14. Anonymous functions

As a functional language, Clojure uses a special form to build anonymous, or unnamed, functions ((Fogus & Houser, 2014)).

```
;; This function has no name
((fn [n]     ;; a vector of function params, n
  (* 3 n))            ;; return value = 3 * n
 2)                  ;; passing in param
```

Output

```
6                       ;; return value
```

There is also a reader feature to define in-place functions using `#()`.

```
usr> (#(* 3 %) 2)
;; The % implicitly declares the accepted argument
```

Adapted from (trptcolin, 2015) and (Fogus & Houser, 2014).

References

Fogus, M., & Houser, C. (2014). *The joy of clojure*. Manning Publ.

trptcolin. (2015). Clojure koans. *Accessed Online from Https:// Github.com/ Functional-Koans/ Clojure-Koans*.