# Lisp Supplemental: Learning Lisp with the

# Adaptive Remote Agent

---

This document is designed to help you learn the fundamentals of Lisp and experiment with the Adaptive Remote Agent available on the Web:

`http://art2.ph-freiburg.de/Lisp-Course`

# 1 Running the Adaptive Remote Agent

In this document, you are requested to login into the Adaptive Remote Agent. The Adaptive Remote Agent was developed in Germany, and is made available for free for the benefit of the AI and Lisp community. Some language problems may persist, and the system is 'old' in terms of 'fashionable' interfaces. Those problems are known, and cannot be fixed. Please be patient about them, and focus on taking advantage of the good features of the software.

Log-in to the agent using the URL above and complete lessons 1-4.

# 2 First Steps in Common Lisp

Complete the following 5 exercises. All lisp functions should be done using ACL in emacs.

1. **Is the following an atom, list, both, or neither?**

   (a) ATOM
   (b) (rest '(1 2 3))
   (c) )(
   (d) (rest '())
   (e) ()

2. **Math** Evaluate the following functions.

   (a) (/ (+ 5 7) (- 1 4))
   (b) (+ (* 3 4) (* 5 (first '(0 1 2))))
   (c) (MAX 3 (MIN 5 2 8))

3. **First/Rest**

   Write the sequence of first's and rest's to get the number 3 from the list x. You may also use compound car/cdr's.

   > **Example:** (setf x '(1 2 3 4))
   > *Answer:* (first (rest (rest x))) or (caddr x)

   (a) (setf x '(1 (2 (3 (4)))))
   (b) (setf x '((1 2) (3 4)))
   (c) (setf x '(1 (2 3) 4))
   (d) (setf x '((((1) 2) 3) 4))

4. **Cond**

   Define a function CHARACTERIZE-SIZE that takes a positive number $n$ as an argument and returns SMALL if $n < 10$, MEDIUM if $10 \leq n < 50$, and BIG if $n \geq 50$

5. **Functions**

   (a) NEGATIVE-P

   Define a predicate function NEGATIVE-P that takes a number as its argument and returns **t** if the number is negative.

   (b) X5

   Define a function X5 that takes one argument, tests whether the argument is a number, and returns the number times 5 otherwise returns the argument itself.

   (c) MANIPULATE

   Define a function MANIPULATE that takes a list of numbers, and multiplies each negative number by 5, leaves every non-negative number as is, and returns the modified list.