Week1. Quiz

2020年3月17日 星期二　　下午11:32

Map() function

Here is a list of faculty teaching this MOOC. Can you write a function and apply it using map() to get a list of all faculty titles and last names (e.g. ['Dr. Brooks', 'Dr. Collins-Thompson', ...]) ?

```
1  people = ['Dr. Christopher Brooks', 'Dr. Kevyn Collins-Thompson', 'Dr. VG
     Vinod Vydiswaran', 'Dr. Daniel Romero']
2
3  def split_title_and_name(person):
4      title = person.split()[0]
5      lname = person.split()[-1]
6      return '{}{}'.format(title, lname)
7
8  list(map(split_title_and_name, people))
9
```

Run

Reset

```
['Dr.Brooks', 'Dr.Collins-Thompson', 'Dr.Vydiswaran', 'Dr.Romero']
```

List comprehensions ①

Here's a solution:

```
1  def times_tables():
2      lst = []
3      for i in range(10):
4          for j in range (10):
5              lst.append(i*j)
6      return lst
7
8  times_tables() == [j*i for i in range(10) for j in range(10)]
9
```

Run

Reset

List comprehension ②

Here's a solution:

```
1  lowercase = 'abcdefghijklmnopqrstuvwxyz'
2  digits = '0123456789'
3
4  correct_answer = [a+b+c+d for a in lowercase for b in lowercase for c in digits
       for d in digits]
5
6  correct_answer[:50] # Display first 50 ids
7
```

Run

Reset

```
['aa00', 'aa01', 'aa02', 'aa03', 'aa04', 'aa05', 'aa06', 'aa07', 'aa08', 'aa09',
 'aa10', 'aa11', 'aa12', 'aa13', 'aa14', 'aa15', 'aa16', 'aa17', 'aa18', 'aa19',
 'aa20', 'aa21', 'aa22', 'aa23', 'aa24', 'aa25', 'aa26', 'aa27', 'aa28', 'aa29',
 'aa30', 'aa31', 'aa32', 'aa33', 'aa34', 'aa35', 'aa36', 'aa37', 'aa38', 'aa39',
 'aa40', 'aa41', 'aa42', 'aa43', 'aa44', 'aa45', 'aa46', 'aa47', 'aa48', 'aa49']
```

Lambda function

Here's a solution:

```
1  people = ['Dr. Christopher Brooks', 'Dr. Kevyn Collins-Thompson', 'Dr. VG
       Vinod Vydiswaran', 'Dr. Daniel Romero']
2
3  def split_title_and_name(person):
4      return person.split()[0] + ' ' + person.split()[-1]
5
6  #option 1
7  for person in people:
8      print(split_title_and_name(person) == (lambda x: x.split()[0] + ' ' + x
           .split()[-1])(person))
9
10 #option 2
11 list(map(split_title_and_name, people)) == list(map(lambda person: person
       .split()[0] + ' ' + person.split()[-1], people))
```

Run

Reset