Errata for *Clojure for the Brave and True* (updated to 8th printing)

Page 2: Looking for the Emacs configuration file, *emacs-for-clojure-book1*? It's right here! https://github.com/flyingmachine/emacs-for-clojure/archive/book1.zip

Page 11: Added a sentence to the end of the first paragraph:
Two alternatives that I recommend and that are well regarded in the community are Cursive (https://cursive-ide.com/) and Nightcode (https://sekao.net/nightcode/).

Page 13: Added a note under the "Configuration" heading:
Note: These tools are constantly being updated, so if the instructions below don't work for you or if you want to use the latest configuration, please read the instructions at https://www.nostarch.com/clojure/.

and we added two sentences to step 2:
(Windows users, your Emacs files will probably live in *C:\Users\<your_user_name>\AppData\Roaming\*. So, for example, you would delete *C:\Users\jason\AppData\Roaming\.emacs.d*.)

and we deleted step 4, and renumbered step 5 as step 4.

and we added this as the penultimate sentence of the final paragraph:
(If you don't see any activity, quit and restart Emacs just for funsies.)

Page 26: In the first sentence of the third paragraph:
In the *core.clj* buffer, use **C-c M-n**.
should now read:
In the *core.clj* buffer, use **C-c M-n n**.

Page 68: We updated Figure 3-1 as follows:

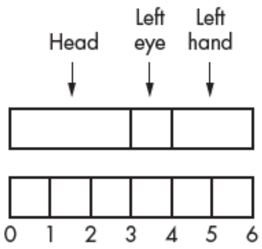

Figure 3-1: Body parts correspond to ranges of numbers and get hit if the target falls within that range.

and in the third paragraph:

The body part is hit if the accumulated size is greater than the target, so if the target is 0, 1, or 2, then the head was hit. Otherwise, you take the next part, the left eye, and increase the accumulated size to 4, yielding a hit if the target is 3. Similarly, the left hand gets hit if the target is 4 or 5.

should now read:

The body part is hit when the accumulated size exceeds the target, so if the target is less than 3, then the head was hit. Otherwise, you take the next part, the left eye, and increase the accumulated size to 4, yielding a hit if the target is greater than or equal to 3 and less than 4. Similarly, the left hand gets hit if the target is greater than or equal to 4 and less than 6.

Page 82: The second sentence that reads:

Each entry has the year, month, day, and what you ate.

should now read:

Each entry has the month, day, and what you ate.

Page 95: In the final paragraph, the second sentence that reads:

First, `map` creates a seq of key-value pairs like (`[:name "Bella Swan"] [:glitter-index] 0`).

should now read:

First, `map` creates a seq of key-value pairs like (`[:name "Bella Swan"] [:glitter-index 0]`).

Page 118: The sentence under Figure 5-4 that reads:

. . . positions 1, 6, and 11 have valid moves . . .

should now read:

. . . positions 1, 6, 11, and 13 have valid moves . . .

Page 182: The first sentence that reads:

. . . the pairs are arranged in `order-details-validation`).

should now read:

. . . the pairs are arranged in `order-details-validations`).

Page 183: The first line of the second code listing that reads:

```
(if-valid order-details order-details-validation errors
```

should now read:

```
(if-valid order-details order-details-validations errors
```

Page 184: The first line of the code listing in Exercise 1 that reads:

```
(when-valid order-details order-details-validation
```

should now read:

```
(when-valid order-details order-details-validations
```

Page 201: In the final paragraph, the two instances of “`@butter-promise`” should be “`butter-promise`.”

Page 212: The third line of the third code listing that reads:

```
    (future (println (:percent-deteriorated zombie-state)))))
```

should now read:

```
    (future (println (:cuddle-hunger-level zombie-state)))))
```

Page 217: The final code line that reads:

```
; This throws "IllegalStateException That's not mathy!"
```

should now read:

```
; This throws "IllegalStateException: That's not mathy!"
```

Page 227: The paragraph under the final code block that reads:

Using `with-redefs` may be more appropriate than using bindings for setting up a test environment. It's also more widely applicable, in that you can use it for any var, not just dynamic ones.

should now read:

`with-redefs` can be used with any var, not just dynamic ones. Because it has such far-reaching effects, you should only use it during testing. For example, you could use it to redefine a function that returns data from a network call, so that the function returns mock data without having to actually make a network request.

Pages 270–271: Three mentions of "`feelings`" (page 270, paragraph 2, sentence 2; page 270, paragraph 4, sentence 4; and page 271, paragraph 4, sentence 1) should now read "`feelings-about`."

Page 286: The final line of Listing B-1 that reads:

```
; => "I don't like 2"
```

should now read:

```
; => "I'm not going to bother doing anything to that"
```