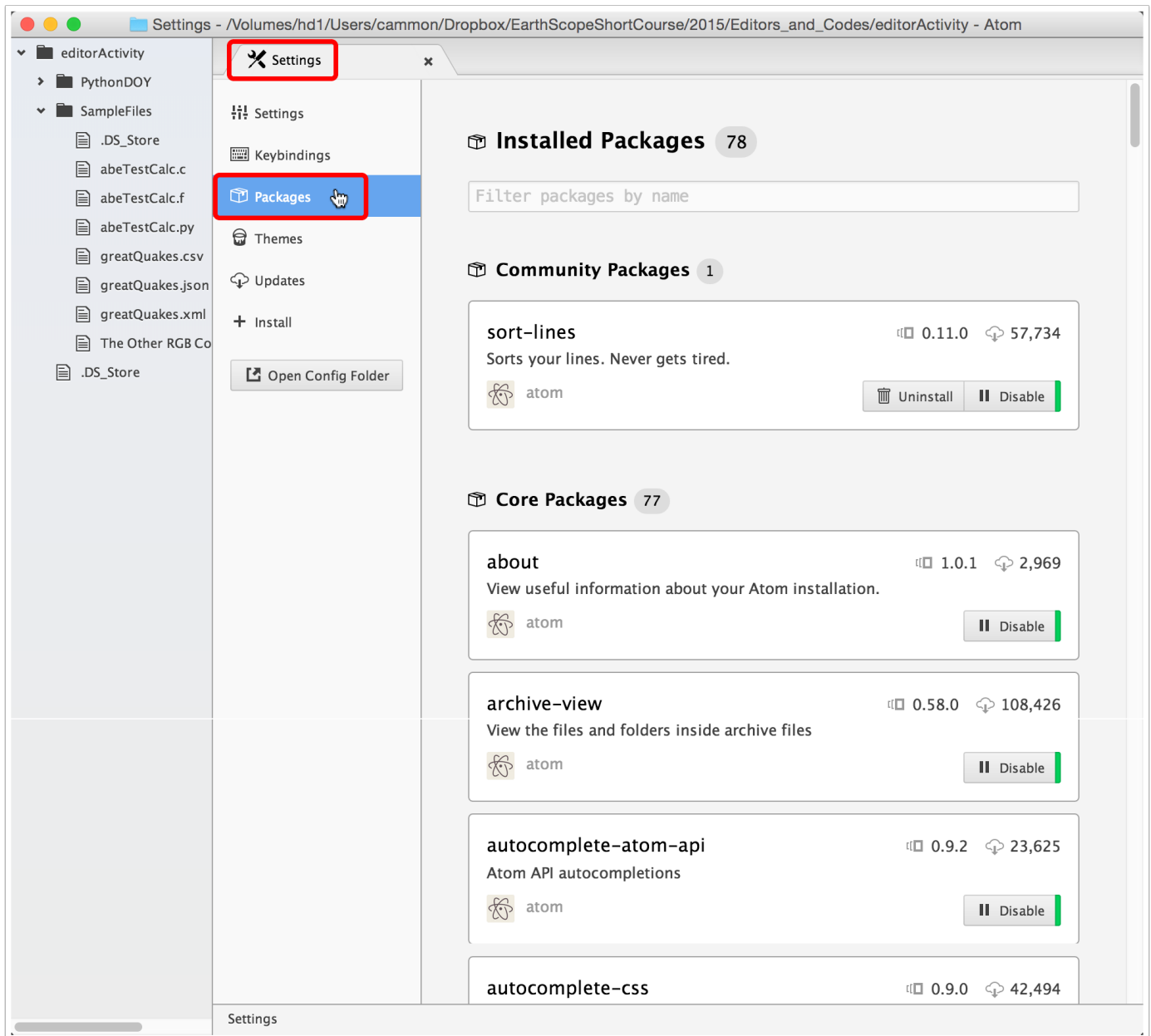
Most modern programming editors are extensible in that they allow users to write small packages or tools that extend the editor's capability. This is true of Atom. We are going to install a couple of simple Atom packages, the first allows Fortran syntax highlighting, the second formats the JSON files into a more readable style. Select preferences from the Atom menu in the upper left.



An Introduction To Atom

What packages are installed?

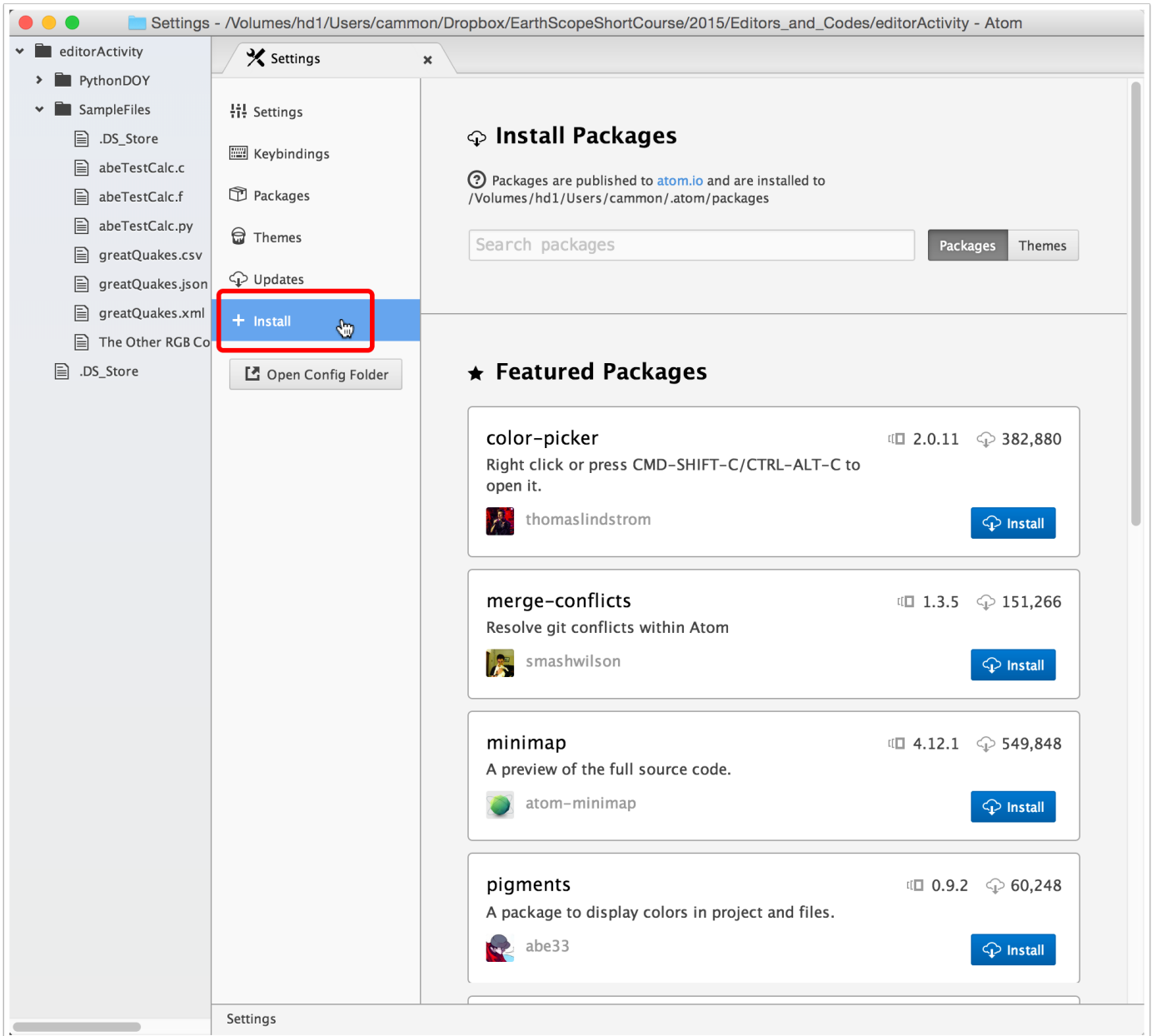
Select packages in the settings tab and "Packages". The pane on the right lists the packages that you have installed. Atom is really a small program that includes a substantial number of packages by default.



An Introduction To Atom

Adding additional packages

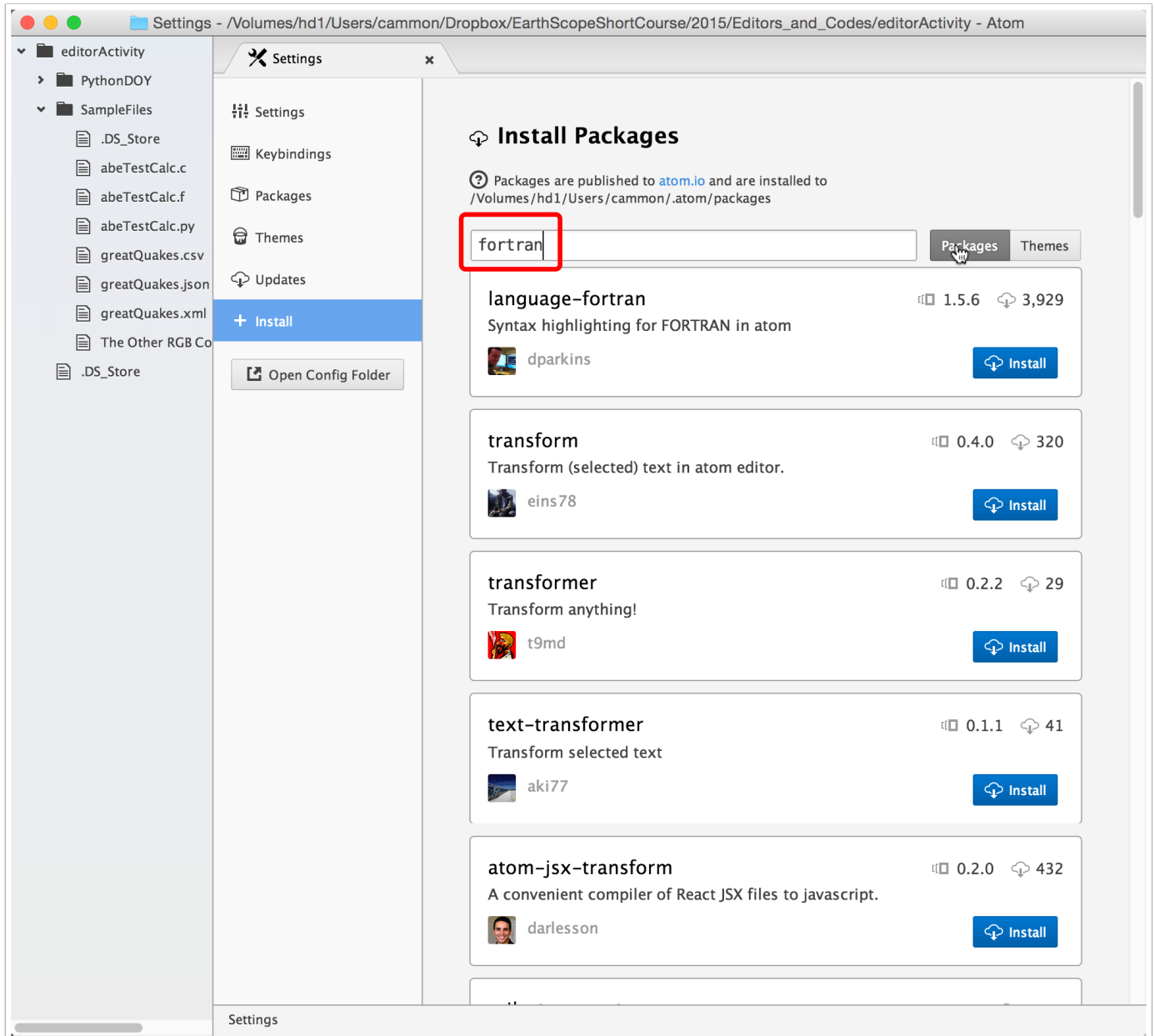
We want to add a package that includes syntax-highlighting for Fortran codes. Click on the + Install button in the settings control pane.



An Introduction To Atom

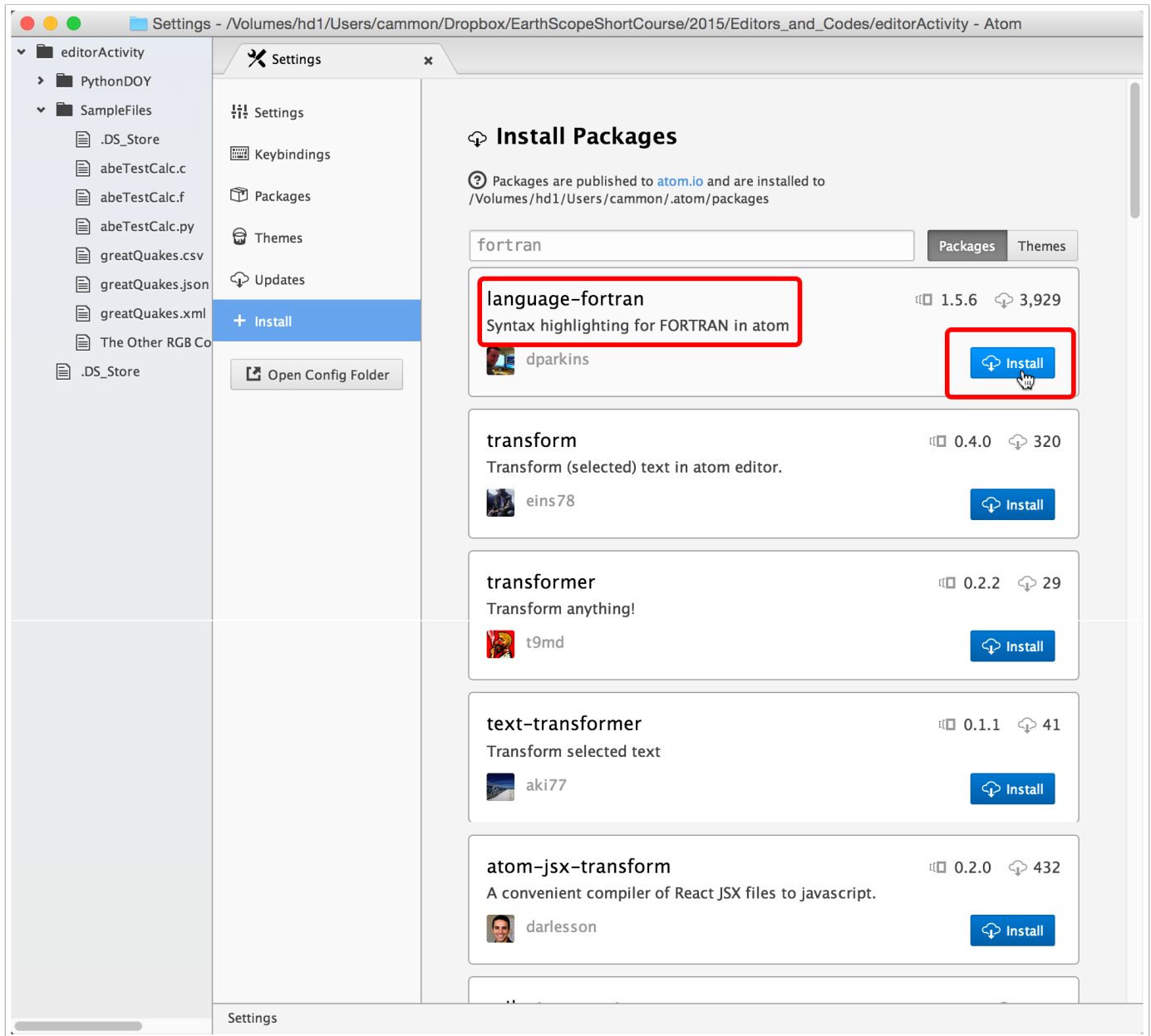
Search for the package you want.

Enter the word "fortran" in the search field and click the packages button.



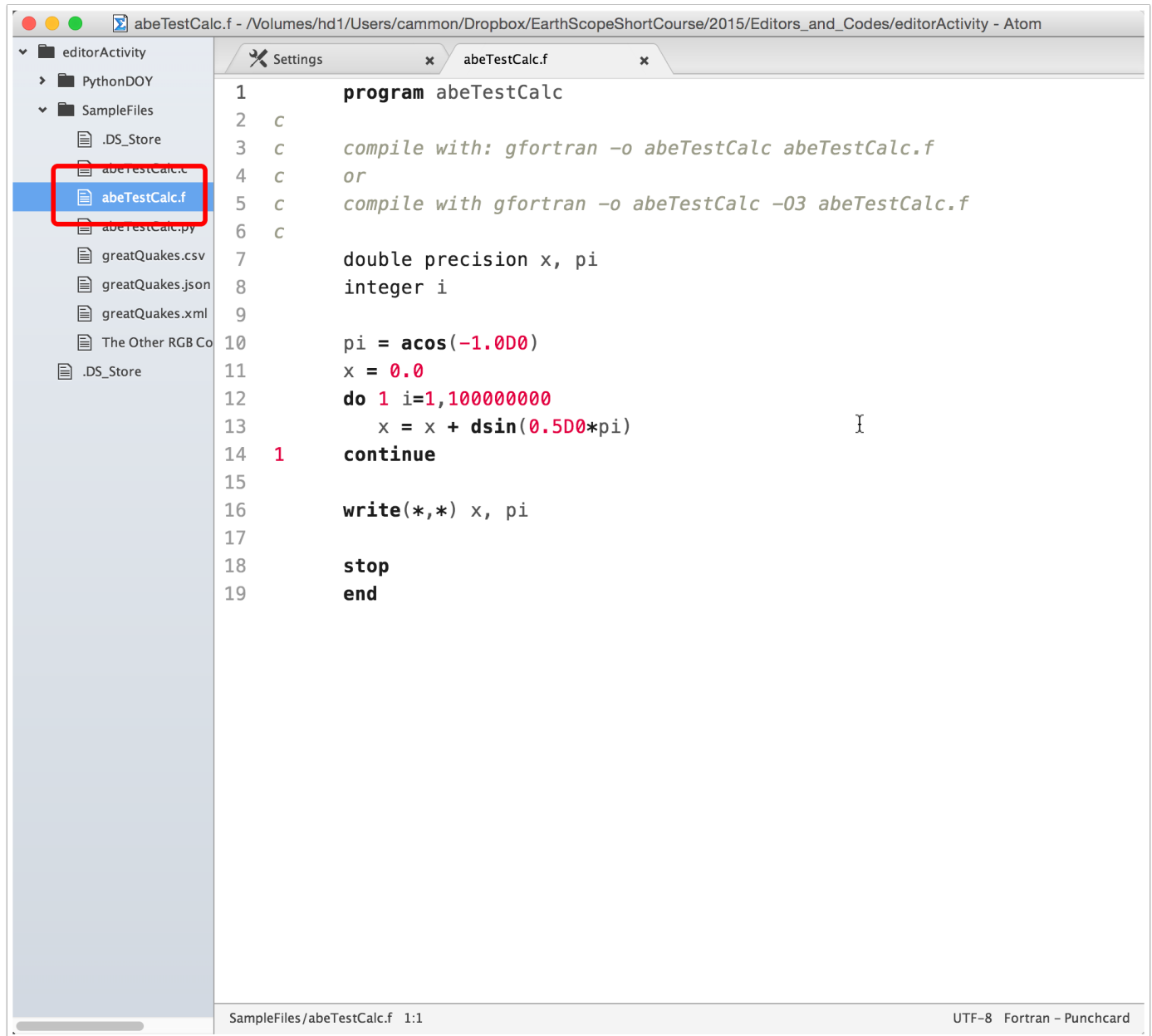
An Introduction To Atom

Install the language-fortran package by single-clicking the install button.



An Introduction To Atom

Now examine the fortan file again.



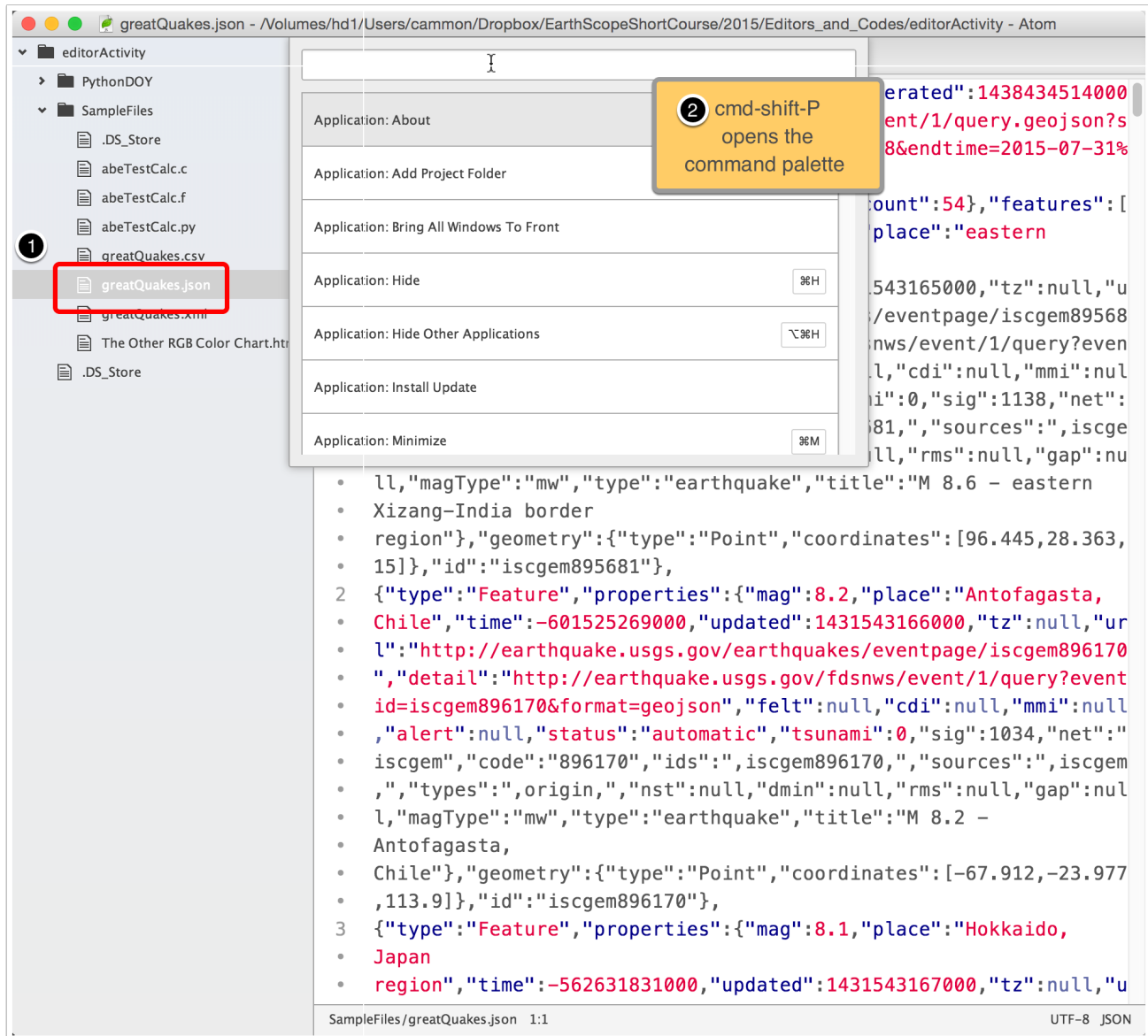
```
1  program abeTestCalc
2  c
3  c   compile with: gfortran -o abeTestCalc abeTestCalc.f
4  c   or
5  c   compile with gfortran -o abeTestCalc -O3 abeTestCalc.f
6  c
7
8  double precision x, pi
9  integer i
10
11  pi = acos(-1.0D0)
12  x = 0.0
13  do 1 i=1,100000000
14      x = x + dsin(0.5D0*pi)
15  continue
16
17  write(*,*) x, pi
18
19  stop
20  end
```

An Introduction To Atom

Using Atom's Packages

To invoke various packages in Atom, you use the do-everything keyboard shortcut **Shift-Command-P**. Let's invoke a package that reformats JSON files into a more readable format. First, close the tabs that you have open then single-click on the `greatQuakes.json` item in the file list. Then invoke the command selector using Shift-Command-P.

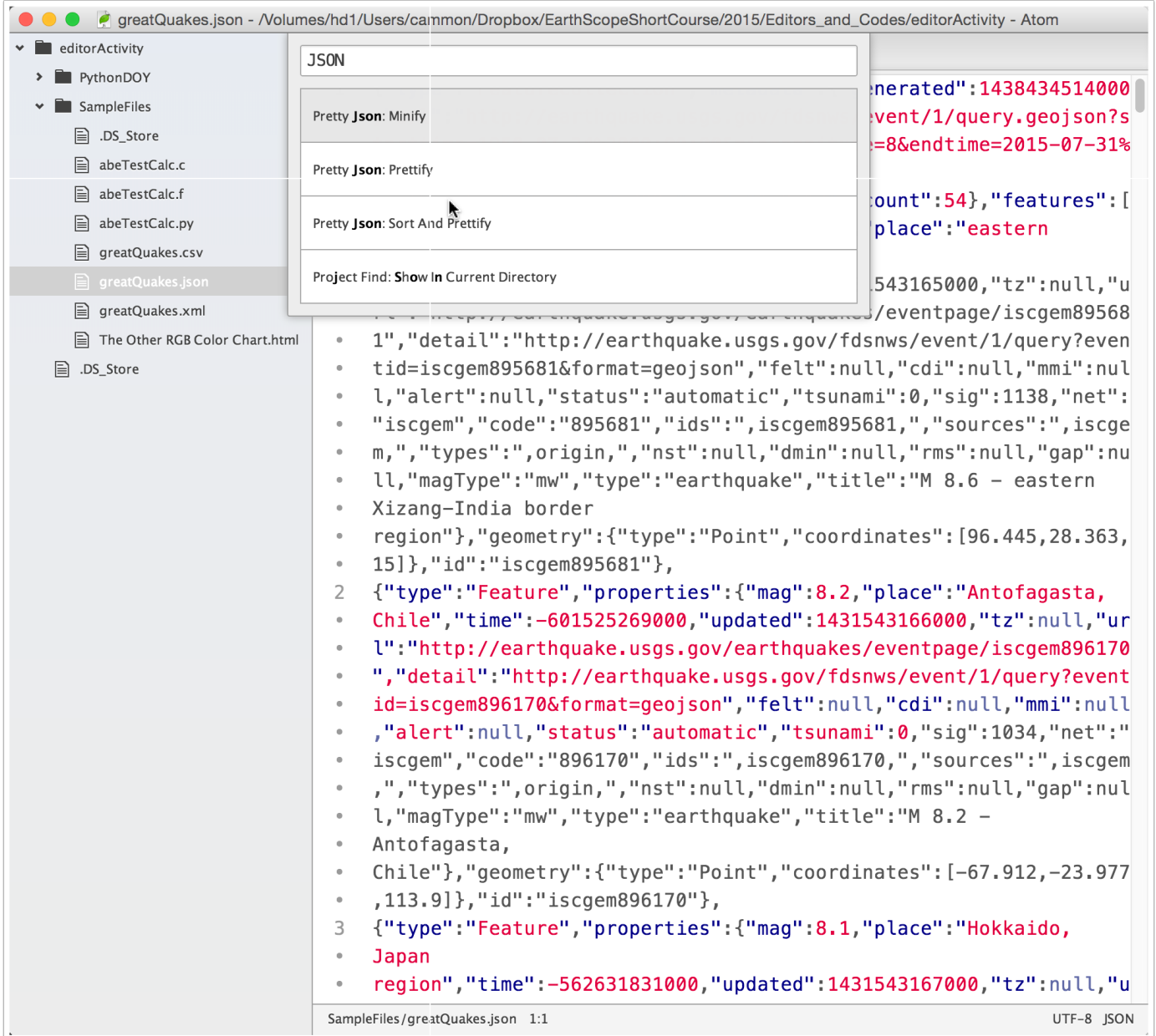
An Introduction To Atom



An Introduction To Atom

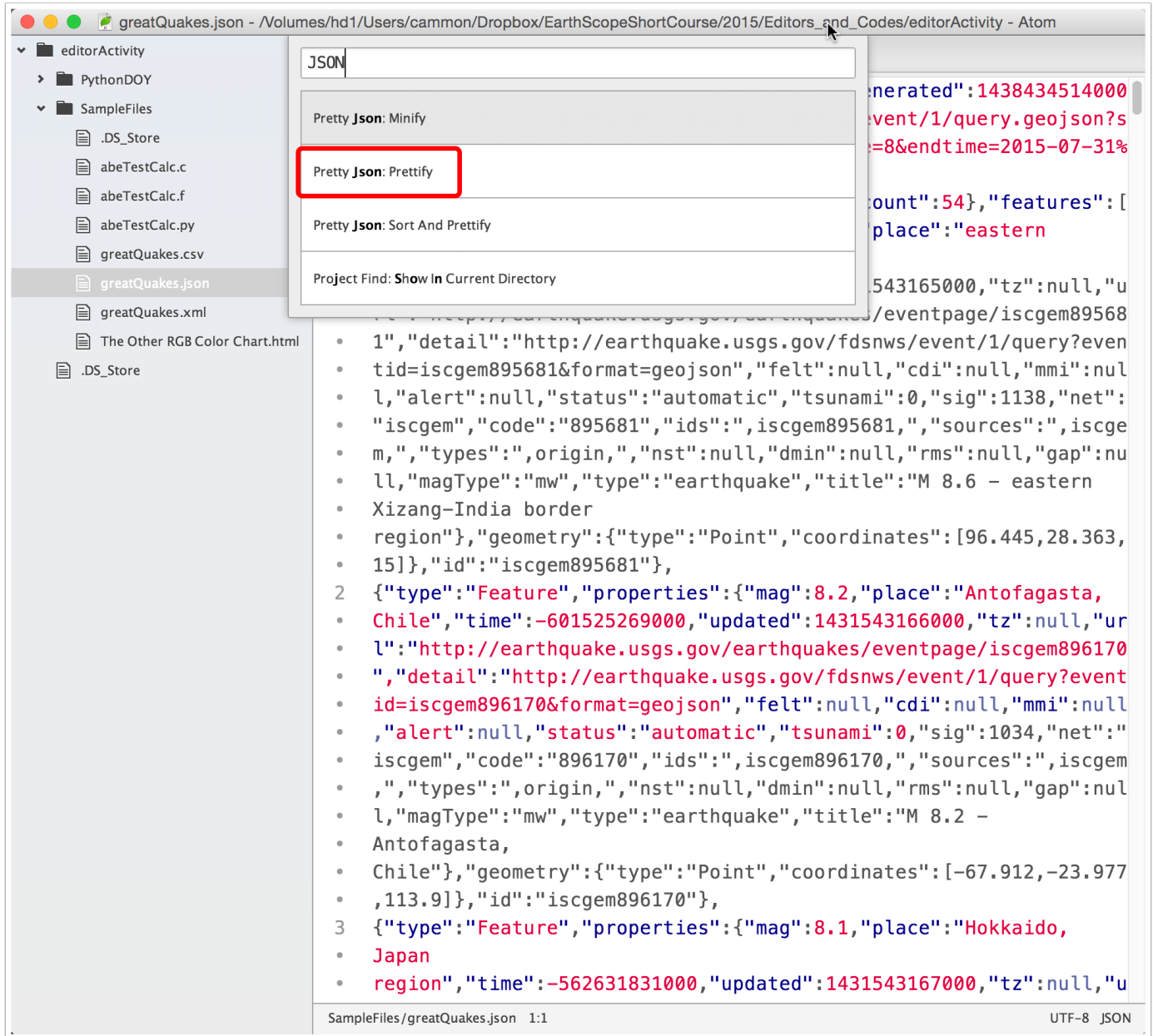
Search for a command.

The do-everything command works because Atom can search the packages and commands very quickly. Enter JSON into the search field.



An Introduction To Atom

Select the Pretty Json: Prettify Item. What happens?

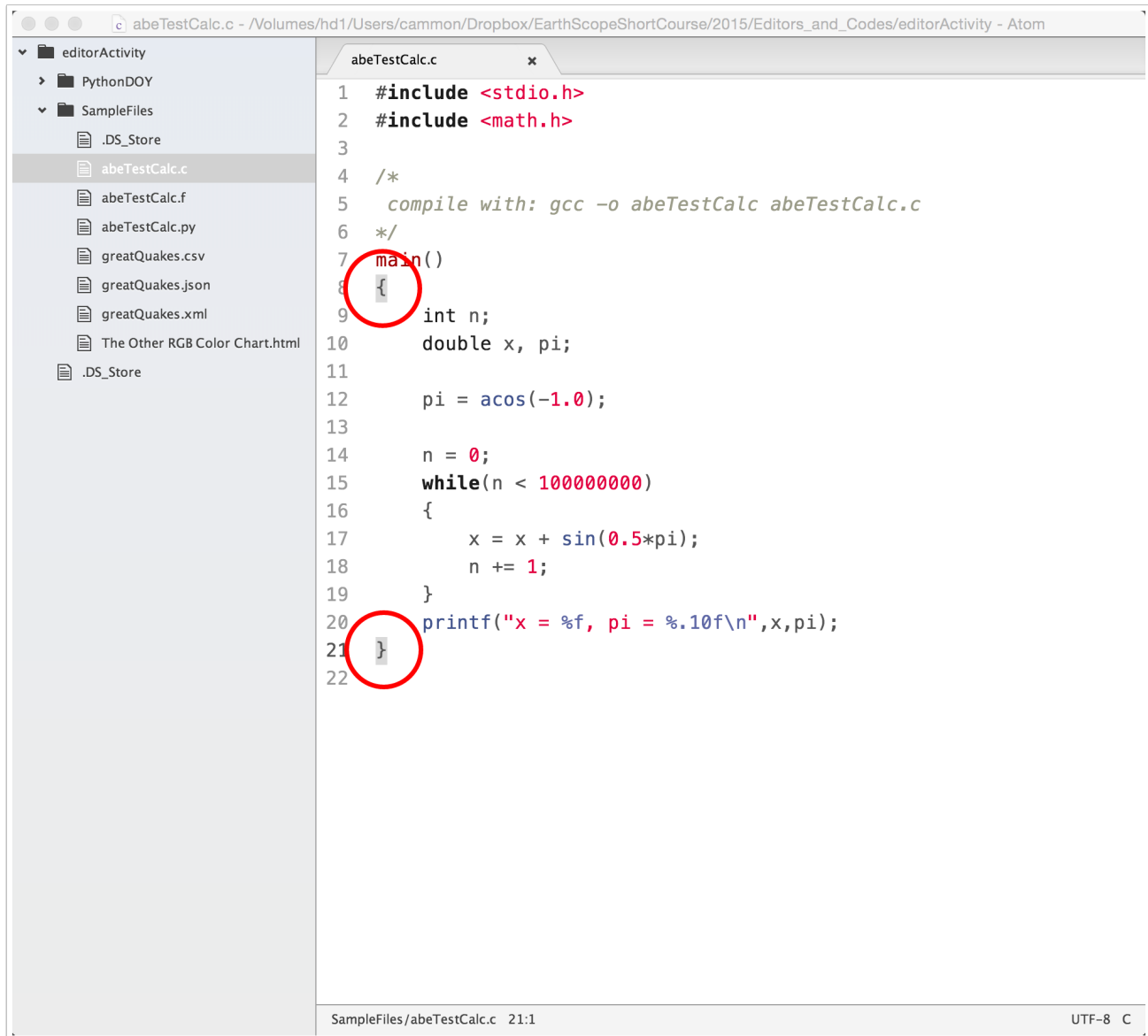


An Introduction To Atom

Language-Sensitive Tools

A good programming editor also knows about features of the language and will help you navigate and check your code. Go to the tab containing the `abeTest.c` file and single-click on one of the curly-brackets. You'll see that Atom identifies the corresponding bracket in the part of the program structure. This is very helpful with structured languages like C.

An Introduction To Atom



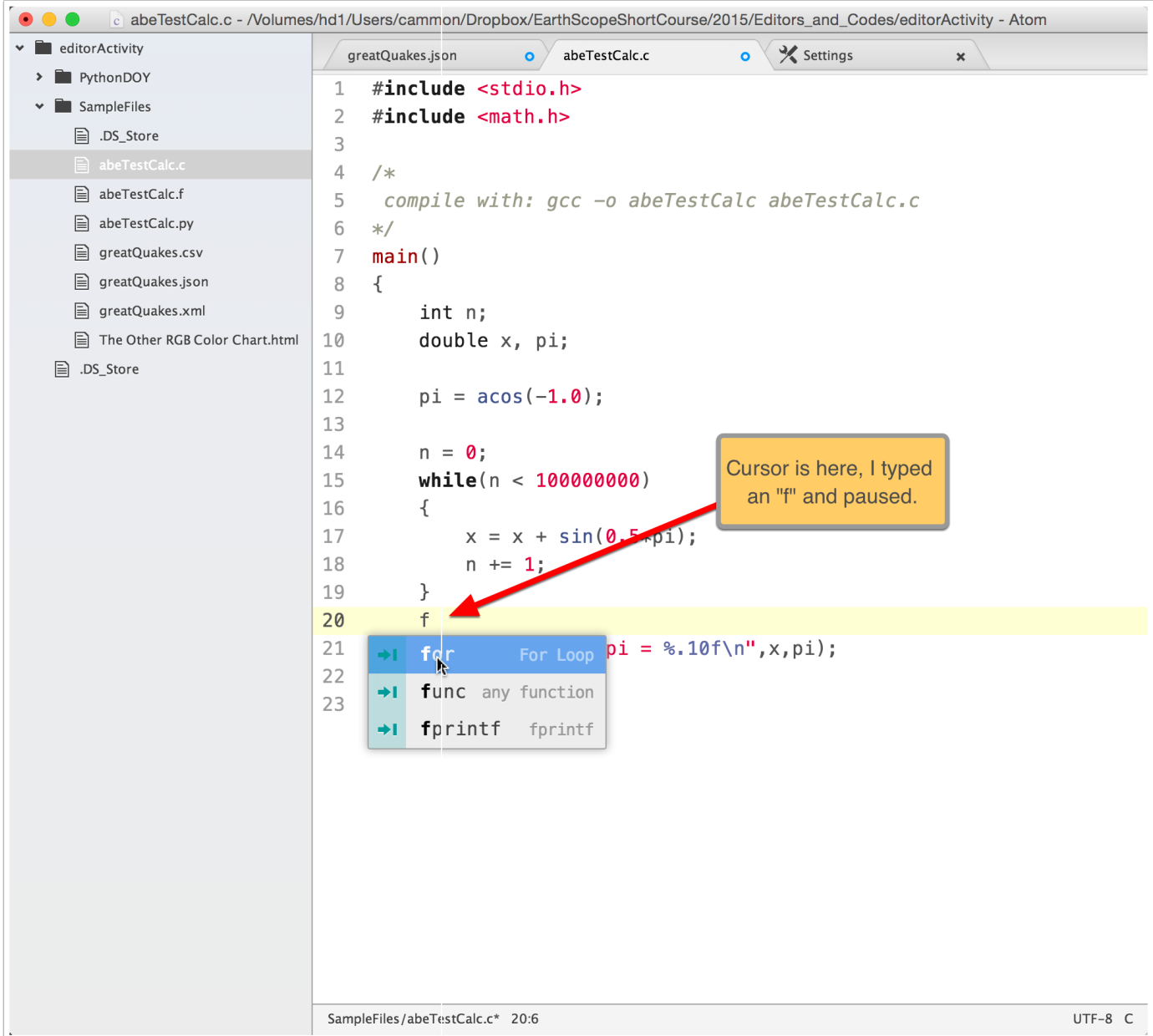
```
1 #include <stdio.h>
2 #include <math.h>
3
4 /*
5  compile with: gcc -o abeTestCalc abeTestCalc.c
6 */
7 main()
8 {
9     int n;
10    double x, pi;
11
12    pi = acos(-1.0);
13
14    n = 0;
15    while(n < 100000000)
16    {
17        x = x + sin(0.5*pi);
18        n += 1;
19    }
20    printf("x = %f, pi = %.10f\n",x,pi);
21 }
22
```

SampleFiles/abeTestCalc.c 21:1 UTF-8 C

An Introduction To Atom

Keyword Completion

When you are programming a good editor will also provide shortcuts to commonly used commands in that language.



The screenshot shows the Atom text editor with a file named `abeTestCalc.c` open. The editor's interface includes a sidebar on the left with a file explorer, a top tab bar, and a main code area. The code is a C program that includes `<stdio.h>` and `<math.h>`, and contains a `main()` function. The function declares `int n;` and `double x, pi;`, and initializes `pi = acos(-1.0);`. It then enters a `while` loop that calculates `x = x + sin(0.5 * pi);` and increments `n` until `n` reaches `100000000`. On line 20, the character `f` is typed, and a keyword completion menu is displayed. The menu lists three options: `for` (For Loop), `func` (any function), and `fprintf` (fprintf). A red arrow points from a text box to the `f` character. The text box contains the text: "Cursor is here, I typed an 'f' and paused."

```
1  #include <stdio.h>
2  #include <math.h>
3
4  /*
5   compile with: gcc -o abeTestCalc abeTestCalc.c
6  */
7  main()
8  {
9      int n;
10     double x, pi;
11
12     pi = acos(-1.0);
13
14     n = 0;
15     while(n < 100000000)
16     {
17         x = x + sin(0.5 * pi);
18         n += 1;
19     }
20     f
21     for      For Loop
22     func     any function
23     fprintf  fprintf
24
25     pi = %.10f\n", x, pi);
```

Cursor is here, I typed an "f" and paused.

SampleFiles/abeTestCalc.c* 20:6 UTF-8 C

An Introduction To Atom

Explore

That's all I wanted to cover in this brief activity.

Good editors are powerful tools, I strongly encourage you to explore your editor. A good editor saves you time and helps you write better code.