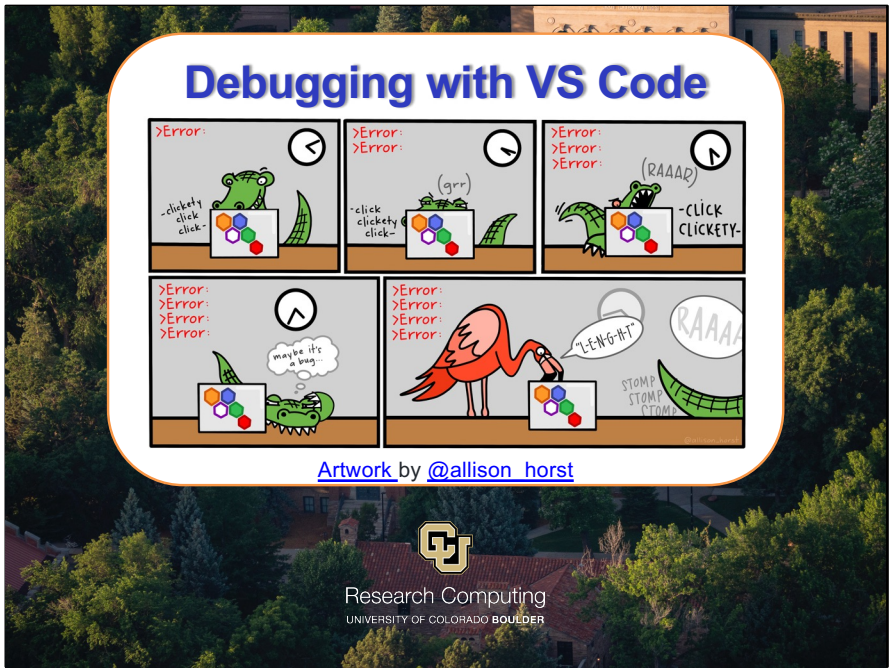


Artwork Source: https://allisonhorst.com/everything-else

Website: www.rc.colorado.edu

Documentation: https://curc.readthedocs.io

Helpdesk: rc-help@colorado.edu

Survey: http://tinyurl.com/curc-survey18

Research Computing
UNIVERSITY OF COLORADO BOULDER

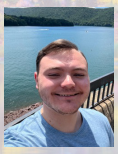
Meet the User Support Team

Layla Freeborn

Brandon Reyes

Andy Monaghan

Michael Schneider



John Reiland

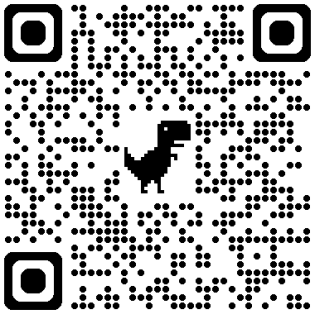
Dylan Gottlieb

Mohal Khandelwal

Ragan Lee

Slides

https://github.com/ResearchComputing/debugging_with_vs_code_shortcourse



Debugging with VS Code

Instructor:

Research Computing

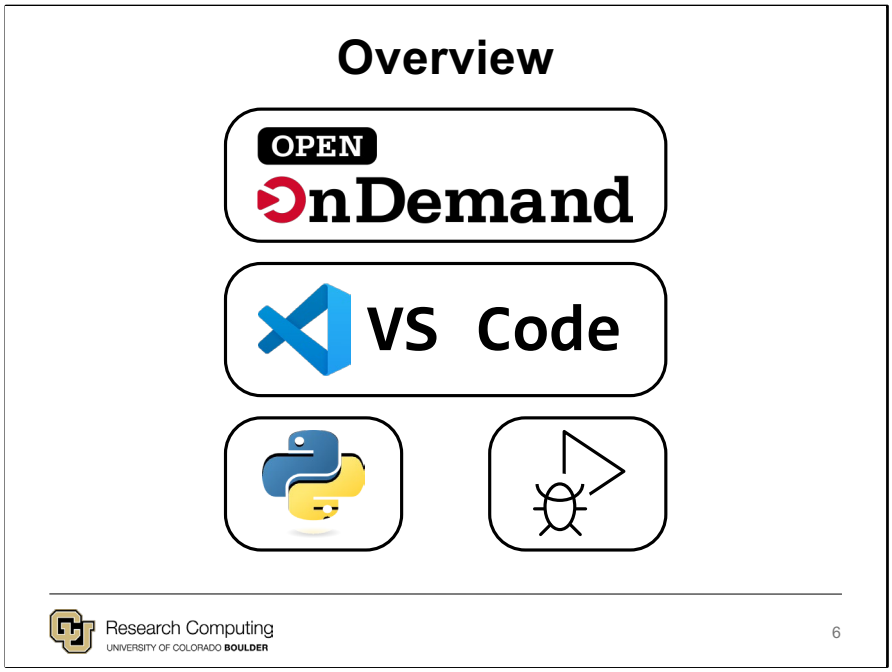
- Website: www.rc.colorado.edu
- Helpdesk: rc-help@colorado.edu

Slides:

https://github.com/ResearchComputing/debugging_with_vs_code_shortcourse

Survey:

http://tinyurl.com/curc-survey18

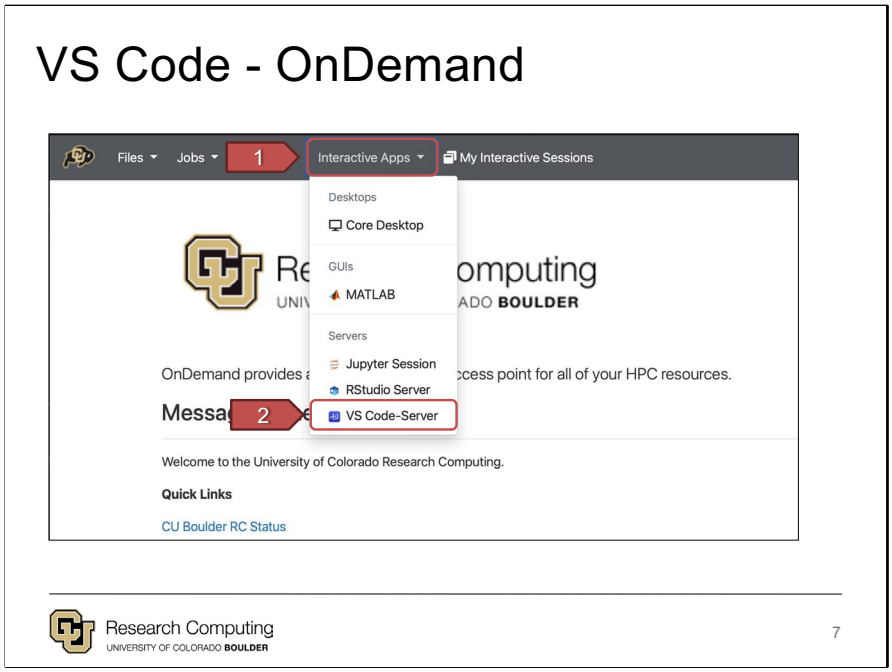


This presentation covers the basics of using VS Code's (Visual Studio Code) built-in debugger and is divided into three parts:

(1) Demonstrate how to start a VS Code Server through OnDemand - https://ondemand.rc.colorado.edu/pun/sys/dashboard

(2) Provide an overview of VS Code's user interface

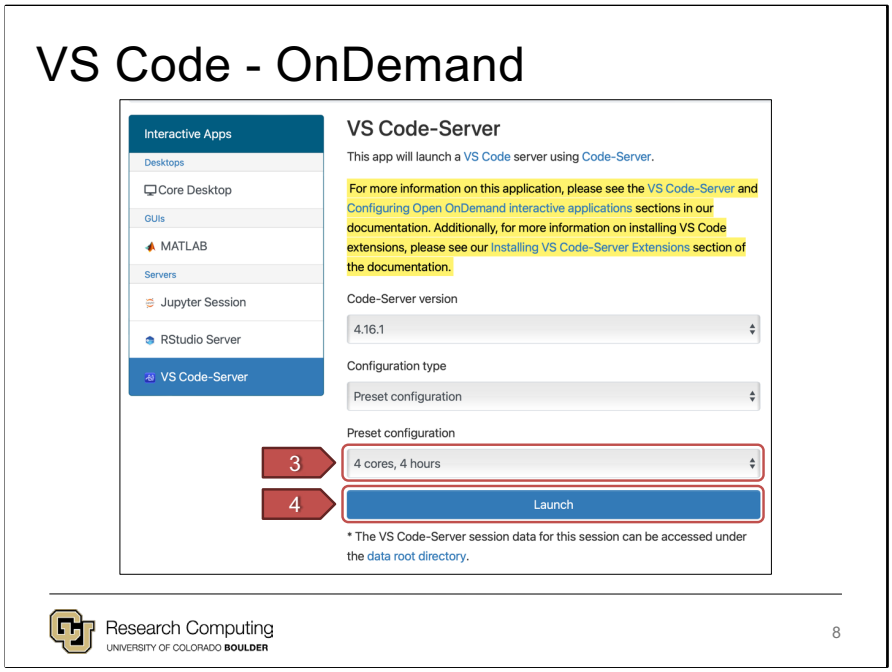
(3) Explain how to prepare and then use the debugger on a Python program



(1) Open the Interactive Apps dropdown (Top Menu Bar)

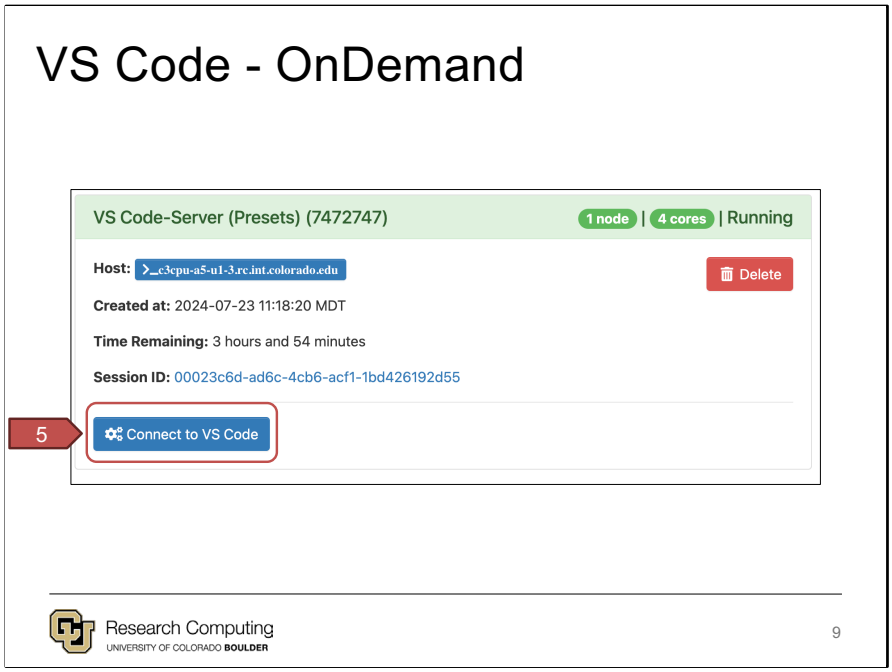
(2) Select “VS Code-Server”

Note:
VS Code OnDemand Documentation:
https://curc.readthedocs.io/en/latest/gateways/OnDemand.html#vs-code-server



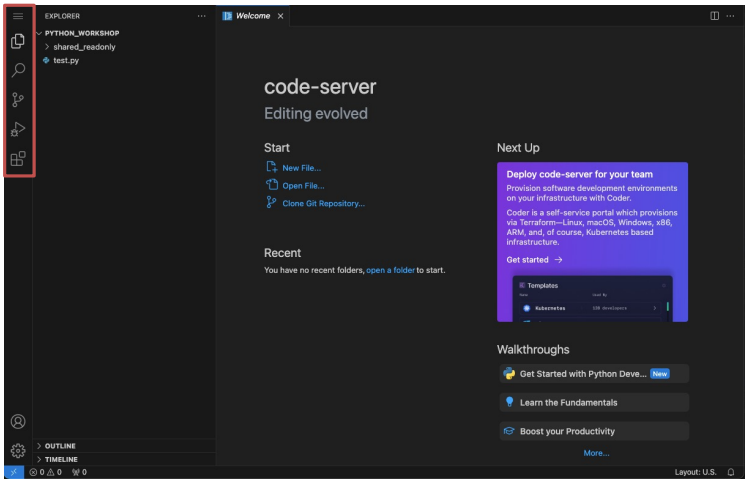
(3) Select your preferred configuration [4 cores, 4 hours]

(4) Launch the VS Code Server



(5) Once the server has started, click “Connect to VS Code”

VS Code - OnDemand



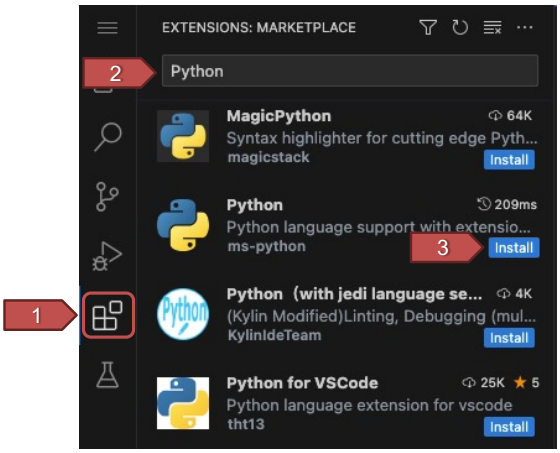
Research Computing
UNIVERSITY OF COLORADO BOULDER

10

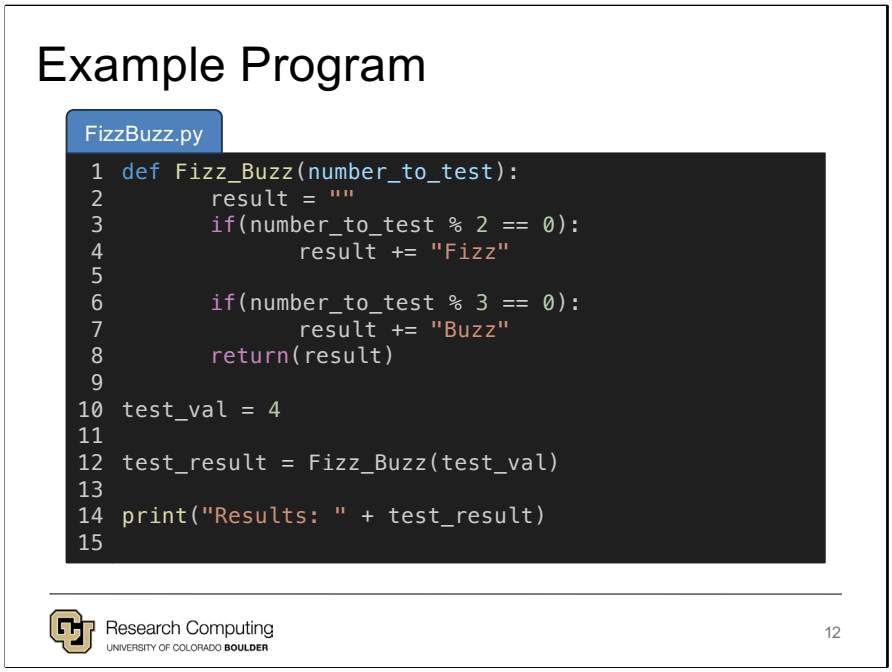
To debug a python program, VS Code will need to have the Python Extension installed.

Extensions can be found by clicking the stack of blocks in the top-left of the user interface.

VS Code – Extensions

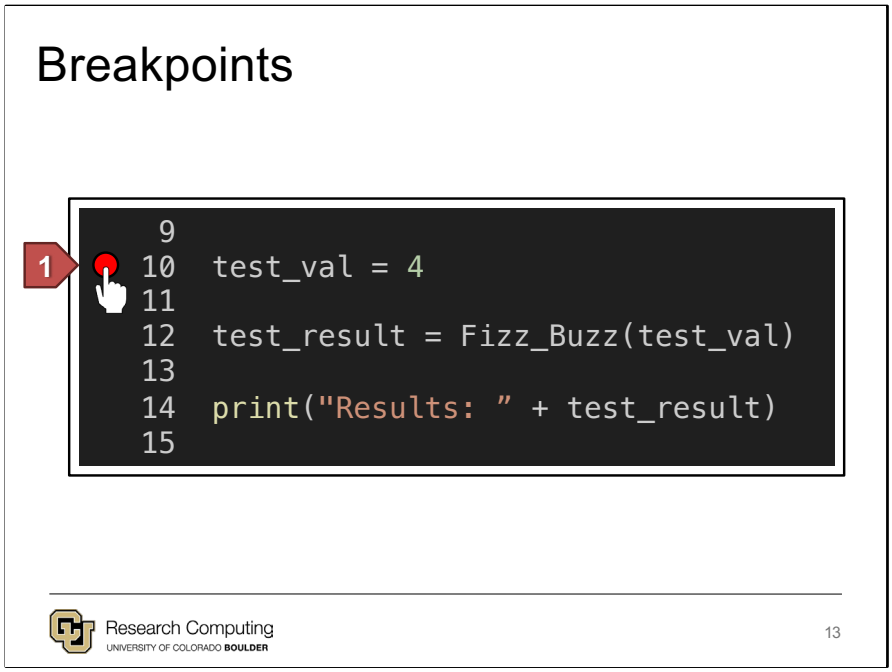


(1) Click the “Extensions” tab
(2) Type “Python” into search provided search bar.
(3) Click “Install” next to the Python extension



Copy+paste the FizzBuzz.py's code into a new file in your VS Code session.

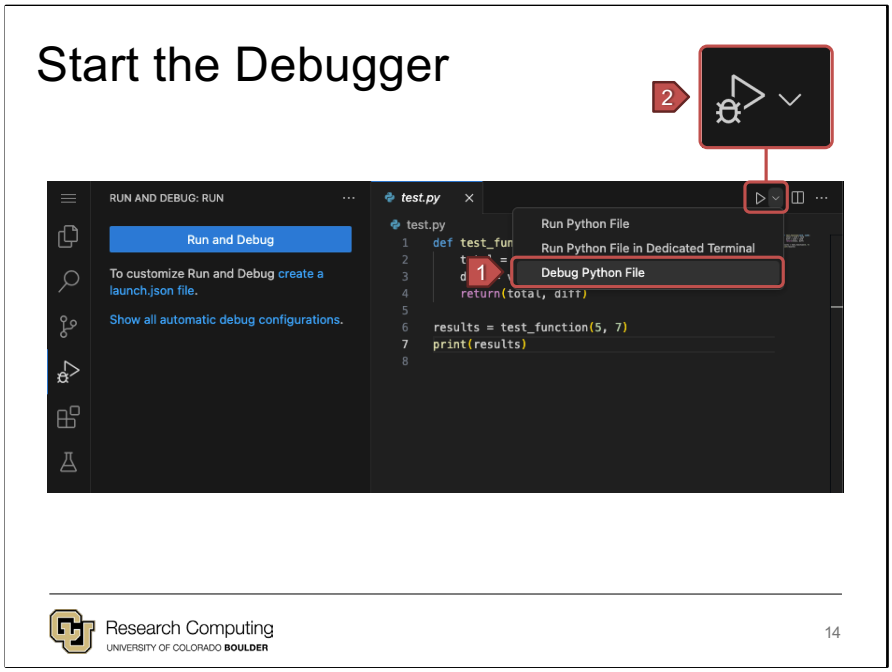
This file is also provided in this presentations Github Repo:
https://github.com/ResearchComputing/debugging_with_vs_code_shortcourse



Breakpoints can be used to temporarily pause a program's execution.

While paused, the state of the program's data (variables, registers, etc.) can be observed and even modified.

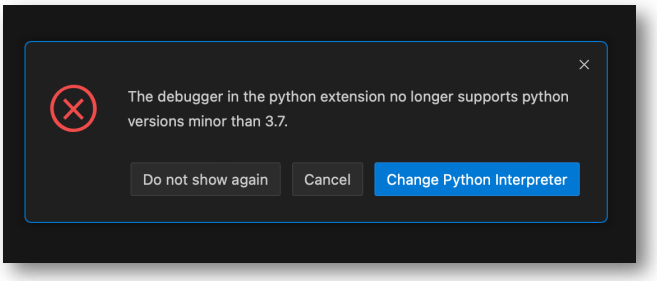
(1) Add a breakpoint on Line 10 test_val=4



(1) Click the dropdown and select “Debug Python File”

(2) The “Run” button should be replaced with the “Debug” button

Python Interpreter Error

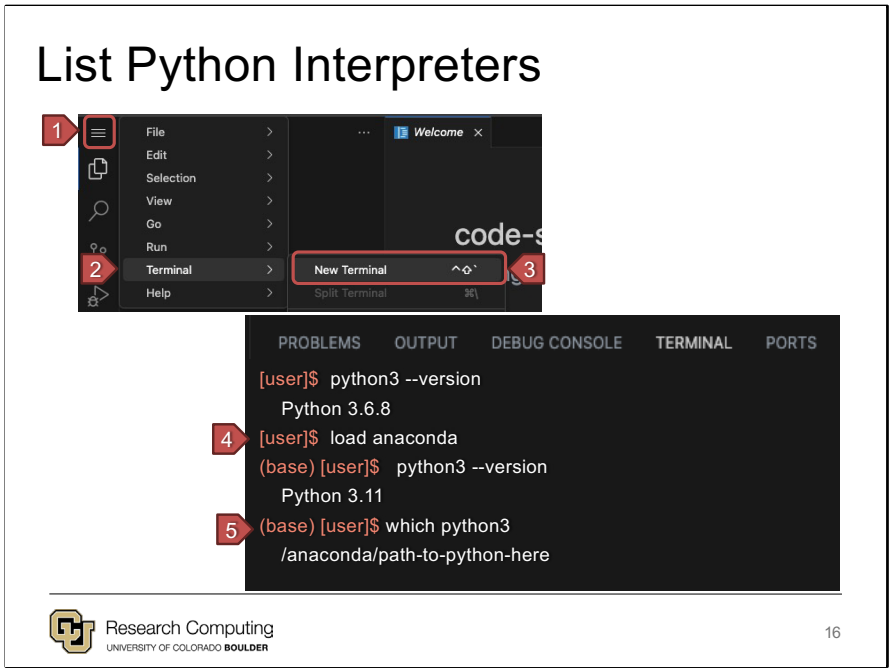


Research Computing
UNIVERSITY OF COLORADO BOULDER

15

The default Python Interpreter for the OnDemand (Viz) nodes is 3.68, which is not supported by the debugger.

If you haven't set the Python Extension to use a different interpreter (like Anaconda), then you will likely see this error message.

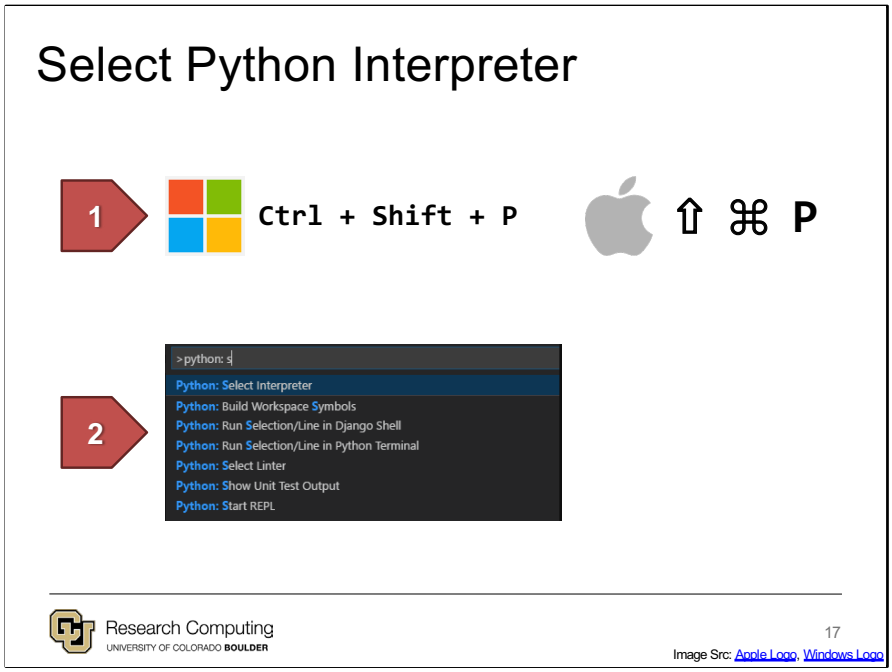


To change the Python Interpreter:
(1-3): Open a New Terminal

In the Terminal:
(4) Load anaconda (base environment)

(5) List the file path to the base Anaconda's python interpreter

Make sure to copy+paste the file path

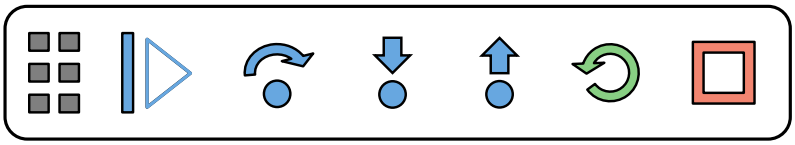


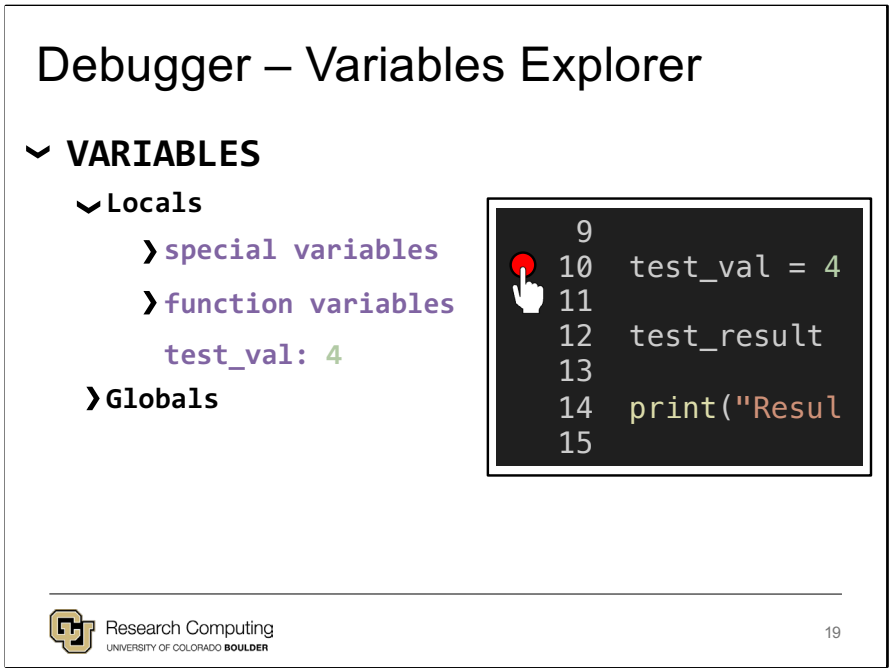
(1) Open the Command Pallete in VS Code (short cut keys for Windows (Left) and Mac (Right)

(2) Pick the python extensions “Select Interpreter” option.

Information on Selecting the Python Interpreter in VS Code: https://code.visualstudio.com/docs/python/environments#_working-with-python-interpreters

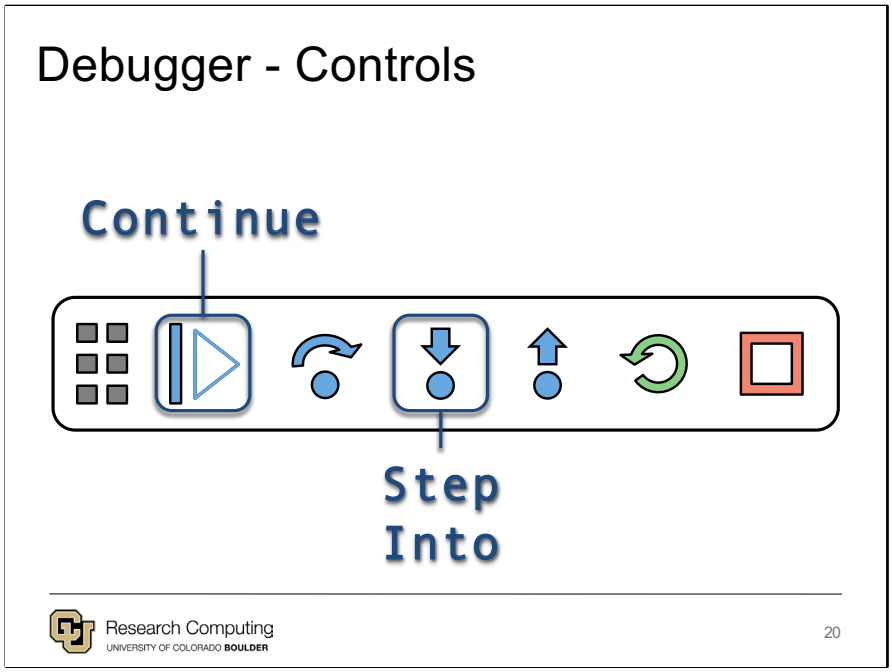
Debugger - Controls





While paused on line 10, the user should see a similar set of variables and values.

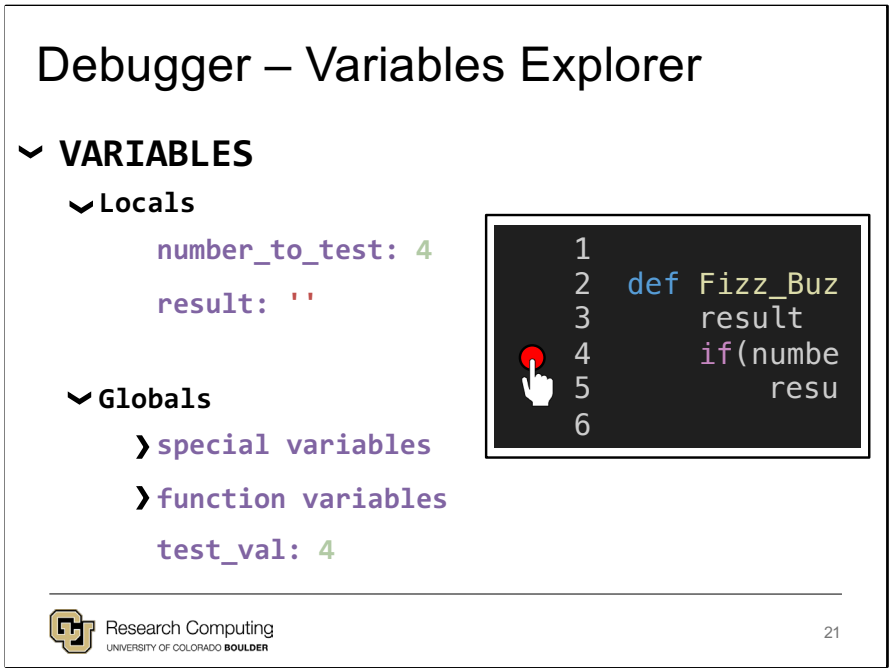
Demo: Show how the variables can be viewed and modified.



The “Continue” button will move from breakpoint to breakpoint, until reaching the end of the program’s execution (or an error is reached).

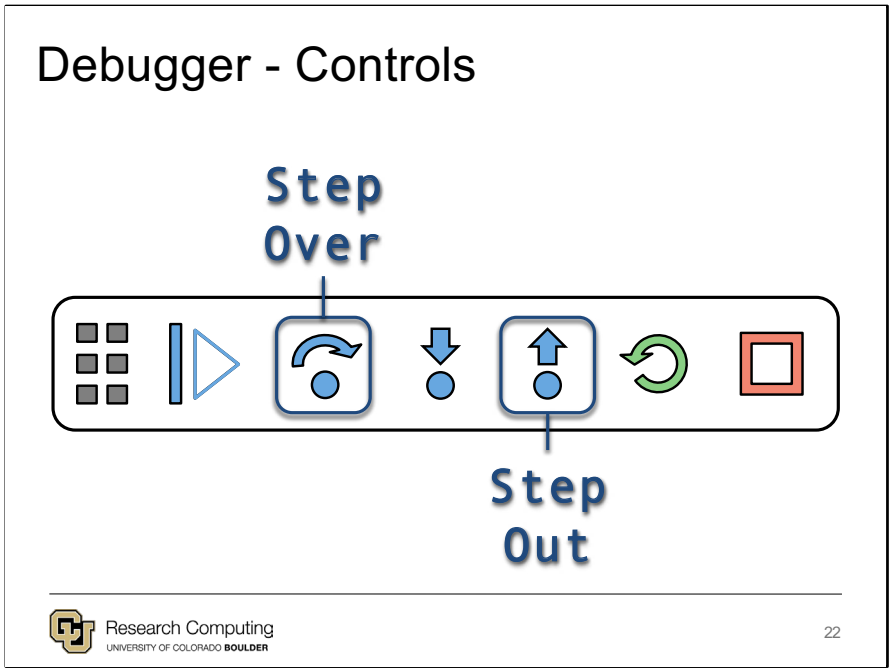
The “Step Into” button, enables users to observe the path of a program’s execution line-by-line, including “stepping into” functions called.

Demo: Show how each option affects the execution of the program.



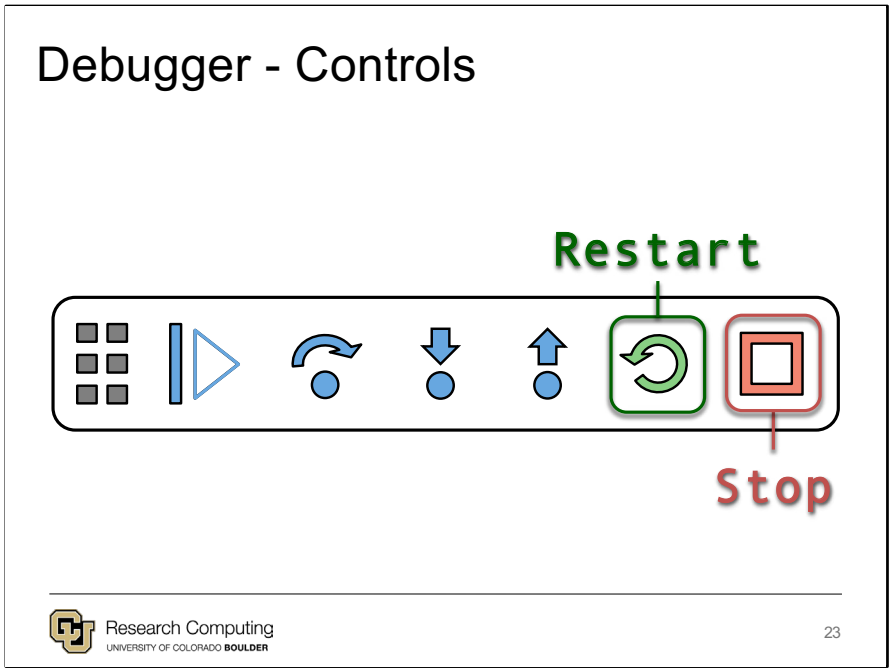
Once inside the “Fizz_Buzz” method, the variable display will update as shown.

Note how test_val is now listed as a ”Global” and the appearance of the method’s variables, “Local”



Step Over – This button enables for coding stepping that stays in the current scope – e.g. no stepping into a method call, loop, or conditional statement

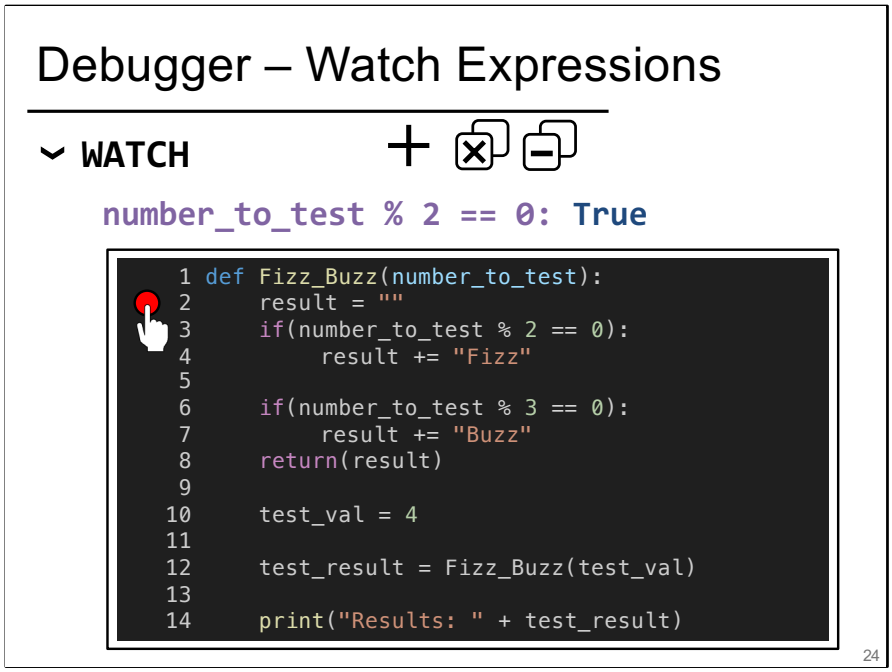
Step Out – This button continues code execution and pauses at the calling/higher level of scope – e.g. the point where a method was call, the end of a loop, or outside a conditional statement.



Restart – This button will end the program's execution and then start over

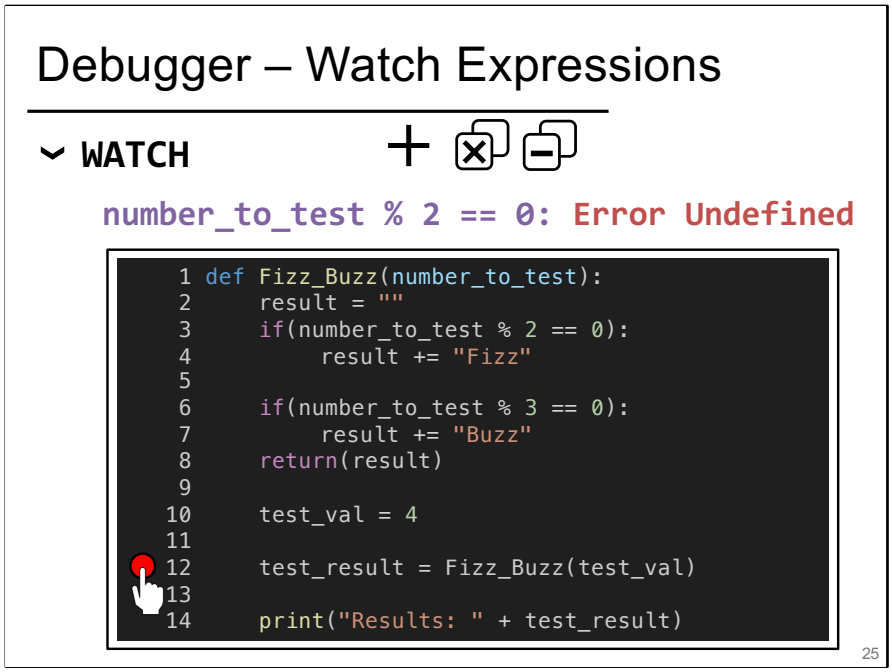
Stop – This will stop both the program's execution and the debugger

Key warning – Any changes made to variables within the debugger are only temporary. That's great for testing, but does mean the data will be gone when restarting or stopping a debugging session



Individual variables (e.g. an object's field) or programmatic expressions can be observed with the "Watch" feature.

The variables or expression being watched will be automatically updated, always providing an up-to-date view of system information.



Be mindful, you will often run into “Undefined” errors with watchlists because the variables used are no longer in-scope.

Survey and feedback

http://tinyurl.com/curc-survey18