

Доброе
изолированное
утро



Программа

1. Быстро повторим методы регулярок
2. Посмотрим на именованные группы
3. Разберем `re.split`
4. Узнаем, кто такие `list-comprehensions`
5. Решим интересную задачу

Методы регулярок (в тетрадке)

Разбиения в регулярных выражениях

Разбиение текста на отдельные слова

```
def get_tokens(text):
```

```
    tokens = text.split()
```

```
    # Долгая чистка всякого мусора
```

```
    return tokens
```

Разбиение текста на отдельные слов

```
def get_tokens(text):  
    tokens = re.split('\W+', text)  
    return tokens
```

```
re.split(pattern, string, maxsplit=0, flags=0)
```

maxsplit

```
text = 'приветливое приведение с причудами'
```

```
print(re.split('\W+', text))
```

```
print(re.split('\W+', text, maxsplit=1))
```

```
['приветливое', 'приведение', 'с', 'причудами']
```

```
['приветливое', 'приведение с причудами']
```


Антоним findall

```
text = 'Иван Родил Девчонку, Велел Тащить Пелёнку'
```

```
print(re.findall('\w+', text))
```

```
print(re.split('\W+', text))
```

```
['Иван', 'Родил', 'Девчонку', 'Велел', 'Тащить', 'Пелёнку']
```

```
['Иван', 'Родил', 'Девчонку', 'Велел', 'Тащить', 'Пелёнку']
```

Группы сохраняются

```
expr = '3 + 2 * 5 / 33 - 2'
```

```
print(re.split('\s*[\+*\-/\]\s*', expr))
```

```
print(re.split('\s*([\+*\-/\])\s*', expr))
```

```
['3', '2', '5', '33', '2']
```

```
['3', '+', '2', '*', '5', '/', '33', '-', '2']
```

```
text = 'ЯШелВ7ЧасовВечераДомой'  
result = re.split('([0-9]+|[А-Я][а-я]*)', text)  
print(result)
```

Что будет напечатано?

List comprehensions (в тетрадке)

Задача (в тетрадке)