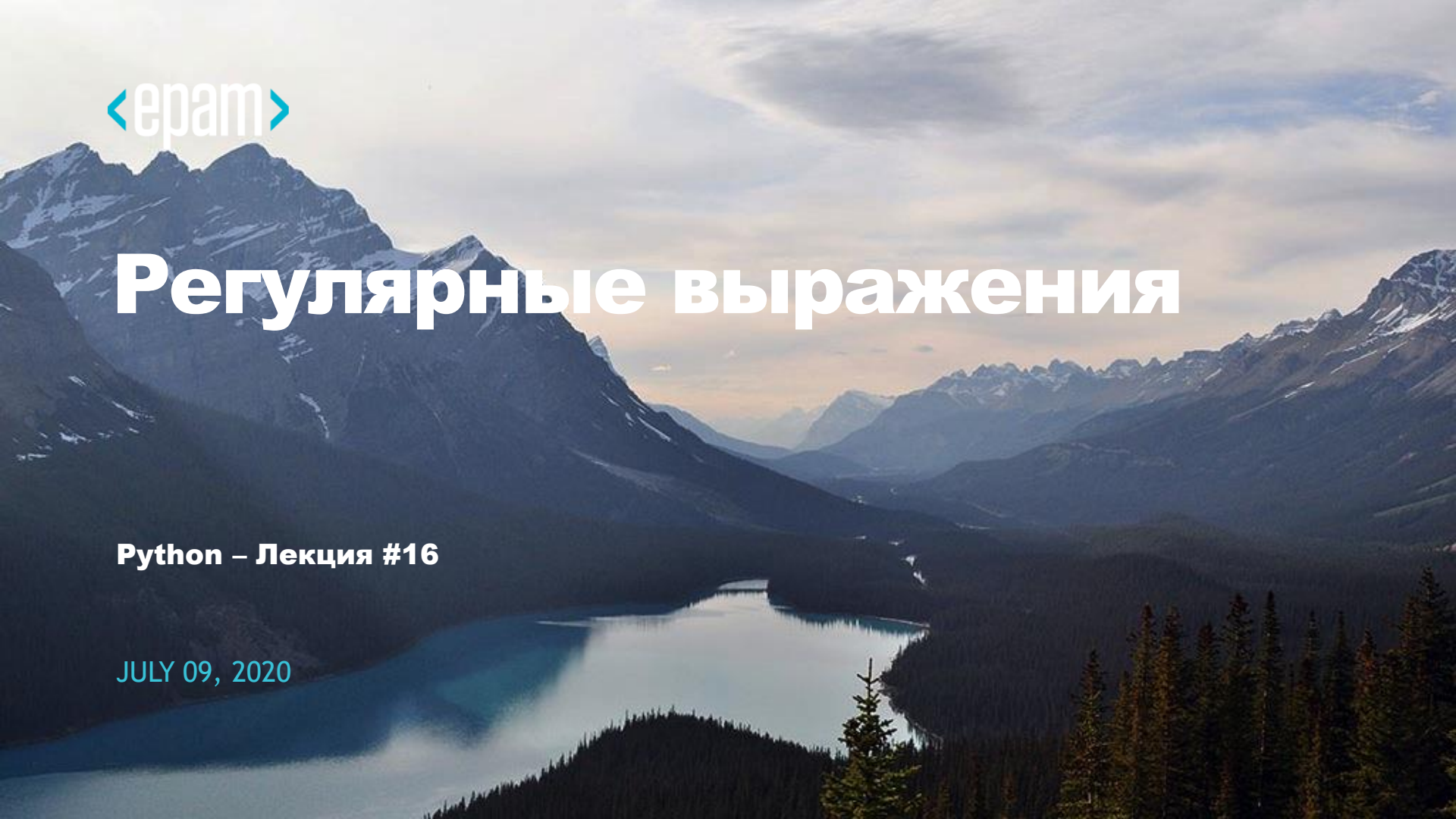


Регулярные выражения

Python – Лекция #16

JULY 09, 2020

ТЕМЫ

1. Основные понятия
2. Специальные операторы
3. Задачи

Часть 1

ОСНОВНЫЕ ПОНЯТИЯ

ОСНОВНЫЕ ПОНЯТИЯ

Регулярное выражение (англ. *Regular expression*) — специальная последовательность символов, которая помогает сравнивать или находить строки или наборы строк, используя специализированный синтаксис составления паттернов.

Широко используются в мире UNIX

В python работа с регулярными выражениями обеспечивается модулем **re** (его необходимо импортировать перед использованием)

Поскольку синтаксис паттернов включает множество спецсимволов, паттерны принято записывать в виде «сырых» строк (**raw** strings)

МЕТОДЫ МОДУЛЯ RE

- re.match()
- re.search()
- re.findall()
- re.split()
- re.sub()
- re.compile()

RE.MATCH

re.match(*pattern*, *string*) - поиск по заданному шаблону в начале строки

```
import re

result = re.match(r'Python', 'Python is the best')
print(result)  # <re.Match object; span=(0, 6), match='Python'>
print(result.group(0))  # Python
print(result.start())  # 0
print(result.end())  # 6

result = re.match(r'the', 'Python is the best')
print(result)  # None
```

RE.SEARCH

re.search(*pattern*, *string*) - поиск по заданному шаблону. Ищет по всей строке, но возвращает только первое найденное совпадение.

```
import re

result = re.search(r'the', 'Python is the best')
print(result)  # <re.Match object; span=(10, 13), match='the'>
print(result.group(0))  # the
print(result.start())  # 10
print(result.end())  # 13
```

СПЕЦИАЛЬНЫЕ ОПЕРАТОРЫ. ПЕРВЫЙ ВЗГЛЯД

.	Один любой символ, кроме новой строки \n.
?	0 или 1 вхождение шаблона слева
+	1 и более вхождений шаблона слева
*	0 и более вхождений шаблона слева
\w	Любая цифра или буква (\W — все, кроме буквы или цифры)
\d	Любая цифра [0-9] (\D — все, кроме цифры)
\s	Любой пробельный символ (\S — любой непробельный символ)
\b	Граница слова
[..]	Один из символов в скобках ([^..] — любой символ, кроме тех, что в скобках)
\	Экранирование специальных символов (\. означает точку или \+ — знак «плюс»)
^ и $	Начало и конец строки соответственно
{n,m}	От n до m вхождений ({,m} — от 0 до m)
a|b	Соответствует a или b
()	Группирует выражение и возвращает найденный текст
\t, \n, \r	Символ табуляции, новой строки и возврата каретки соответственно

ИСПОЛЬЗОВАНИЕ СПЕЦИАЛЬНЫХ ОПЕРАТОРОВ

*	0 и более вхождений шаблона слева
\w	Любая цифра или буква (\W — все, кроме буквы или цифры)

```
import re

result = re.search(r'\w*th\w*', 'Python is the best')
print(result)  # <re.Match object; span=(0, 6), match='Python'>
print(result.group(0))  # Python
print(result.start())  # 0
print(result.end())  # 6
```

RE.FINDALL

re.findall(*pattern*, *string*) - возвращает список всех найденных совпадений.

```
import re

result = re.findall(r'\w*th\w*', 'Python is the best')
print(result)  # ['Python', 'the']
```

RE.SPLIT

re.split(*pattern*, *string*, *[maxsplit=0]*) - разделяет строку по заданному шаблону.

```
import re

result = re.split(r'\W+', 'Python is the best. Moreover, re is the best')
print(result)  # ['Python', 'is', 'the', 'best', 'Moreover', 're', 'is', 'the', 'best']
```

RE.SUB

re.sub(*pattern*, *repl*, *string*) - ищет шаблон в строке и заменяет его на указанную подстроку. Если шаблон не найден, строка остается неизменной.

```
import re

result = re.sub(r'Russia', 'the World', 'Python is the best prog lang in Russia')
print(result)  # Python is the best prog lang in the World
```

RE.COMPILE

re.compile(*pattern*, *repl*, *string*) - собрать регулярное выражение в отдельный объект, который может быть использован для поиска. Это также избавляет от переписывания одного и того же выражения.

```
import re

pattern = re.compile(r'\w*th\w*’)

result = pattern.findall('Python is the best')
print(result)  # ['Python', 'the']

result2 = pattern.findall('Cython is the thing’)
print(result2)  # ['Cython', 'the', 'thing']
```

Часть 2

СПЕЦИАЛЬНЫЕ ОПЕРАТОРЫ

СПЕЦИАЛЬНЫЕ ОПЕРАТОРЫ

Шаблон	Описание	Пример	Применяем к тексту
.	Один любой символ, кроме новой строки \n.	м.л.ко	молоко, малако, Им0л0коИхлеб
\d	Любая цифра	СУ\d\d	СУ35, СУ111, АЛСУ14
\D	Любой символ, кроме цифры	926\D123	926)123, 1926-1234
\s	Любой пробельный символ (пробел, табуляция, конец строки и т.п.)	бор\soda	бор ода, бор ода, борода
\S	Любой непробельный символ	\S123	X123, я123, !123456, 1 + 123456
\w	Любая буква (то, что может быть частью слова), а также цифры и _	\w\w\w	Год, f_3, qwert
\W	Любая не-буква, не-цифра и не подчёркивание	сом\W	сом!, сом?
[..]	Один из символов в скобках, а также любой символ из диапазона a-b (если нужен минус, его нужно указать последним или первым)	[0-9][0-9A-Fa-f]	12, 1F, 4B
[^..]	Любой символ, кроме перечисленных	<[^>]>	<1>, <a>, <>

СПЕЦИАЛЬНЫЕ ОПЕРАТОРЫ

Шаблон	Описание	Пример	Применяем к тексту
\b	Начало или конец слова (слева пусто или не-буква, справа буква и наоборот). В отличие от предыдущих соответствует позиции, а не символу	\bвал	вал, перевал, Перевалка
\B	Не граница слова: либо и слева, и справа буквы, либо и слева, и справа НЕ буквы	\Bвал	перевал, вал, Перевалка
		\Bвал\B	перевал, вал, Перевалка

КВАНТИФИКАТОРЫ И ЖАДНОСТЬ

Шаблон	Описание	Пример	Применяем к тексту
{n}	Ровно n повторений	\d{4}	1, 12, 123, 1234, 12345
{m,n}	От m до n повторений включительно	\d{2,4}	1, 12, 123, 1234, 12345
{m,}	Не менее m повторений	\d{3,}	1, 12, 123, 1234, 12345
{,n}	Не более n повторений	\d{,2}	1, 12, 123
?	Ноль или одно вхождение, синоним {0,1}	валы?	вал, валы, валов
*	Ноль или более, синоним {0,}	СУ\d*	СУ, СУ1, СУ12, ...
+	Одно или более, синоним {1,}	a\)+	a), a)), a))), ba)])
*? +? ?? {m,n}? {,n}? {m,}?	По умолчанию квантификаторы *жадные* — захватывают максимально возможное число символов. Добавление ? делает их *ленивыми*, они захватывают минимально возможное число символов	\(.*\) \(.*?\)	(a + b) * (c + d) * (e + f) (a + b) * (c + d) * (e + f)

Часть 3

ЗАДАЧИ

ЗАДАЧИ

Задача 1: Вернуть первое слово из строки.

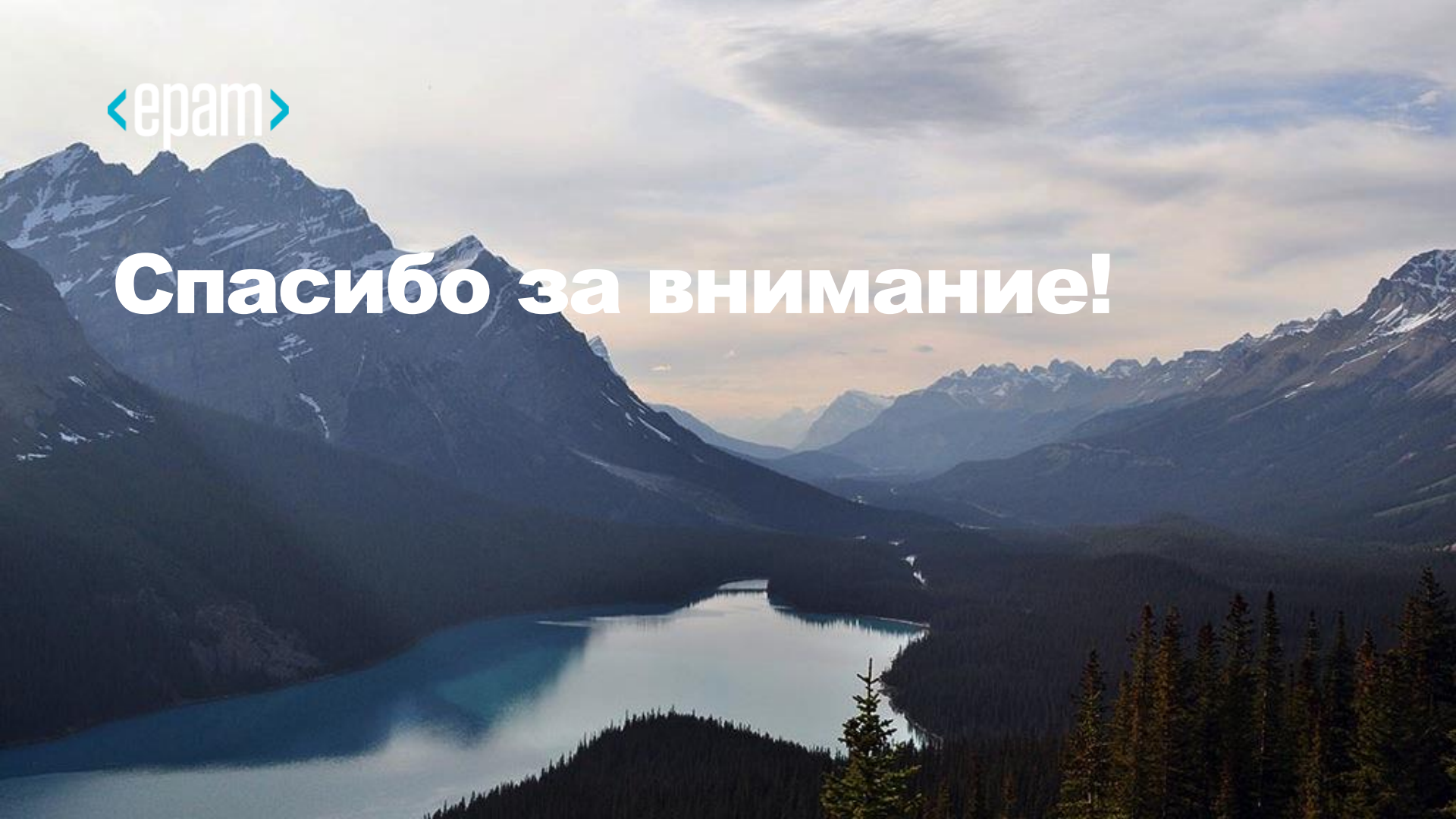
Задача 2: Вернуть первые два символа каждого слова.

Задача 3: Вернуть список доменов из списка адресов электронной почты. Доп.: вытащить только домен верхнего уровня

Задача 4: Извлечь дату из строки ('Amit 34-3456 12-05-2007, XYZ 56-4532 11-11-2011, ABC 67-8945 12-01-2009'). Доп.: вытащить только год

Задача 5: Извлечь все слова, начинающиеся на гласную

Задача 6: Разбить строку по нескольким разделителям


epam
Спасибо за внимание!