CS 61A Week 5

Topic: Hierarchical data

Midterm Wednesday , 7–9pm.

Reading: Abelson & Sussman, Section 2.2.2–2.2.3, 2.3.1, 2.3.3

Homework:

• Abelson & Sussman, exercises 2.24, 2.26, 2.29, 2.30, 2.31, 2.32, 2.36, 2.37, 2.38, 2.54.

Some of these exercises are harder than they look; don't give up in frustration if your early attempts fail.

• Extend the calculator program from lecture to include words as data, providing the operations `first`, `butfirst`, `last`, `butlast`, and `word`. Unlike Scheme, **your calculator should treat words as self-evaluating expressions** except when seen as the operator of a compound expression. That is, it should work like these examples:

```
calc: foo
foo
calc: (first foo)
f
calc: (first (butfirst hello))
e
```

The program is in `~cs61a/lib/calc.scm`

Note: Programming project 2 is also due next week. It consists of all the exercises in Section 2.2.4 of the text. You can't actually draw anything until you finish the project! To begin, copy the file `~cs61a/lib/picture.scm` to your directory. To draw pictures, once you've completed the exercises:

```
> (cs)
> (ht)
> (===your-painter=== full-frame)
```

For example:

```
> (wave full-frame)
> ((square-limit wave 3) full-frame)
```

Continued on next page.

Week 5 continued...

Extra for experts:

Read section 2.3.4 and do exercises 2.67–2.72.

Programming by example: In some programming systems, instead of writing an algorithm, you give examples of how you'd like the program to behave, and the language figures out the algorithm itself:

```
> (define pairup (regroup '((1 2) (3 4) ...)))
> (pairup '(the rain in spain stays mainly on the plain))
((the rain) (in spain) (stays mainly) (on the))
```

Write `regroup`. Read `~cs61a/lib/regroup.problem` for details.

Unix feature of the week: `head`, `tail`, `more`, `cat`

Emacs feature of the week: `M-x search-forward-regexp`, `M-x query-replace-regexp`