



# Самое необходимое о классах: ДЗ






## Условие

Реализуйте класс [CountVectorizer](#), имеющий

- метод [fit transform](#)
- метод [get feature names](#)

Условия:

- пользоваться внешними пакетами запрещено
- решение должно быть в \*.py файлах
- не должно быть замечаний PEP8
- решение загружено на github 
- если умеешь, то напиши проверки/тесты

Далее на слайдах пример получения [терм-документной матрицы](#)



## Исходные тексты

### Crock Pot Pasta

Never boil pasta again



### Pasta Pomodoro

Fresh ingredients Parmesan to taste





## Подсчитываем кол-во слов

### Crock Pot Pasta

Never boil pasta again

### Pasta Pomodoro

Fresh ingredients Parmesan to taste

### Список слов

0	crock	7	fresh
1	pot	8	ingredients
2	pasta	9	parmesan
3	never	10	to
4	boil	11	taste
5	again		
6	pomodoro		



## Строим матрицу

### Crock Pot Pasta

Never boil pasta again

### Pasta Pomodoro

Fresh ingredients Parmesan to taste

#### Список слов

0	crock	7	fresh
1	pot	8	ingredients
2	pasta	9	parmesan
3	never	10	to
4	boil	11	taste
5	again		
6	pomodoro		

#### Вектора

1 [1, 1, 2, 1, 1, 1, 0, 0, 0, 0, 0, 0]  
2 [0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1]



## Пример использования

Реализуйте класс `CountVectorizer`, имеющий метод `fit_transform`

```
1 corpus = [  
2     'Crock Pot Pasta Never boil pasta again',  
3     'Pasta Pomodoro Fresh ingredients Parmesan to taste'  
4 ]  
5  
6 vectorizer = CountVectorizer()  
7 count_matrix = vectorizer.fit_transform(corpus)  
8 print(vectorizer.get_feature_names())  
9 Out: ['crock', 'pot', 'pasta', 'never', 'boil', 'again', 'pomodoro',  
10      'fresh', 'ingredients', 'parmesan', 'to', 'taste']  
11  
12 print(count_matrix)  
13 Out: [[1, 1, 2, 1, 1, 1, 0, 0, 0, 0, 0, 0],  
14      [0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1]]
```



Спасибо за проделанную работу!