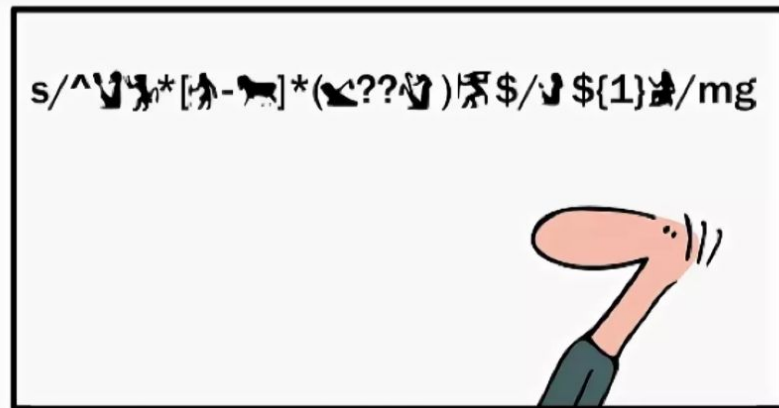










Регулярные выражения



ANCIENT EGYPTIAN REGEXP

Регулярные выражения (regular expressions, регекспы) — язык поиска подстрок в тексте

 regular expressions 101  @regex101  donate  sponsor  contact  bug reports & feedback  wiki 

REGULAR EXPRESSION no match

`ir"` | insert your regular expression here " gm

TEST STRING

ВШЭ · была · создана · постановлением · Правительства ·. · Руководит · ВШЭ · Ярослав · Кузьминов · — · бессменный · ректор ·, · участвовавший · в · создании · вуза ·. · Органами · управления · выступают · различные · советы: · учёный ·, · наблюдательный ·, · попечительский ·, · международный · экспертный ·. · Наблюдательный · совет · возглавляет · Сергей · Кириенко ·, · до · него · на · этом · посту · был · Вячеслав · Володин ·. · Попечительский · совет · с · Германом · Грефом · во · главе · отвечает · за · привлечение · финансов ·, · в · состав · также · входят · Виктор · Вексельберг ·, · Аркадий · Волож ·, · Михаил · Задорнов ·, · Леонид · Михельсон ·, · Вадим · Мошкович · и · Михаил · Прохоров ·.

Сооснователь · ВШЭ ·, · бывший · министр · экономики · Евгений · Ясин ·, · занимает · должность · научного · руководителя · университета ·, · представляет · вуз · в · других · научных · организациях ·. · Президент · вуза · — · бывший · вице-премьер · Александр · Шохин ·, · в · его · обязанности · входит · представлять · Вышку · в · госструктурах ·. · С · 2016 · года · на · должности · вице-президента · находится · Игорь · Агамирзян ·, · занимавший · до · этого · пост · в · совете · директоров · РБК ·. · На · постах · первых · проректоров · Вышки · в · настоящий · момент · находятся · экономисты · Вадим · Радаев ·, · Александр · Шамрин ·, · Леонид · Гохберг · и · Лев · Якобсон ·.

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
прочитать 200 Мб писем в поисках
чего-то похожего по формату с адресом!

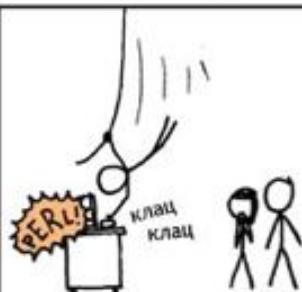


Это безнадежно!

Всем расступиться



Я знаю регулярные
выражения



HOW TO REGEX

STEP 1: OPEN YOUR FAVORITE EDITOR



STEP 2: LET YOUR CAT PLAY ON YOUR KEYBOARD





С другой стороны:

Некоторые люди, столкнувшись с проблемой, думают: «О, а использую-ка я регулярные выражения». Теперь у них есть две проблемы.

Джейми Завински
легендарный программист, разработчик браузера Netscape, сооснователь Mozilla



Метасимволы не соответствуют сами себе:

. ^ \$ * + ? [] { } \ | ()



Типичные проблемы

1. Регулярные выражения могут выглядеть нечитаемо

`^\+\d{1,3}\(\d{3}\)(|-)?\d{3}(|-)?\d{2}(|-)?\d{2}$`

2. Регулярные выражения не так просто отлаживать
(есть специальные инструменты, но в обычных штуках типа питона или блокнота, они или просто работают — или просто НЕ работают)



Например, мы можем описать одним регекспом:

- Все варианты названия профессии «риелтор» (риелтор, риелтер, риэлтер, риэлтор)
- Все четырехзначные даты в тексте (или только те из них, что не находятся в более длинных цепочках чисел)
- Все почтовые ящики на русских доменах (vasya1999@yandex.ru, petya2001@mail.ru, daskorinkin@edu.hse.ru, но не skorinkin@gmail.com)
- Все формы слова грузовик (грузовик, грузовика, грузовиками, грузовиках и т.п., но не грузовиковый)
- Все повторы одного слова (любого)
- Все основы, которые начинаются на -на и кончаются на -ся (напиться, наесться, намарафетиться)
- Все теги/атрибуты/комментарии в HTML-коде

«Муаммар Каддафи»

$$\begin{array}{c}
 M \\
 \left\{ \begin{array}{c} u \\ o \\ ou \end{array} \right. \quad \left\{ \begin{array}{c} \emptyset \\ , \end{array} \right. \quad \left\{ \begin{array}{c} \emptyset \\ a \end{array} \right. \quad \left\{ \begin{array}{c} mm \\ m \end{array} \right. \quad ar \quad \left\{ \begin{array}{c} Al \\ al \\ El \\ el \\ \emptyset \end{array} \right. \quad \left\{ \begin{array}{c} - \\ _ \\ \emptyset \end{array} \right. \quad \left\{ \begin{array}{c} Q \\ G \\ K \\ Kh \end{array} \right. \quad a \left\{ \begin{array}{c} d \\ dh \\ dd \\ dh dh \\ th \\ zz \end{array} \right. \quad a \left\{ \begin{array}{c} f \\ ff \end{array} \right. \quad \left\{ \begin{array}{c} i \\ y \end{array} \right.
 \end{array}$$

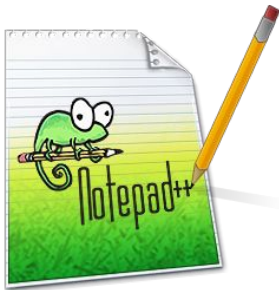
M(u|ou?)'a?mm?ar

((A|a|E|e)l)?-?

(Q|G|Kh?)a(d(d|h(dh)?)|th|zz)ff?(i|y)

Где можно использовать регекспы

perl

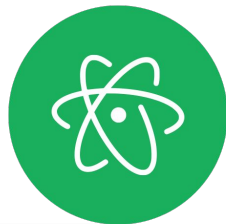


python™

Microsoft
PowerShell



gr3p



C++



Brackets



PSPad
Freeware editor



Регулярные выражения в Python

- **import** re
- Основные функции:
 - `re.search(r'a', 'abca') -> abca`
 - `re.findall(r'a', 'abca') -> abca`
 - `re.match(r'a', 'abca') -> abca`
 - `re.split(r'b', 'abca') -> ['a', 'ca']`
 - `re.sub(r'b', 'z', 'abca') -> 'azca'`

regex101.com



REGULAR EXPRESSION

2 matches, 50 steps (~1 ms)

/

[a-я]

*

)

(

c

)

[a-я]

*

)

← Сюда пишем регулярное выражение

/g

TEST STRING

SWITCH TO UNIT TESTS ▶

вася и тася

Сюда кладем текст, в котором ищем что-то

—

. (точка)

. (точка) — любой символ

● Примеры:

Регулярное выражение	Что найдется
K.T	

. (точка) — любой символ

● Примеры:

Регулярное выражение	Что найдется
K.T	КОТ, КИТ, КАТ

. (точка) — любой символ

● Примеры:

Регулярное выражение	Что найдется
к.т	кот, кит, кат
к.шка	

. (точка) — любой символ

● Примеры:

Регулярное выражение	Что найдется
К.Т	КОТ, КИТ, КАТ
К.шка	кошка, кашка, кишка, кушка

. (точка) — любой символ

● Примеры:

Регулярное выражение	Что найдется
к.т	кот, кит, кат
к.шка	кошка, кашка, кишка, кушка
..к.шка	

. (точка) — любой символ

● Примеры:

Регулярное выражение	Что найдется
к.т	кот, кит, кат
к.шка	кошка, кашка, кишка, кушка
..к.шка	букашка, макушка

. (точка) — любой символ

● Примеры:

Регулярное выражение	Что найдется
К.Т	КОТ, КИТ, КАТ
К.шка	кошка, кашка, кишка, кушка
..к.шка	букашка, макушка
.В.Н.	

. (точка) — любой символ

● Примеры:

Регулярное выражение	Что найдется
К.Т	кот, кит, кат
К.шка	кошка, кашка, кишка, кушка
..К.шка	букашка, макушка
.В.Н.	Ивану, авеню, Ивины, звона



Задание:

- Напишите регулярное выражение, которое сопоставится со словами *риэлтор*, *риэлтер*, *риелтор* и *риелтер* (но НЕ сопоставится со словам *рипитер* и *реактор*, например), **используя из метасимволов только . точку**
- То есть хочется такой результат:

TEST STRING

Регулярное · выражение · сопоставляется · со · словами · **риэлтор** ,
риэлтер , **риелтор** · и · **риелтер** , но · не · рипитер · или · реактор .
Из · метасимволов · только | · " · "

что означает * (звездочка)?

***(звездочка) — предыдущий символ повторяется
от нуля до бесконечности раз**

● Примеры:

Регулярное выражение	Что найдется
ка*линка	

***(звездочка) — предыдущий символ повторяется
от нуля до бесконечности раз**

● Примеры:

Регулярное выражение	Что найдется
ка*линка	кааааалинка, калинка, клинка

***(звездочка) — предыдущий (мета)символ
повторяется от нуля до бесконечности раз**

● Примеры:

Регулярное выражение	Что найдется
ка*линка	кааааалинка, калинка, клинка
к.*т	

***(звездочка) — предыдущий символ повторяется от нуля до бесконечности раз**

● Примеры:

Регулярное выражение	Что найдется
ка*линка	кааааалинка, калинка, клинка
к.*т	кот, коооооооооот, крот койот, компот, кашалот, коверкот, к7@%5т, кт

***(звездочка) — предыдущий (мета)символ повторяется от нуля до бесконечности раз**

● Примеры:

Регулярное выражение	Что найдется
ка*линка	кааааалинка, калинка, клинка
к.*т	кот, коооооооооот, крот койот, компот, кашалот, коверкот, к7@%5т, кт
.*кот.*	

***(звездочка) — предыдущий (мета)символ повторяется от нуля до бесконечности раз**

● Примеры:

Регулярное выражение	Что найдется
ка*линка	кааааалинка, калинка, клинка
к.*т	кот, коооооооооот, крот койот, компот, кашалот, коверкот, к7@%5т, кт
.*кот.*	кот, апперкот, который, облокотиться



Задание:

- Напишите регулярное выражение, которое полностью накроет любую строку, которая начинается на "горох":

TEST STRING

горох

гороховый

гороховая · каша

перловая · каша

перлово-гороховая · каша

горохами

огорошил

—

+ (плюс)

+(плюс) — предыдущий символ повторяется от одного до бесконечности

● Примеры:

Регулярное выражение	Что найдется
ка+линка	

+(плюс) — предыдущий символ повторяется от одного до бесконечности

● Примеры:

Регулярное выражение	Что найдется
ка+линка	кааааалинка, калинка

+(плюс) — предыдущий (мета)символ повторяется от одного до бесконечности

● Примеры:

Регулярное выражение	Что найдется
ка+линка	кааааалинка, калинка
тр.л.л.+	

+(плюс) — предыдущий (мета)символ повторяется от одного до бесконечности

● Примеры:

Регулярное выражение	Что найдется
ка+линка	кааааалинка, калинка
тр.л.л.+	трололо, труляля, тралалаалаляля

+(плюс) — предыдущий (мета)символ повторяется от одного до бесконечности

● Примеры:

Регулярное выражение	Что найдется
ка+линка	кааааалинка, калинка
тр.л.л.+	трололо, труляля, тралалаалаяляля
.+есть.+	

+(плюс) — предыдущий (мета)символ повторяется от одного до бесконечности

● Примеры:

Регулярное выражение	Что найдется
ка+линка	кааааалинка, калинка
тр.л.л.+	трололо, труляля, тралалаалаяляля
.+ешь.+	наешься, поешьте, решаешься

—

? (вопрос)

?(вопрос) — предыдущий символ повторяется от нуля до одного

● Примеры:

Регулярное выражение	Что найдется
кр?от	

?(вопрос) — предыдущий символ повторяется от нуля до одного

● Примеры:

Регулярное выражение	Что найдется
кр?от	кот, крот

?(вопрос) — предыдущий (мета)символ повторяется от нуля до одного

● Примеры:

Регулярное выражение	Что найдется
кр?от	кот, крот
кр?от.?	

?(вопрос) — предыдущий (мета)символ повторяется от нуля до одного

● Примеры:

Регулярное выражение	Что найдется
кр?от	кот, крот
кр?от.?	кот, крот, кроту, коту, ката, коте, коты, кроты,

{ } (фигурные скобки)

- $\{n\}$ (фигурные скобки) — предыдущий символ повторяется n раз
- $\{n,m\}$ (фигурные скобки) — предыдущий символ повторяется от n до m раз
- Примеры:

Регулярное выражение	Что найдется
$k\{3\}омбо$	кккомбо
$k\{3,5\}омбо$	кккомбо, ккккомбо, кккккомбо

По умолчанию квантификаторы жадные:

`a.*` будет пытаться съесть максимально длинную строку:

`a.*a` -> **араукария**

Ленивый — максимально короткую.

Чтобы сделать ленивый, надо добавить к квантификатору вопрос:

`a.*?a` -> **а**раукария

