Lisp Cheatsheet

Lisp is a family of programming languages that are based on the lambda calculus and characterized by their use of parentheses for code blocks. Lisp is a powerful language with a simple syntax, making it a popular choice for artificial intelligence, data processing, and web development.

Unique Features

- Code as data
- Macros
- Dynamic typing
- Recursion
- Functional programming
- Interactive development

Variables

Variables in Lisp are declared using the `defvar` or `setq` function. Lisp supports dynamic typing, so you don't need to specify the type of the variable.

```
(defvar name "John")
(setq age 30)
(setq pi 3.14)
```

Functions

Functions in Lisp are declared using the `defun` function followed by the function name and parameters. Lisp supports lambda expressions, which are anonymous functions that can be assigned to variables and passed as arguments to other functions.

```
(defun greet (name)
  (format t "Hello, ~a!" name))

(greet "John")

(setq add (lambda (a b)
  (+ a b)))

(print (funcall add 2 3))
```

Loops

Lisp supports `do`, `dotimes`, and `dolist` loops, as well as recursion.

```
(setq numbers '(1 2 3 4 5))

(dolist (number numbers)
  (print number))
```

```
(dotimes (i 5)
  (print i))

(defun countdown (n)
  (when (> n 0)
    (print n)
    (countdown (- n 1))))

(countdown 5)
```

Conditionals

Lisp supports `if`, `when`, and `unless` statements, as well as the `cond` control structure.

```
(setq age 30)

(if (< age 18)
  (print "You are too young to vote.")
  (if (< age 21)
    (print "You can vote, but not drink.")
    (print "You can vote and drink.")))

(setq result (if (>= age 18) "You are an adult" "You are not an adult"))
(print result)
```

File Manipulation

Lisp provides several ways to read and write files. You can use the `with-open-file` macro to create, read, write, and delete files.

```
(with-open-file (file "example.txt" :direction :output)
  (write-line "Hello, world!" file))

(with-open-file (file "example.txt" :direction :input)
  (let ((content (read-line file)))
    (print content)))

(delete-file "example.txt")
```

Resources

- Common Lisp HyperSpec
- Practical Common Lisp
- Lisp Koans
- Lisp REPL