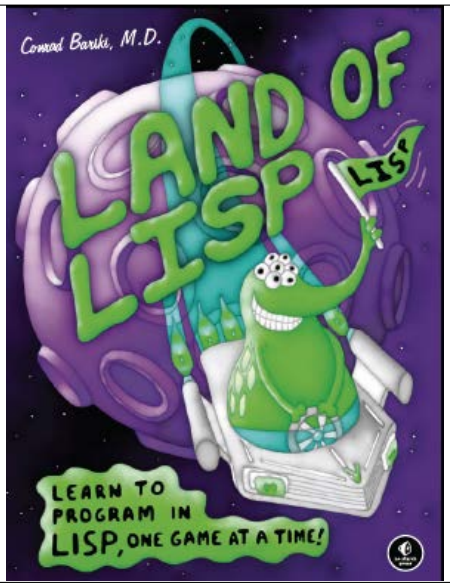


Ch. 1-2

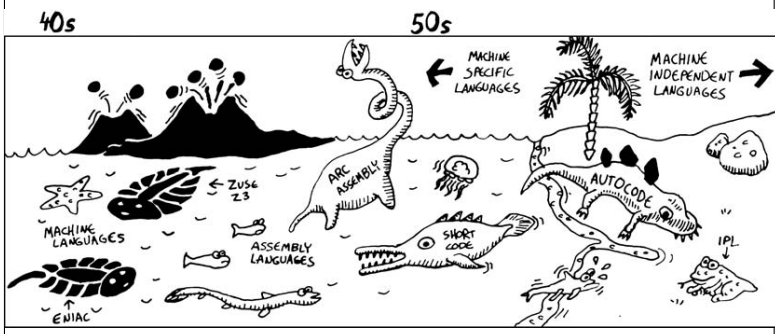
Why Lisp?

Lisp gives you insights into computer programming that are so profound that *every serious programmer should have some experience with this unusual language*, even if it requires a little effort.

What Makes Lisp so Cool and Unusual?

- Lisp is a very ***expressive language***. *Lisp is designed to let you take the most complicated programming ideas* and express them in a clear and appropriate way.
- Lispers have the freedom to write a program in exactly the way that is most helpful for solving any problem at hand.

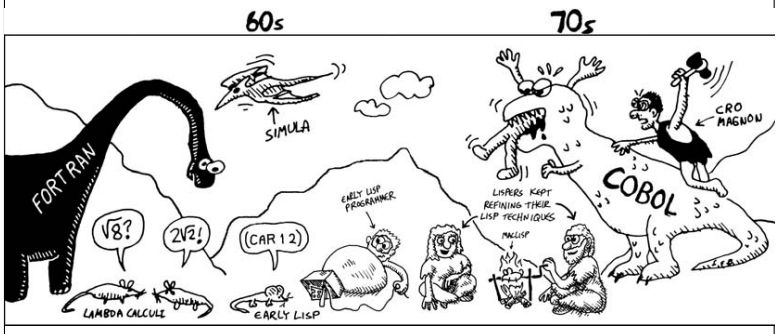
Machine Language, Assembly Languages, Machine Specific Languages



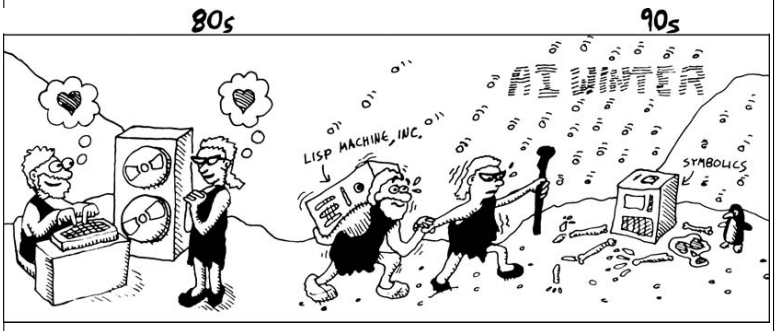
Machine Independent Languages

- For the first time, programmers could use languages that were *designed to make computer programming a pleasant activity*, without needing to operate at the primitive level of the computer hardware.

Machine Independent Languages, Fortran, Lisp, COBOL

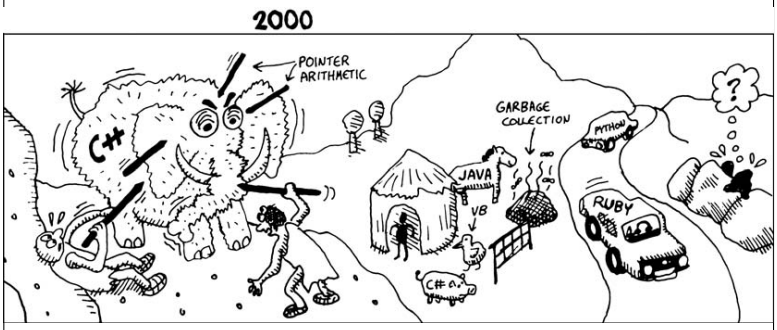


Specialized Hardware for Lisp

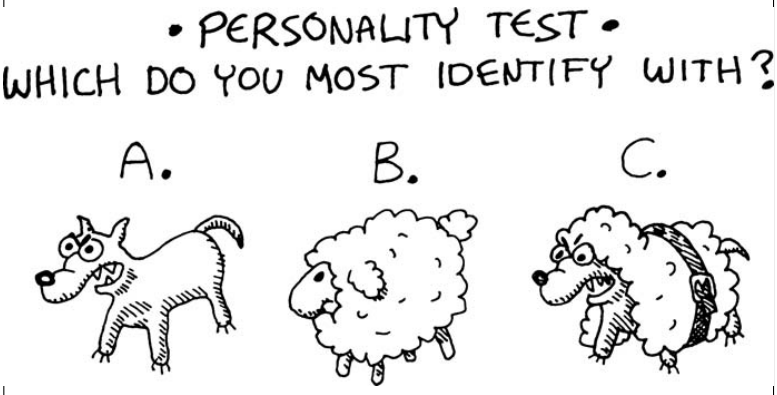


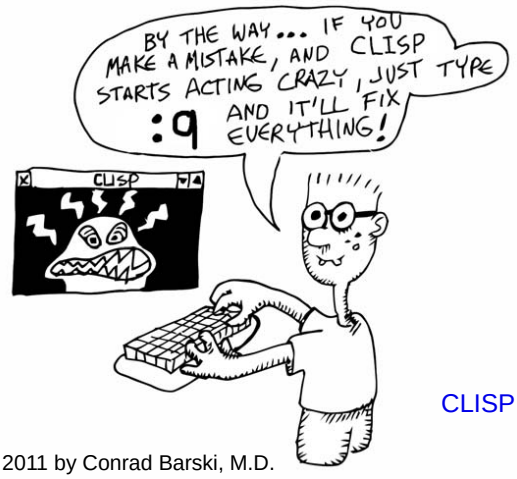
Early Lispers …

Programming Paradigms, New Languages



Lisp and Scheme Power and Elegance





CLISP

Creating Your First Lisp Program

- Guess my number game
- Choose a number between 1 and 100
- Computer keeps guessing until the correct number is guessed

```
guess-my-number function
bigger
smaller
start-over
```


Guess My Number Algorithm

1. Computer starts by initializing the upper and lower limit of the player's number. In this case, the smallest number is 1 and the biggest is 100.
2. Computer guesses a number in between the high and low numbers (we'll use binary search (why?))
3. If
   1. the guess is correct, you're DONE. (Call `start-over` to reset high/low for a new game.)
   2. the real number is smaller than the computer's guess, the player calls the `smaller` function that lowers the big limit.
   3. the real number is bigger than the guess, the player calls the `bigger` function that raises the small limit.

- Note: Doesn't include error checking or incorrect user answers.

Guess.lisp

guess.lisp

```
(defparameter *small* 1)
(defparameter *big* 100)

(defun guess-my-number ()
     (ash (+ *small* *big*) -1))

(defun smaller ()
     (setf *big* (1- (guess-my-number)))
     (guess-my-number))

(defun bigger ()
     (setf *small* (1+ (guess-my-number)))
     (guess-my-number))

(defun start-over ()
   (defparameter *small* 1)
   (defparameter *big* 100)
   (guess-my-number))
```


Defining Global Variables in Lisp

defparameter

```
> (defparameter
*foo* 5)
*FOO*
> *foo*
5
> (defparameter
*foo* 6)
*FOO*
> *foo*
6
```

defvar

```
> (defvar *foo* 5)
*FOO*
> *foo*
5
> (defvar *foo* 6)
*FOO*
> *foo*
5
```

Defvar doesn't overwrite the current value!!!!

Defining Global Functions in Lisp

- `defun`

(**defun** *function_name* (*arguments*) ...)

- Example

```
> (defun guess-my-number ()
     (ash (+ *small* *big*) -1))
 GUESS-MY-NUMBER
```

The built-in Lisp function **ash** looks at a number in *binary* form, and then *shifts* its binary bits to the left or right, dropping any bits lost in the process.

What does ash do in the function guess-my-number?

Functions Smaller and Bigger

```
> (defun smaller ()
    (setf *big*
          (1- (guess-my-number)))
    (guess-my-number))
SMALLER
> (defun bigger ()
    (setf *small*
          (1+ (guess-my-number)))
    (guess-my-number))
BIGGER
```


Defining Local Variables in Lisp

- `let`
- `let*`

```
(let (variable declarations)
     ...body...)
> (let ((a 5)
        (b 6))
       (+ a b))
11
```


Defining Local Functions in Lisp

- `flet`
- `lambda`

```
(flet ((function_name
                (arguments)
          ...function body...))
             ...body...)
> (flet ((f (n)
           (+ n 10)))
         (f 5))
15
```


labels

- The `labels` command lets you call one local function from another, and it allows you to have a function call itself.
- The syntax is like `flet`.

```
> (labels ((a (n)
             (+ n 5))
           (b (n)
             (+ (a n) 6)))
         (b 10))
21
```


Summary

- Why Lisp?
- What makes Lisp so cool?
- A brief evolution of programming languages
- Lisp is Power
- Guess my number game
- Defining global variables and functions
- Defining local variables and functions