Simulating Common Lisp Macros in Python: Higher Order Functions

November 26, 2008

I'm using the feedparser library to extract data from rss feed items.

After I wrote this function, which returns a list of item titles, I noticed that most item attributes would be retrieved the same way, i.e., the function would look exactly the same, except for the single data.append line inside the for loop.

```
def item_titles (feed_url):
    """Return a list of the item titles found in this feed url"""
    data = []
    feed = feedparser.parse(feed_url)
    if feed:
        if len(feed.version) > 0:
            for e in feed.entries:
                data.append(e.title.encode('utf-8'))
    return data
```

In Common Lisp, I could simply write a macro, then replace the data.append line depending on which attribute I wanted.

I wondered if there was anything similar in Python, and found that I could use higher-order functions (thanks, Chris), like this:

```
def _read_item_data (feed_url, get_fn):
    """Return a list of item data (specified by the get_fn) found in this feed url"""
    data = []
    feed = feedparser.parse(feed_url)
    if feed:
        if feed.version and len(feed.version) > 0:
            for e in feed.entries:
                try:
                    data.append(get_fn(e))
                except AttributeError, detail:
                     data.append(None)
    return data
```

So for the item titles, all I need to call is this:

```
def item_titles (feed_url):
    """Return a list of the item titles found in this feed url"""
    return _read_item_data(feed_url, lambda x: normalize_string(x.title))
```

And for the item links, I can do this:

```
def item_links (feed_url):
    """Return a list of the item links found in this feed url"""
    return _read_item_data(feed_url, lambda x: normalize_string(x.link))
```

And so on, where the normalize_string() function takes care of any encoding and other formatting.

Archived from the original at http://denis.papathanasiou.org/
Bitcoin Donate: 14TM4ADKJbaGEi8Qr8dh4KfPBQmjTshkZ2