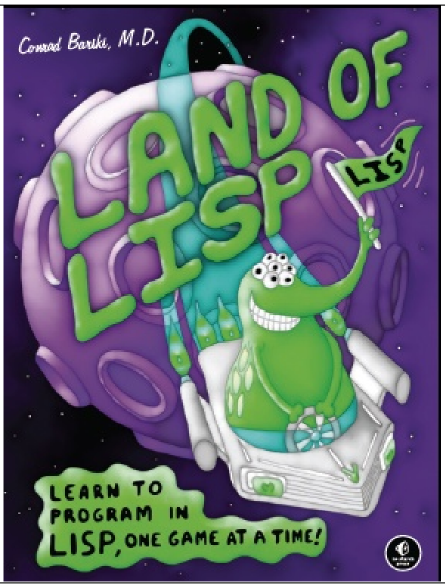

Ch. 6.5

Ch 6.5 Lambda, a Function So Important it Deserves its Own Chapter!!

HMM... THE GUY THAT LIVES HERE MUST BE SO ANNOYING, THEY GAVE HIM HIS OWN TOWN...
NOW ENTERING LAMBDA POPULATION 1
λ

First Class Functions

- In Lisp, functions are actually s-expressions (lists) that we can view and pass around just as if they were numbers or strings or …
- An experienced Lisp programmer would say that functions are *first-class objects* in Lisp

```
> (defun half (n)     ; define fn
        (/ n 2))
            ; return from defn
#<FUNCTION HALF ...>
```

Lambda is a Special Form

- Lambda is like a Macro in that it doesn't evaluate its parameters first
- However, the actual value that lambda returns is a regular Lisp function!
- Functions with lambda instead of a name are called anonymous functions

```
>(mapcar
     (lambda (n)
            (/ n 2))
            '(2 4 6))
(1 2 3)
```

Why Lambda is So Important

- The ability to pass around functions as if they were just plain old pieces of data is incredibly valuable.
- This opens up all kinds of conceptual possibilities in the design of your programs!
- The name for the *style of programming* that relies heavily on passing functions as values is called ***higher-order functional programming.***

Lambda Summary

- By using `lambda`, you can create a function *without a name*.
- Many functions in Lisp *accept functions* as parameters or *return functions* as the result or may do *both*.
- When you use these functions, you are using the **higher-order functional programming** technique.