

A Quick Macro for Pollen

SATURDAY, JULY 7TH, 2018

One of Pollen's somewhat recent¹ additions is the `for/splice`² function, which lets you loop through one or more lists and inject the results into a surrounding expression.

Here's an example from the code for the home page on this blog:

```
index.html.pp
1 ...␣
2 for/splice[[ (post (in-list (latest-posts 10))) ]]{
3   <article>␣
4   (hash-ref post 'header_html)␣
5   (hash-ref post 'html)
6   </article>
7   <hr>
8 }
9 ...
```

It's very useful, but I'm sure you've noticed something about this code: *many many parentheses* in the arguments list for `for/splice`. Well, it is the way it is, because the `for/splice` function is a direct analogue of Racket's `for/list` function³, which supports looping through multiple lists at the same time. It's nice to have the flexibility. But there's no denying the readability of the code suffers in the most common use cases.

If, like me, you usually only need to loop through a single list, you can put a macro like the one below in your `pollen.rkt`:

```
pollen.rkt
1 (require pollen/core)
2
3 ...
4
5 (provide for/s)
6 (define-syntax (for/s stx)
7   (syntax-case stx ()))
```

1. https://groups.google.com/d/msg/pollenpub/FdWdGgL9jSk/1VWpX_nsAAAJ
2. http://docs.racket-lang.org/pollen/Core.html#%28form._%28%28lib._pollen%2Fcore..rkt%29._for%2Fsplice%29%29
3. https://docs.racket-lang.org/reference/for.html?q=for%2Flist#%28form._%28%28lib._racket%2Fprivate%2Fbase..rkt%29._for%2Flist%29%29

```
8  [(_ thing listofthings result-expr ...)  
9  #'(for/splice ([thing (in-list listofthings)]) result-expr ...)))]))
```

This cuts down on the parentheses quite a bit:

```
index.html.pp  
1  ...  
2  for/s[post (latest-posts 10)]{  
3    <article>  
4      (hash-ref post 'header_html)  
5      (hash-ref post 'html)  
6    </article>  
7    <hr>  
8  }  
9  ...
```