

Discussion 8: Regular Expressions

SI 206: Data-Oriented Programming

Instructor: Dr. Barbara (Barb) Ericson

GSI: Kexuan (Michael) Huang

IA: Cristina & Jade

School of Information
University of Michigan

Fall 2023

Deadlines

- Homework 5 due this Friday (Oct 27th)
- Midterm 2 on week 10 (now we are at week 9!)

Table of Contents

Introduction

Tools

Practice
Problem

- ➊ Introduction
- ➋ Tools
- ➌ Practice Problem

Table of Contents

Introduction

Tools

Practice
Problem

- 1 Introduction
- 2 Tools
- 3 Practice Problem

Why Regular Expressions?



Validate or search for data with **pattern**, for example:

- Email
- Address
- Phone number

Table of Contents

Introduction

Tools

Practice
Problem

- 1 Introduction
- 2 Tools
- 3 Practice Problem

How to get started?

Introduction

Tools

Practice
Problem

Start with
regex101.com



HOW TO REGEX



STEP 1: OPEN YOUR FAVORITE EDITOR



STEP 2: LET YOUR CAT PLAY ON YOUR KEYBOARD



Use Regex101

Introduction

Tools

Practice
Problem

The screenshot shows the regex101.com website interface. The browser address bar displays "regex101.com". The website header includes a menu icon, the text "regular expressions 101", and links for "social", "donate", and "info".

The main content area is divided into two columns:

- REGULAR EXPRESSION:** Displays the regex `r" \(?\d{3}\)?(?:\s|-)\d{3}-\d{4}` with a status of "4 matches (187 steps, 0.0ms)" and a version dropdown set to "v1".
- TEST STRING:** Lists several test strings, with the following ones highlighted in blue:
 - `(302) 338-2933`
 - `abc-def-ghij`
 - `808 393-2333`
 - `(123)-4567-890`
 - `303-393-2839`
 - `608-888 8888`
 - `(404)-881-9023`
- EXPLANATION:** Provides a detailed breakdown of the regex components:
 - `r` matches the character `r` with index 40₁₀ (28₁₆ or 50₈) literally (case sensitive).
 - `\d` matches a digit (equivalent to `[0-9]`).
 - `\)` matches the character `)` with index 41₁₀ (29₁₆ or 51₈) literally (case sensitive).
 - Non-capturing group** `(?:\s|-)`.
 - `\d` matches a digit (equivalent to `[0-9]`).
- MATCH INFORMATION:** Lists the four matches found:
 - Match 1**: 0-14 `(302) 338-2933`
 - Match 2**: 28-40 `808 393-2333`
 - Match 3**: 56-68 `303-393-2839`
 - Match 4**: 82-96 `(404)-881-9023`
- QUICK REFERENCE:** A section at the bottom for additional resources.

Use Regex101 with Cheat-sheet!

Introduction

Tools

Practice
Problem

Regular Expression Basics	Regular Expression Character Classes	Regular Expression Flags
.	[ab-d] One character of: a, b, c, d	i Ignore case
a	[^ab-d] One character except: a, b, c, d	m ^ and \$ match start and end of line
ab	[b] Backspace character	s . matches newline as well
a b a or b	\d One digit	x Allow spaces and comments
a* 0 or more a's	\D One non-digit	L Locale character classes
\ Escapes a special character	\s One whitespace	u Unicode character classes
	\S One non-whitespace	(?iLmsux) Set flags within regex
	\w One word character	
	\W One non-word character	
Regular Expression Quantifiers	Regular Expression Assertions	Regular Expression Special Characters
* 0 or more	^ Start of string	\n Newline
+ 1 or more	\A Start of string, ignores m flag	\r Carriage return
? 0 or 1	\$ End of string	\t Tab
{2} Exactly 2	\Z End of string, ignores m flag	\YYY Octal character YYY
{2, 5} Between 2 and 5	\b Word boundary	\xYY Hexadecimal character YY
{2,} 2 or more	\B Non-word boundary	
{,5} Up to 5	(?=...) Positive lookahead	
	(?!...) Negative lookahead	
	(?<=...) Positive lookbehind	
	(?<!...) Negative lookbehind	
	(?00) Conditional	
Regular Expression Groups		Regular Expression Replacement
(...) Capturing group		\g<0> Insert entire match
(?P<Y>...) Capturing group named Y		\g<Y> Insert match Y (name or number)
(?:...) Non-capturing group		\Y Insert group numbered Y
\Y Match the Y'th captured group		
(?P=Y) Match the named group Y		
(?#...) Comment		

Use Regex101 with Cheat-sheet!

Introduction

Tools

Practice
Problem

- ① Think of all possible versions of the thing you'd like to retrieve
- ② Try it out on regex101.com
- ③ Code with Python

Table of Contents

Introduction

Tools

Practice
Problem

① Introduction

② Tools

③ Practice Problem

Practice Problem

Introduction

Tools

Practice
Problem

Background and Data

You and your partner are detectives working on a case. You've intercepted a series of fake emails between the killer and victim. Your task is to extract relevant information using regular expressions to aid in your investigation.

```
1  Sender: Vinny Su (vinnysu@gmail.com)
2  To: nicolelam@gmail.com
3  Subject: Re: Dinner Plans
4
5  Sure, dinner sounds great! Why not we meet at Michigan Union at 7pm?
6  The location is: 530 S State St, Ann Arbor, MI 48109
7
8  Vinny
9  666-555-1111
```

Practice Problems

Go to Canvas → Assignment → Discussion 8 and clone the GitHub Repo.

Your tasks

- Implement `get_email_count()`: returns a dictionary that counts email address
- Implement `get_phone_list()`: returns a list of phone numbers
- Implement `get_address_list()`: returns a list of tuples, each tuple contains state, city, zip code, street name and street number

```
{'mikex@gmail.com': 3, 'nicolelam@gmail.com': 4, 'vinnysu@gmail.com': 2}

['(555)765-4321', '666-555-1111', '(888)765 4321', '608 901 2345']

[('MI', 'Ann Arbor', '48104', 'E William St', '516'),
 ('MI', 'Ann Arbor', '48109', 'S State St', '530'),
 ('MI', 'Ann Arbor', '48104', 'Maynard St', '347'),
 ('MI', 'Ann Arbor', '48109', 'Duffield St', '1931')]
```