

# pudding Eater <sup>1</sup>

Eric Bailey

October 10, 2017 <sup>2</sup>

Who ate my pudding?

## Defining the Arch-Enemy Variable

Since at first we don't know who the pudding eater (a.k.a. our `*arch-enemy*`) is, set the initial value to `nil`.

```
1b <* 1a>+≡
    (defvar *arch-enemy* nil)
```

Defines:

`*arch-enemy*`, used in chunks 1 and 2c.

## Defining the Pudding-Eater Function

This chapter introduces `cond`, an extremely versatile function that's "been around since the Lisp Stone Age." The basic form is as follows.

```
(cond (test-form form*)
      ...)
```

Since `test-forms` are evaluated for truthiness, we can branch on which `person` ate our pudding.

If *(it was Henry 1c)*, the Lisp alien,

```
1d <blame Henry. 1d>≡
    (setf *arch-enemy* 'stupid-lisp-alien)
    '(curse you lisp alien - you ate my pudding)
```

This code is used in chunk 2.

Uses `*arch-enemy*` 1b.

If *(it was Johnny 1e)*,

```
1f <blame Johnny. 1f>≡
    (setf *arch-enemy* 'useless-old-johnny)
    '(i hope you choked on my pudding johnny)
```

This code is used in chunk 2.

Uses `*arch-enemy*` 1b.

If it was someone else, *(ask them why. 1g)*.

<sup>1</sup>

Conrad Barski. *Land of Lisp: Learn to Program in Lisp, One Game at a Time!*, chapter 4, pages 49–66. No Starch Press, 2010. ISBN 9781593273491. URL <http://landoflisp.com>

<sup>2</sup> Last updated October 18, 2017

src/pudding.lisp:

```
1a <* 1a>≡
    (in-package :cl-user)
    (defpackage lol.pudding
      (:use :cl :prove)
      (:export pudding-eater))
    (in-package :lol.pudding)
```

This definition is continued in chunks 1b and 2b.

Root chunk (not used in this document).

Defines:

`lol.pudding`, used in chunk 2c.

Uses `pudding-eater` 2b.

"Global variable names should start and end with asterisks (also known in this context as earmuffs)" [Brown and Rideau, 2017].

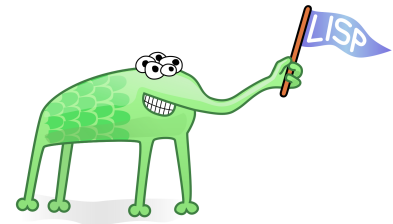


Figure 1: Henry, the Lisp Alien

```
1c <it was Henry 1c>≡
    (eq person 'henry)
```

This code is used in chunk 2a.

```
1e <it was Johnny 1e>≡
    (eq person 'johnny)
```

This code is used in chunk 2a.

```
1g <ask them why. 1g>≡
    '(why you eat my pudding stranger ?)
```

This code is used in chunk 2.

The `cond` version of `pudding-eater` then, would look like this:

```
2a <cond-flavoured pudding-eater 2a>≡
  (λ (person)
    (cond (<it was Johnny 1e> <blame Johnny. 1f>)
          (<it was Henry 1c> <blame Henry. 1d>)
          (t <ask them why. 1g>))))
```

Root chunk (not used in this document).

More succinctly, with `case`, we can define the `pudding-eater` function.

```
2b <* 1a>+≡
  (defun pudding-eater (person)
    (case person
      ((henry) <blame Henry. 1d>)
      ((johnny) <blame Johnny. 1f>)
      (otherwise <ask them why. 1g>))))
```

Defines:

`pudding-eater`, used in chunks [1a](#) and [2c](#).

## Tests

```
2c <test/pudding.lisp 2c>≡
  (in-package :lol.pudding)

  (plan 1)

  (subtest "A Plausible Session"
    (is (pudding-eater 'johnny)
      '(I HOPE YOU CHOKED ON MY PUDDING JOHNNY)
      "I hope you choked on my pudding, Johnny!")
    (is *arch-enemy*
      'USELESS-OLD-JOHNNY
      "Useless old Johnny!")
    (is (pudding-eater 'george-clooney)
      '(WHY YOU EAT MY PUDDING STRANGER ?)))

  (finalize)
```

Root chunk (not used in this document).

Uses `*arch-enemy*` [1b](#), `lol.pudding` [1a](#), and `pudding-eater` [2b](#).

## Glossary

*empty list* the list containing no elements. 3

`nil` represents both boolean `false` and the *empty list*. Alternatively notated as `()` to emphasize its use as an *empty list*. 1, 3

## References

Conrad Barski. *Land of Lisp: Learn to Program in Lisp, One Game at a Time!*, chapter 4, pages 49–66. No Starch Press, 2010. ISBN 9781593273491. URL <http://landoflisp.com>.

Robert Brown and François-René Rideau. Google Common Lisp Style Guide: Global variables and constants. [https://google.github.io/styleguide/lispguide.xml?showone=Global\\_variables\\_and\\_constants#Global\\_variables\\_and\\_constants](https://google.github.io/styleguide/lispguide.xml?showone=Global_variables_and_constants#Global_variables_and_constants), September 2017. Accessed: 2017-10-08.

Kent M. Pitman. CLHS: Glossary. [http://www.lispworks.com/documentation/HyperSpec/Body/26\\_a.htm](http://www.lispworks.com/documentation/HyperSpec/Body/26_a.htm), April 2005. Accessed: 2017-10-17.