How To:
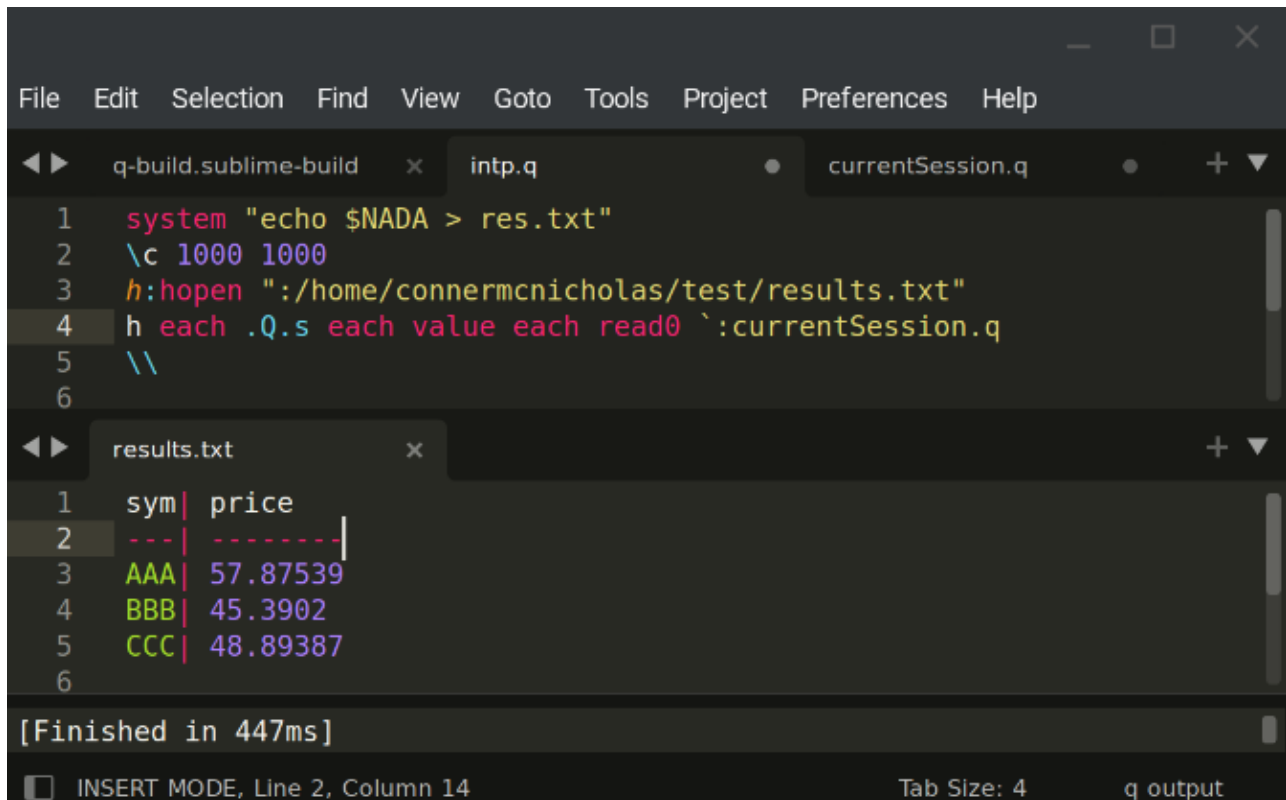
Create An Interactive q/KDB+ Sublime-Text Environment

Motive:

If you caught David Attenborough's latest documentary about KDB+ developers[1], you learned that this species prefers to code directly in the terminal. And if you read the 2021 Stack Overflow Developer's Survey[2], you know that developers who instead choose to employ an IDE generate code consisting of 78% for-loops. *In the words of Donald Trump - **"BAD!"***

That being said, not everybody has the intellectual horsepower required to install rlwrap. Sara McLaughlin reminds us in tear-jerking infomercials that practitioners of the object-oriented paradigm are particularly challenged by the concept of keyboard shortcuts. *In the words of Donald Trump - **"SAD!"***

I haven't done community service without a court order, so to claw back the karma to which I am entitled - I present this guide as a selfless gesture of philanthropy.

Steps:

From Sublime Text, create 2 groups: **[From Toolbar] View -> Groups -> New Group**

Split groups horizontally: **[From Toolbar] View -> Layout -> Rows: 2**

Create a folder in the location of your choice. I will be using a folder "SubQ" in my home directory.

With focus on Group 1 (upper group), create file "q-build.sublime-build", saving it in the Sublime build system subdirectory.

In Mac and Linux, respectively:

```
~/Library/Application Support/Sublime Text 3/Packages/User/
~/.config/sublime-text-3/Packages/User/
```

The file content should read as follows, substituting your [username] and your [OS/memory].

```
{
  "cmd": ["/home/[USERNAME]/q/[w|m|l|32|64]/q","intp.q"]
}
```

From the SubQ directory, create symbolic link to Sublime build system directory, .e.g:

```
sudo ln -s "~/.config/sublime-text-3/Packages/User/q-build.sublime-build" q-build.sublime-build
```

Create another file "intp.q" (don't question my naming conventions - I was a "Q For Gods" instructor, k?)

```
system "echo $NADA > results.txt"
\c 1000 1000
h:hopen ":/home/[USERNAME]/SubQ/results.txt"
h each .Q.s each value each read0 `:currentSession.q
\\
```

Change focus to Group 2: **[From Toolbar] View -> Focus Group -> Next**

Start a new file in Group 2 , saving it empty as "results.txt" within your SubQ directory.

Revert back to Group 1, and start a new file within your SubQ directory, "currentSession.q". This file serves as the "pane" from which you will form and execute your q commands from. Upon generating an q expression, such as:

```
([]sym:10?`AAA`BBB`CCC;price:10?100f;size:1000i)
```

Press Cmd+B to execute the statement.

Output is generated in "results.txt", located in the lower pane. The console at the very bottom will also display any errors, as well as timings for the queries, e.g.

```
[Finished in 447ms]
```

*To enable syntax highlighting in the q scripts and results file, you should also install the [Sublime-Text q/kdb+ package](#):

Reflections:

This approach to q development is discouraged. I only did it to learn how to configure build systems in Sublime Text.

If you aren't a professional q dev with an enterprise license for the Kx Systems IDE stack (Control/Developer/Analyst/Dashboards), there are a few open source options, such as qConsole: https://code.kx.com/q/learn/tour/ide/#third-party-ides

Personally, I use JupyterQ.

Or, that's what I *would* say if I was totally incompetent. I use the terminal.

Bibliography:

[1] Planet Q, Narrated by Sir David Attenborough

[2] 2021 Stack Overflow Developer's Survey