Pudding Eater [1]

Eric Bailey

October 10, 2017 [2]

[1] 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

[2] Last updated October 11, 2017

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.

1 ⟨*src/pudding.lisp 1*⟩≡
```
(defvar *arch-enemy* nil)
```
This definition is continued in chunk 8.
Root chunk (not used in this document).
Defines:
arch-enemy, used in chunks 3 and 5.

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

Defining the Pudding-Eater Function

This chapter introduces cond, an extremely versatile function that's "been around sine 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 2*⟩, the Lisp alien, ⟨*blame Henry 3*⟩;
If ⟨*it was Johnny 4*⟩, ⟨*blame Johnny 5*⟩.
If it was someone else, ⟨*ask them why 6*⟩.
The cond version of pudding-eater then, would look like this:

7 ⟨cond-*flavoured* pudding-eater 7⟩≡
```
(λ (person)
  (cond (⟨it was Johnny 4⟩  ⟨blame Johnny 5⟩)
        (⟨it was Henry 2⟩   ⟨blame Henry 3⟩)
        (t                  ⟨ask them why 6⟩))))
```
Root chunk (not used in this document).

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

8 ⟨*src/pudding.lisp 1*⟩+≡
```
(defun pudding-eater (person)
  (case person
        ((henry)    ⟨blame Henry 3⟩)
        ((johnny)   ⟨blame Johnny 5⟩)
        (otherwise ⟨ask them why 6⟩)))
```
Defines:
pudding-eater, never used.

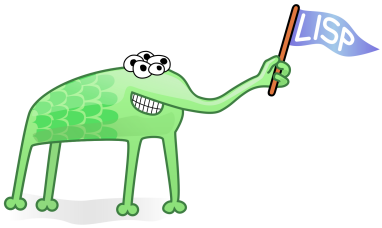

Figure 1: Henry, the Lisp Alien

2 ⟨*it was Henry 2*⟩≡
```
(eq person 'henry)
```
This code is used in chunk 7.

3 ⟨*blame Henry 3*⟩≡
```
(setf *arch-enemy* 'stupid-lisp-alien)
'(curse you lisp alien - you ate my pudding)
```
This code is used in chunks 7 and 8.
Uses *arch-enemy* 1.

4 ⟨*it was Johnny 4*⟩≡
```
(eq person 'johnny)
```
This code is used in chunk 7.

5 ⟨*blame Johnny 5*⟩≡
```
(setf *arch-enemy* 'useless-old-johnny)
'(i hope you chocked on my pudding johnny)
```
This code is used in chunks 7 and 8.
Uses *arch-enemy* 1.

6 ⟨*ask them why 6*⟩≡
```
'(why you eat my pudding stranger ?)
```
This code is used in chunks 7 and 8.

Full Listing

⟨*src/pudding.lisp* 1⟩:

```
(defvar *arch-enemy* nil)
(defun pudding-eater (person)
  (case person
        ((henry)   (setf *arch-enemy* 'stupid-lisp-alien)
                   '(curse you lisp alien - you ate my pudding))
        ((johnny)  (setf *arch-enemy* 'useless-old-johnny)
                   '(i hope you chocked on my pudding johnny))
        (otherwise '(why you eat my pudding stranger ?))))
```

Example Session

After loading ⟨*src/pudding.lisp* 1⟩, you might have a session like this:

```
$ rlwrap sbcl --load src/pudding.lisp
```

```
> (pudding-eater 'johnny)
(I HOPE YOU CHOKED ON MY PUDDING JOHNNY)
> *arch-enemy*
USELESS-OLD-JOHNNY
> (pudding-eater 'george-clooney)
(WHY YOU EAT MY PUDDING STRANGER ?)
```

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, September 2017. Accessed: 2017-10-08.