

تست توابع داخلی

برای این تست، باید از توابع داخلی استفاده کرده و توانایی نوشتن توابع درخواستی را در یک خط داشته باشید.

مسئله ۱

از تابع `map()` برای ایجاد یک تابع استفاده کنید که طول هر کلمه را در عبارت (با فاصله‌ها تفکیک شده) پیدا کرده و مقادیر را در یک لیست برگرداند.

ورودی تابع یک رشته است و خروجی یک لیست اعداد صحیح است.

In [1]:

```
1 def word_lengths(phrase):
2
3     words = phrase.split()
4
5     return list( map(len, words) )
```

In [2]:

```
1 word_lengths('How long are the words in this phrase')
```

Out[2]:

```
[3, 4, 3, 3, 5, 2, 4, 6]
```

مسئله ۲

از تابع `reduce()` برای گرفتن یک لیست از ارقام استفاده کنید و عددی که با آن مطابقت دارند را برگردانید. به عنوان مثال، [3، 2، 1] متناظر با صد و بیست و سه است. اعداد را به رشته تبدیل نکنید!

In [3]:

```
1 from functools import reduce
2
3 def digits_to_num(digits):
4
5     return reduce( lambda x, y: x * 10 + y, digits)
```

In [4]:

```
1 digits_to_num([3,4,3,2,1])
```

Out[4]:

34321

مسئله 3

از تابع فیلتر برای استخراج کلماتی که با حرف اول مشخص شده شروع شده اند استفاده نمایید.

In [8]:

```
1 def filter_words(word_list, letter):
2
3     return list( filter(lambda word: word[0] == letter, word_list ))
```

In [11]:

```
1 l = ['hello', 'are', 'cat', 'dog', 'ham', 'hi', 'go', 'to', 'heart']
2 filter_words(l, 'c')
```

Out[11]:

['cat']

Problem 4

از تابع `zip()` و یک `list comprehension` استفاده کنید تا یک لیست با همان طول تولید شود که هر مقدار آن دو رشته از `L1` و `L2` است که با رابطه بین آنها ادغام شده‌اند. نمونه خروجی را در زیر ببینید:

In [12]:

```
1 def concatenate(L1, L2, connector):
2
3     return [word1 + connector + word2 for word1, word2 in zip(L1, L2) ]
```

In [14]:

```
1 concatenate(['A', 'B', 'C'], ['a', 'b', 'c'], '-+-')
```

Out[14]:

['A+-a', 'B+-b', 'C+-c']

Problem 5

از تابع `enumerate()` و مهارت‌های دیگر برای تولید یک دیکشنری استفاده کنید که مقادیر لیست را به عنوان کلیدها و شاخص را به عنوان مقدار داشته باشد. می‌توانید فرض کنید که هر مقدار تنها یک بار در لیست داده شده ظاهر می‌شود.

In [20]:

```
1 def d_list(L):
2
3     return {key: value for value, key in enumerate(L)}
```

In [21]:

```
1 d_list(['a','b','c','d','e'])
```

Out[21]:

```
{'a': 0, 'b': 1, 'c': 2, 'd': 3, 'e': 4}
```

Problem 6

از تابع `enumerate()` و مهارت‌های دیگر از بالا استفاده کنید تا تعداد مواردی را که مقدار آنها با شاخصشان برابر است، بشمارید.

In [24]:

```
1 def count_match_index(L):
2
3     return len([num for index, num in enumerate(L) if num == index])
```

In [25]:

```
1 count_match_index([0,2,2,1,5,5,6,10])
```

Out[25]:

4

عالی بود!