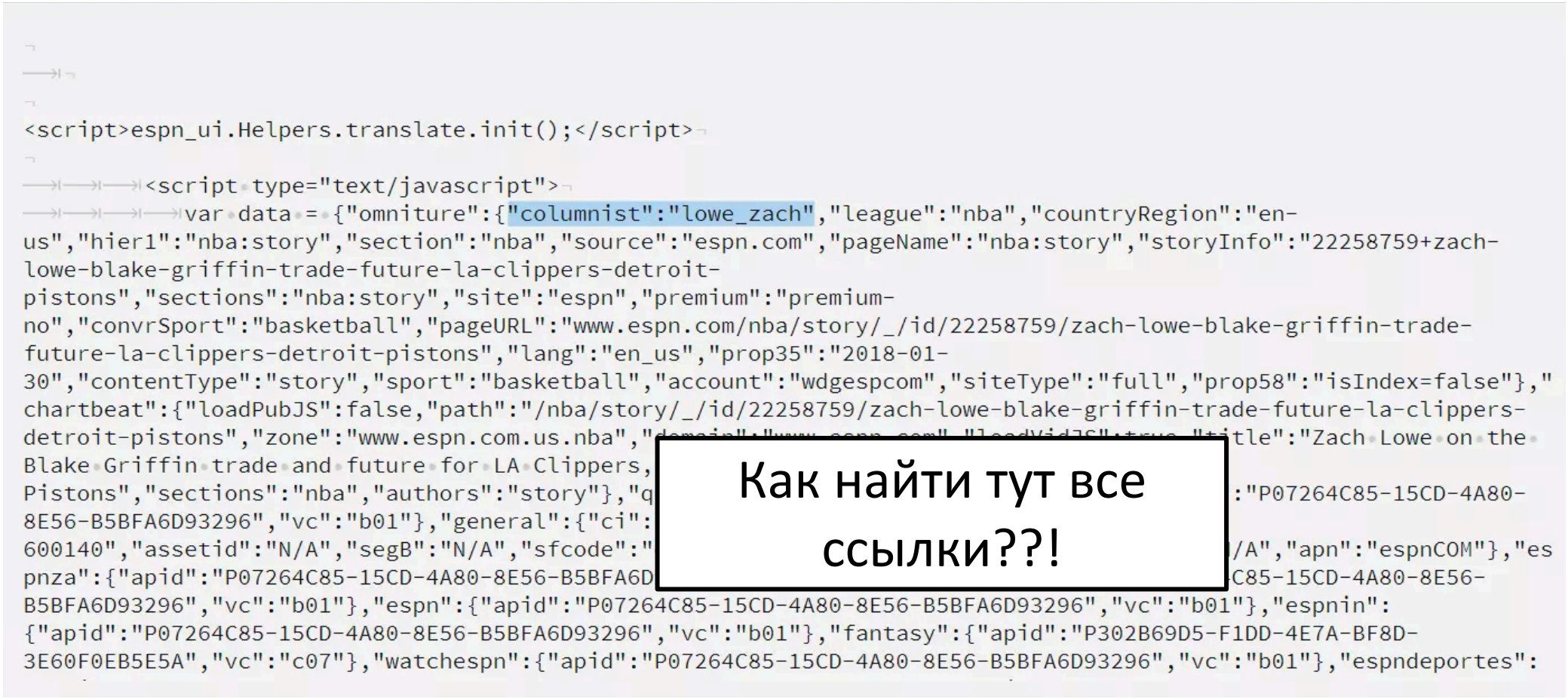
# Регулярные выражения

# Поиск в огромном массиве текста



# Поиск в огромном массиве текста

Т.е. я ищу:

www.текст.com/что-то там/...

шаблон

# Где понадобится?

- парсинг результатов вычислений
- парсинг сайта
- поиск и редактирование информации в человеческой речи
- работа с путями и файлами в ОС
- …

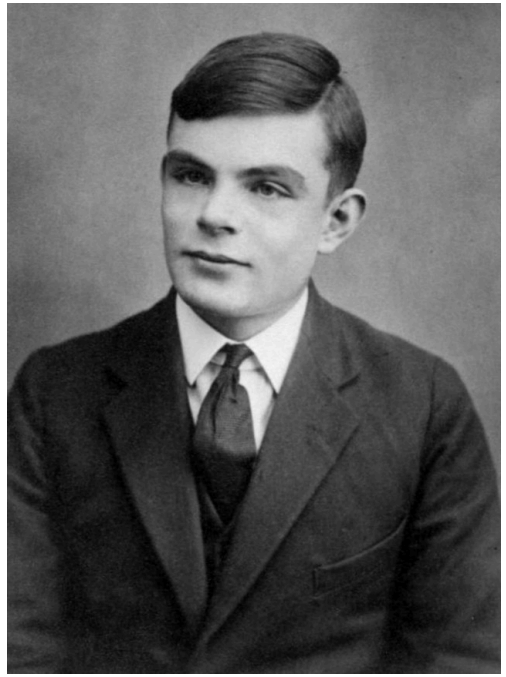

Во время изучения чего-то нового, я самозабвенно выдумываю невероятные ситуации, в которых это умение поможет мне спасти мир
О нет! Убийца должно быть последовал за ней в отпуск!
Но чтобы узнать где он, нам нужно прочесать 200 Мб писем в поисках чего-то схожего по формату с адресом!
Это безнадёжно!
Всем расступиться
Я знаю регулярные выражения
PERL!
клац клац

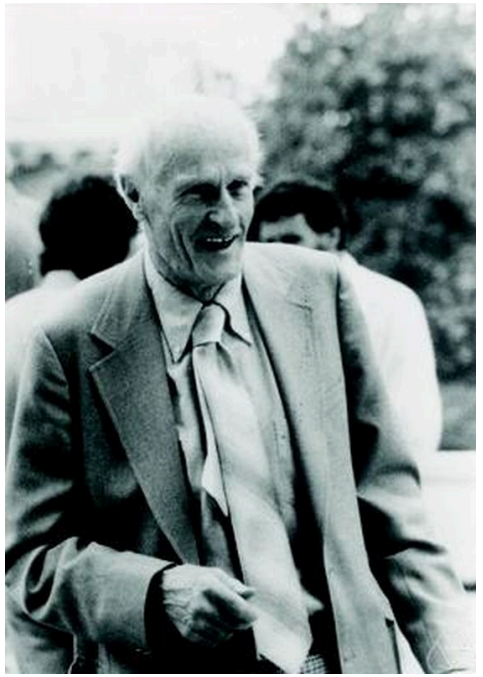
# Математика

- 1970е – У. Маккалок и У. Питтс (USA) и А.Тьюринг (Eng). Теория конечных автоматов и теория формальных языков.

- 1951 – С.Кинли (USA) Регулярные (распознаваемые) множества и языки

- 1962 – первые реализации в программировании (SNOBOL)

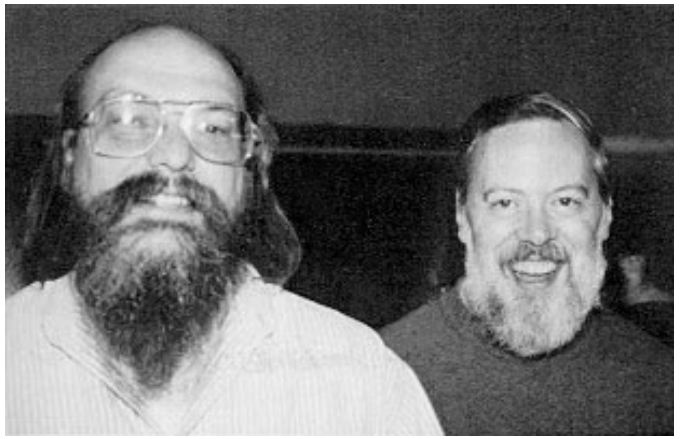




# IT

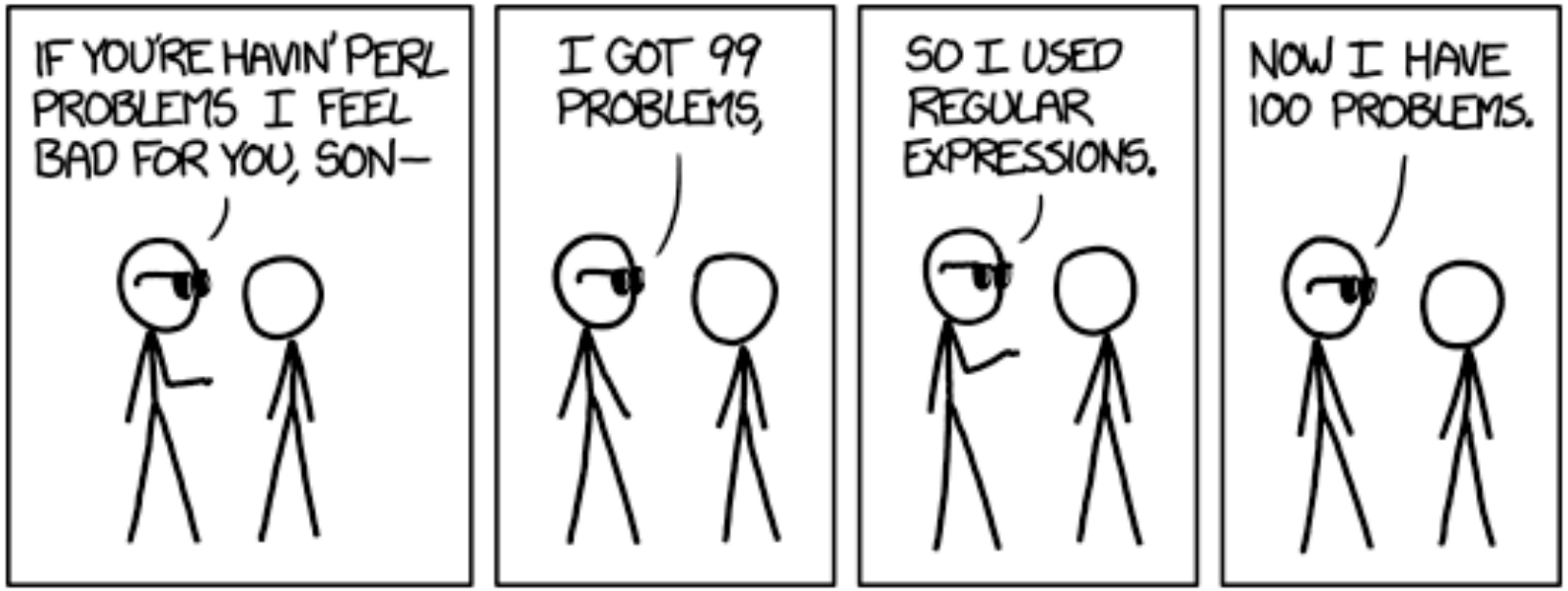
- 1970е - Кен Томпсон и Деннис Ритчи внедряют Regex в UNIX
  vi, ed и др. редакторы
- 1980е – PostgreSQL
- и понеслась

```
$ ed fstab
Newline appended
116
%l
/dev/hda2   /       ext2   defaults   1 1\r$
/dev/hdb1   /home   ext2   defaults   1 2\r$
/dev/hda1   swap    swap   pri=40     0 0$
3s/40/42/
w fstab
116
_
```



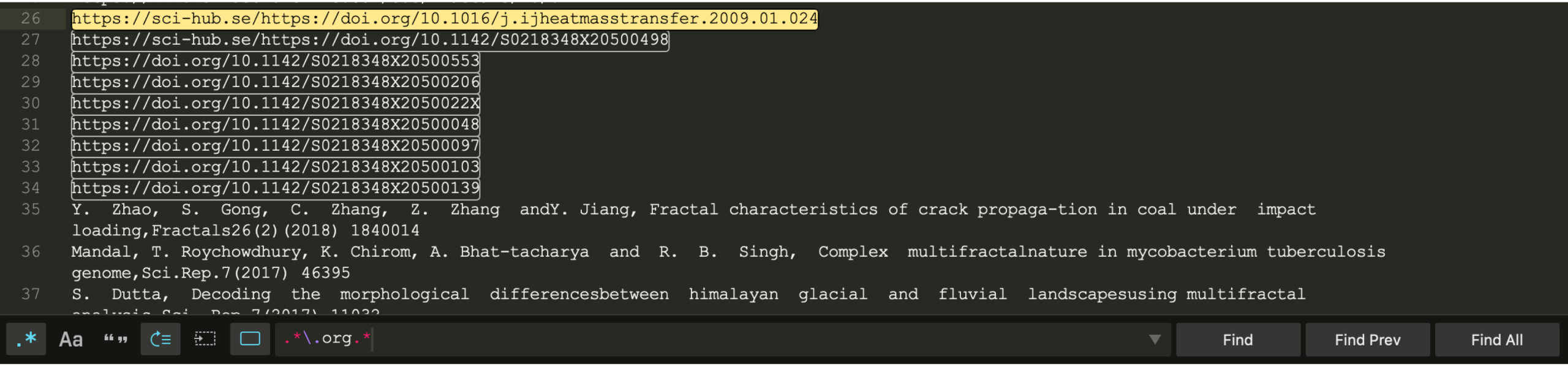
# IT

- 1980е - Ларри Уолл и его Perl



# IT

сейчас поддержка regex есть во всех адекватных редакторах текста: Sublime, nano, atom, notepad++ ... Свои реализации есть и в Word.



# Да что же это такое?

- Регулярные выражения (regex – regular expressions) – это **последовательность** [спец] символов, позволяющая находить в тексте **совпадение** с одним или несколькими **шаблонами**

- *.txt – регулярное выражение, с помощью которого я буду искать все файлы формата “.txt”

- regex - это математический концпет, который может быть по-своему реализован в разных языках программирования

| Оператор | Описание |
|---|---|
| . | 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 |

полный список тут: https://habr.com/ru/post/349860/


When you do a regex expression correctly first try without using google for help
You know, I'm something of a scientist myself.
imgflip.com

# Как можно

| **Регулярка** | **Её смысл** |
| --- | --- |
| simple text | В точности текст «simple text» |
| \d{5} | Последовательности из 5 цифр<br>\d означает любую цифру<br>{5} — ровно 5 раз |
| \d\d/\d\d/\d{4} | Даты в формате ДД/ММ/ГГГГ<br>(и прочие куски, на них похожие, например, 98/76/5432) |

# Как не стоит

(?:[a-z0-9!#$%&'*+/=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*+/=?^_`{|}~-]+)*|"(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])*")@(?:(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?|\[(?:(?:25[0-5]| 2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?|[a-z0-9-]*[a-z0-9]:(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]|\\[\x01-\x09\x0b\x0c\x0e-\x7f])+)\])

*позволяет распознать в тексте email

# Несколько заданий

- https://regex101.com/
- https://tproger.ru/translations/regular-expression-python/

1. Найти подстроку “cat” в строке
Проверить на: ”CatcatCATCaT”

*Решение: r”cat”*


2. Найти все подстроки, в которых после символа“a” идёт 2 или 3 символа b.
Проверить на: “Ab”, “Cgiabb_abbb_ab_abbbbb”

*Решение: r”ab{2,3}”*

3. Найти все подстроки, в которых две подстроки из букв нижнего регистра отделены символом“_”
Проверить на:
”John_Smith name_surname_Name_Surname”
“John_Smith_name_surname_Name_Surname”

*Решение: r”[a-z]+_[a-z]+”*

# Предостережение

будьте аккуратнее, используя регулярные выражения в web-запросах (сайты/ssh/email-протоколы/…)

за такое запросто можно словить


используйте официальные API

# HOW TO REGEX


STEP 1: OPEN YOUR FAVORITE EDITOR
@GARABATOKID
STEP 2: LET YOUR CAT PLAY ON YOUR KEYBOARD
/^([A-Z0-9_\.-

# One more thing

| G | H |
|---|---|
| Ссылка на репозиторий | Результаты теста 23/10 |
| https://github.com/AlexanderSwobodskii/Course_project.git | + |
| https://github.com/Sergey1vstr/Semestral-Project | + |
| https://github.com/Anastasiia-star/Kproject | + |
| https://github.com/Anastasiia-star/Kproject | + |
| | переписать |
| | + |
| | + |
| https://github.com/Natalie-Palchikovskaya/my_project | + |
| https://github.com/AfanasyevAA6/my-project | + |
| https://github.com/anzh-bal/kursovaya | + |
| | + |