

```
1 #lang racket
2
3
4 (define (hello-world) ;normal naming convention is
4 lower-case with hyphens for whitespace
5   (printf "Hello world!\n"))
6
7 (hello-world)
8
9 (define (iterate x)
10   (printf (string-append (number->string x) "\n"))
11   (if (= x 0)
12       (printf "Done!\n")
13       (iterate (- x 1))))
14
15 (iterate 10)
16
17 (define (factorial x)
18   (if (= x 1)
19       1
20       (* x (factorial (- x 1)))))
21
22 (factorial 6)
23
24
25 (define (count-up x y)
26   (printf (number->string x))
27   (if (= x y)
28       (void)
29       (count-up (+ x 1) y)))
30
31 (count-up 1 5)
32
33 (define (count-up-2 x)
34   (count-up 1 x))
35
36 (count-up-2 5)
37
38 (define (reverse-helper str n)
39   (if (= n (string-length str))
40       ""
```

```
41 |         (string-append (reverse-helper str (+ n 1))
42 |                       (string (string-ref str n))))
43 |
44 | (reverse-helper "cat" 0)
45 |
46 | (define (reverse-helper-2 str n)
47 |   (if (= 1 (string-length str))
48 |       str
49 |       (string-append (reverse-helper (substring str n
49 | (string-length str)) (+ n 1))
50 |                     (string (string-ref str n)))))
51 | (reverse-helper-2 "cat" 0)
52 |
53 |
```