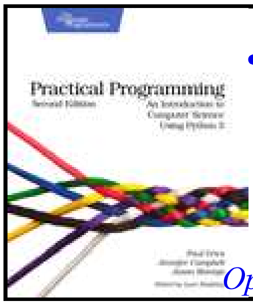
Parallel Lists and Strings



- Chapter 9.4 Processing Lists Using Indices

Optional reading

Correspondings Elements

Two lists are *parallel* if they have the same length and the items at each index are somehow related. The items at the same index are said to be at *corresponding positions*.

Consider these two lists:

```
list1 = [1, 2, 3]
list2 = [2, 4, 2]
```

In these two lists, the corresponding element of `list1[0]` is `list2[0]`, the corresponding element of `list2[1]` is `list1[1]`, and so on.

Example of Corresponding Elements

```
def match_characters(s1, s2):
    ''' (str, str) -> int

    Return the number of characters in s1 that are the same as the character
    at the corresponding position of s2.

    Precondition: len(s1) == len(s2)

    >>> match_characters('ate', 'ape')
    2
    >>> match_characters('head', 'hard')
    2
    '''

    num_matches = 0

    for i in range(len(s1)):
        if s1[i] == s2[i]:
            num_matches = num_matches + 1

    return num_matches
```

The function above counts the corresponding elements of the two strings that are the same character. If a character of `s1` at index `i` is the same as the character of `s2` at the same index, then we increment `num_matches` by 1 (since they match). Otherwise, we continue on to the next pair of corresponding elements and compare them.

Jennifer Campbell • Paul Gries
University of Toronto