SAFIA ABDALLA
@CAPTAINSAFIA
SAFIA.ROCKS

HOW TO CROSS THE ASTEROID BELT

INTRODUCTION

A FEW QUESTIONS

- Do you have experience writing code in Python?
- Do you have experience writing code in JavaScript?
- Have you been using Jupyter Notebooks for less than a year?
- Have you been using Jupyter Notebooks for between 1 and 3 years?
- Have you been using Jupyter Notebooks for more than 3 years?
- Have you installed a Jupyter extension before?

Outline:

- Introduction (15 minutes)
- A JavaScript primer (45 minutes): Get a brief overview of JavaScript, the central language in the Jupyter ecosystem
- Jupyter extensions (45 minutes): Learn how to develop Jupyter extensions and develop an extension that ~~autoformats the code in code cells~~ creates a table of contents
- Jupyter kernels (1 hour): Learn how the Jupyter frontend interfaces with Jupyter kernels and ~~develop a bash kernel using the Python wrapper~~ develop a Pig Latin kernel using the Python wrapper
- Conclusion and Q&A (15 minutes): Explore resources for learning more about developing on the Jupyter Notebook, example exercises, and how to use your new knowledge to improve workflows and productivity at your company or research lab

GOALS

- You will be comfortable building your own notebook extensions
- You will be comfortable implementing your own kernel implementations
- You will be comfortable with the underlying structure of the Jupyter ecosystem

JAVASCRIPT PRIMER

JUPYTER EXTENSIONS

JUPYTER EXTENSIONS

- Written in JavaScript, Python, or both
- Can be applied to the notebook, file tree, or editor
- Can be used to extend server-side and front-end functionality
- Can be distributed as Python packages or JavaScript files
- Interfaced with via the `jupyter nbextension` or `jupyter serverextension` commands

JUPYTER KERNELS

WHAT IS A KERNEL?

- A kernel is a program that runs and introspects the user's code.
- The Jupyter front-end and console communicate with the kernel via JSON payloads.
- Kernels can be implemented in a Python wrapper or in the native language of the kernel.
- Communication happens over sockets implemented in ZeroMQ.

TYPES OF KERNEL SOCKETS

- Shell
  - Request-reply pattern
  - Used for
    - Tab completion
    - Code Execution
- IO Publication
- Standard Input
- Control
- Heartbeat

TYPES OF KERNEL SOCKETS

- Shell
- IO Publication
  - Emits information
  - Used for:
    - Providing information about kernel state
    - Streams outputs that are side effects of executions
- Standard Input
- Control
- Heartbeat

TYPES OF KERNEL SOCKETS

- Shell
- IO Publication
- Standard Input
  - Used to request input from the user
- Control
- Heartbeat

TYPES OF KERNEL SOCKETS

- Shell
- IO Publication
- Standard Input
- Control
  - Similar to the shell socket but operates on a different socket
  - Used for:
    - Kernel interrupt
    - Kernel shutdown
- Heartbeat

TYPES OF KERNEL SOCKETS

- Shell
- IO Publication
- Standard Input
- Control
- Heartbeat
  - Used to ensure that the kernel is still alive

CONCLUSION

ADDITIONAL RESOURCES

- Echo Kernel [link]
- Kernel Class [link]
- Messaging in Jupyter [link]
- Distributing Jupyter extensions as Python packages [link]
- nteract: building on top of Jupyter [link]
- Jupyter Contrib Extensions [link]

QUESTION AND ANSWER