Hands On! Exercises

▫ EXERCISE 1: Try Nightcode InstaREPL

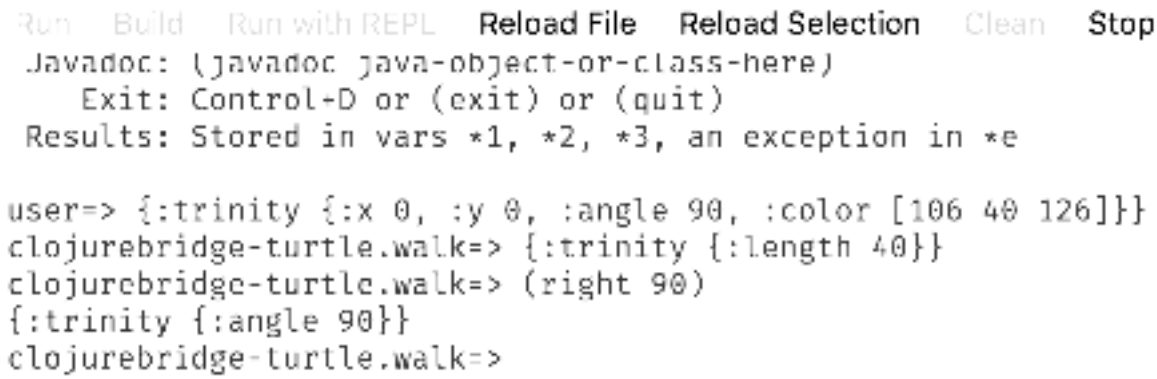
1. Start Nightcode
2. Import myproject (which you created while testing Leiningen setup)
3. Open core.clj (myproject -> src -> myproject -> core.clj)
4. Click InstREPL button
5. Type the Clojure functions below and see what happens

```
(print-str "Hello, World!")
(print-str "Hello, World!" " " "from Clojure")
(+ 3 4)
(- 3 4)
(* 3 4)
```

!!! Make sure you type the lines exactly as you see them above, taking care to put the parentheses in the right locations.

▫ EXERCISE 2: Evaluate file and line

- Open the file welcometoclojurebridge/src/clojurebridge_turtle/walk.clj
- Evaluate the entire file by hitting "Run with REPL" followed by "Reload File"
- See what happens
- Type `(forward 40)` on the bottom line of walk.clj in the editor. Evaluate this line by selecting line and hitting "Reload Selection"
- See what happens
- Type `(right 90)` and "enter" in the REPL pane (bottom)



- See what happens to the turtle
- Take a look Turtles App API (http://j.mp/clojure-turtle-api) and How To Walk Turtles [section 1 and 2] (http://j.mp/clojure-walk-turtles), and try more commands to walk your turtle

▫ EXERCISE 3: Look at Clojure docs

- In the bottom REPL pane, try to look up the documentation for a function you have used
- You can use the `(doc function-name)` command to do this
- Try `(doc +)` and `(doc forward)` on the REPL
- Try other functions we used so far, for example, -, *, or doc

▫ EXERCISE 4: Basic arithmetic

- How many minutes have elapsed since you arrived at the workshop today? (!!! keep it simple)
- Convert this value from minutes to seconds.

▫ EXERCISE 5: See turtle names

1. Add a turtle using a piece of code in the file
    - Go to walk.clj file
    - Add to the end of the file: `(add-turtle :neo)`
    - Select this line and click "Reload Selection"
2. (Optional) add a turtle using REPL
    - Type `(add-turtle :oracle)` followed by enter on the bottom REPL pane
3. See turtle names
    - Type `(turtle-names)` on the bottom REPL pane and see the result

▫ EXERCISE 6: Make a vector

- Go to myproject's core.clj and start InstaREPL
- Make a vector of the high temperatures for the next 7 days in the town where you live.
- Then use the nth function to get the high temperature for next Tuesday.

▫ EXERCISE 7: See turtles states

- Go to walk.clj file
- Try examples of previous two slides on the REPL
- See what values you get
  !!! Don't forget to hit **enter** when you type code on the REPL

```
(state-all)
(def states (state-all))
(first states)
(def st (first states))
st
(get st :trinity)
(get-in st [:trinity :angle])
```

▫ EXERCISE 8: Modeling Yourself

- Use the myproject's core.clj and InstaREPL
- Make a map representing yourself
- Make sure it contains your first name and last name
- Then, add your hometown to the map using assoc or merge.

▫ EXERCISE 9: Y value within a frame (part 1)

- Write a function **y-within-frame** that takes y (vertical position) as an argument.
- You may use if example in the slide.
- The function should return the y value that won't exceed 150.

▫ EXERCISE 10: Y value within a frame (part 2)

The function we wrote in the previous exercise, y-within-frame, has a flaw. If the given y value is -1000, the function will return -960. Since y value of the frame bottom is -150, -960 is beyond that. Your turtle will go invisible area. Let's make it real within-frame function using cond.

- Write a function **y-within-frame-cond** that takes y (vertical position) as an argument.
- You may use the cond example in the slide.
- The function should return the y value between -150 and 150.

▫ EXERCISE 11: Move turtles using function

1. Write a function
    a. Go to walk.clj
    b. On the editor, write **forward-right** function (below) which appeared in the slide.
    c. Select whole forward-right function and hit Eval Selection
2. Use a function
    a. Type `(forward-right :trinity)` on right REPL pane
    b. Repeat above at least 8 times (use up arrow and hit enter)

```
(defn forward-right
  "Moves specified turtle forward and tilts its head"
  [turtle]
  (forward turtle 60)
  (right turtle 135))
```

▫ EXERCISE 12: Move turtles using function with parameters

- Go to walk.clj
- On the editor, write **forward-right-with-len-ang** function that takes three arguments: **turtle**, **len**, and **angle** (extension of **forward-right-with-len**)
- Select entire **forward-right-with-len-ang** function and hit Reload Selection
- On the REPL pane, type `(forward-right-with-len-ang :trinity 60 120)`
- Repeat above, evaluating the function on REPL, many times.