Hamilton College CS

Clojure in VS Code using Calva

We will be using VS Code with a plugin called Calva to edit and run Clojure. You will need to install these on your own machines. Here are the high-level instructions:

Install Java

1. Go to: https://www.java.com/en/
2. Follow the install directions

Install VS Code and Calva

1. Install VS Code: https://code.visualstudio.com/
2. Install Calva in VS Code:
   a. In VS Code, click the "menu" gear
   b. Click Extensions
   c. Search for Calva
   d. Install Calva
3. Check out the Calva user's guide: https://calva.io/

Creating a new project

1. Choose a project name made only of lower-case letters. In the instructions below, I'll assume you chose the name `haha`. You should replace haha with the name that you actually chose.
2. Create an empty folder called `haha`.
3. Create the files and folders needed for the project in `haha`:
   a. Create a text file in haha called `deps.edn` that contains just this: `{}`
   b. Create an empty folder in `haha` called `src`
   c. Create an empty folder in `haha/src` that is also called `haha`
   d. Create a text file in `haha/src/haha` called `core.clj` that contains just this: `(ns haha.core)`
   e. Make sure that the text files you created don't have `.txt` extensions added by your operating system, which might add them but then hide them if it's really sneaky. The extension for the first one should be just `.edn`, and the second one should be just `.clj`.
4. You should end up with the following files and directories:

```
haha
├── deps.edn  -- contains just {}
└── src
    └── haha
        └── core.clj -- contains just (ns haha.core)
```

Running Clojure in VS Code

To run Calva commands, open VS Code's command pallet using `View > Command Pallet` (or use the keyboard shortcut, which on Mac is `Cmd + Shift + P`). Then, type in the command name and hit enter. You can use this to look up the keyboard shortcuts to each of these.

1. Load your project directory (i.e. `haha`) in VS Code
   a. I just drag the directory to VS Code
2. Open a Clojure file, such as `src/haha/core.clj`
3. To connect to a REPL: `Calva: Start a Project REPL and Connect (aka Jack-In)`
   a. you'll need to then select `deps.edn`.
4. From within your file: `Calva: Load/Evaluate Current File and its Requires/Dependencies`
5. Create a `comment` block with a function call to run inside it
6. On the line of the function call, run the code and show the result using `Calva: Evaluate Top Level Form (defun)`
7. On the line of the function call, run the code and print the result in a comment using `Calva: Evaluate Top Level Form (defun) to Comment`

Some other useful Calva commands

Note: Calva uses paredit by default, which ensures matching parentheses. I recommend you use paredit to your advantage, and get used to opening and closing pairs of parentheses at a time.

- To turn paredit off (sometimes you need to), click the [λ] in the lower-right corner of VS Code. Click it again to turn it back on.
- `tab` will reindent your code, which can help find places where parentheses are off.
- To remove enclosing parentheses: `Calva Paredit: Splice Sexp`
- To terminate an infinite process: `Calva: Interrupt Running Evaluations`
- Learn more about paredit, slurping, and barfing here: https://calva.io/paredit/
- Learn more about using Calva by running: `Calva: Fire up the Getting Started REPL`