

```
1 #lang racket
2
3 (define (check-type y)
4   (match y
5     ((? string? x) "string")
6     ((? number? banana) "number")
7     ((? boolean? strawberry) "boolean")
8     ((? list? x) "list")
9     (_ "other")))
10
11 (check-type (list 1 2 3))
12
13 (check-type (lambda (x) x))
14
15 (define (get-last lst)
16   (match lst
17     ((list a) a)
18     ((list a b) b)
19     ((list a b c) c)))
20
21 (get-last (list 1 2 3))
22
23 (define (get-last-2 lst)
24   (match lst
25     ((list a) a)
26     ((list _ a) a)
27     ((list _ ... a) a)
28     (else "hi")))
29
30 (get-last-2 (list 1 2 3 4))
31
32 (define (palindrome-checker lst)
33   (println lst)
34   (match lst
35     ((list a) #t)
36     ((list a a) #t)
37     ((list a _ ... a) (palindrome-checker (drop-right
37 (rest lst) 1)))
38     (else #f)))
39
40 (palindrome-checker (list 1 2 3 2 1))
```

```
41 |
42 | (define (matcher lst)
43 |   (match lst
44 |     ;;((list-rest a rest) rest)
45 |     ((list-rest a a rest) rest)))
46 |
47 | (matcher (list "cat" "cat" #f 1 "cat"))
48 |
49 |
50 |
```