Introspection:
Exploring the running system

6/1/22 Brent Roman brent@mbari.org

I don’t remember how to...

- Learn how to ask the system to help jog your memory
- If you can remember a couple starting characters
  - Press <TAB> key twice to see possible completion options
    - If only one, system will immediately complete the cmd.
- List of all methods
  - pretaining to a class (e.g. Syringe, Clamp, Sampler)
  - and the text files where they are defined
- View Ruby source code for any method

TAB key for Command Completion

- Press TAB at either the Linux($) or ESP(->) prompts
  - If there is only one completion possible, it will be input for you
  - If there are multiple possible, they will be listed in a table
    - just keep typing
      - pressing TAB when you what a list of options
    - Try:

```
-> acc<TAB>        #shows access
-> access<TAB>     #shows access  accessTube
```

  - Press ***<RETURN>*** to enter the command displayed

  - This also works on the Unix $ shell command line!
  - Try:

```
$ du<TAB>          #shows du   dumplog
```

List ESP methods for any Class

- To list all the methods unique to a given class:
  -> ***Class***.source
  - Try:
    -> Sampler.source #lists all the Sampler methods
    - and shows where they are defined as <*fileName:line#*>

    -> list Sampler/:puckEvac #lists code for puckEvac

```
def puckEvac
#evacuate the puck the appropriate number of times
  @numberOfEvacs.times {puckEvacOnce}
end
```

-> list CC.method :close #lists code for CC.close

```
  def close *args
  #closing the collection clamp causes the next sampling
  #to recalibrate pressure sensors and reset sampling speed
...
```

View and Edit ESP methods

- Caveat: View and Edit may not work from espclient :-(

- To view a method_name's source in a text editor:
  `-> view` ***`Object`***`.method` ***`:method_name`***
  - Try:
    `-> view CC.method :close` #lists code for CC.close

    - omit ***Object.*** if method is called without it
      `-> view method :noKill` #lists source code noKill

    - Use reconnect command as alternative to espclient

Reconnect to ESP server

- Alternative to “start esp” (or “start”, actually)
  - Prevents esp server from exiting on network failure
  - Does NOT produce a *mode*.out log file
  - Requires that reconnect script be installed on your laptop
    - Known to work from Linux and MacOS
    - Built atop Unix “screen” command

- Start server from remote laptop (or desktop):

  ```
  $ reconnect esp@ESPhostname esp
  ```

  ... wait while esp server starts ...

  - Press key sequence <CTRL-a> <CTRL-d> to disconnect
    - or break the network connection some other way
  - Later, to reconnect to the server:

    ```
    $ reconnect esp@ESPhostname  #omit esp command
    ```

    - your screen will appear exactly as you left it!

    ```
    $ reconnect -help  #displays reconnect options
    ```