Regularni izrazi kroz primere

Mina Šekularac

Matematička gimnazija

NEDELJA$^4_{\text{INFORMATIKE}}$

21. mart 2018.

Čemu služe RegEx?

- Pronađu nam u datom tekstu sve one delove koji svojom strukturom zadovoljavaju postavljeni obrazac.

- Obezbede način za definisanje obrasca po kome će se kreirati određeni tekstualni podaci.

Skupovi karaktera

- Moguće je definisati skup karaktera:
  - `[a-z]`: sva mala slova
  - `[A-Z]`: sva velika slova
  - `[A-Z a-z]`: velika i mala slova
  - `[xyz]`: slova x,y,z
  - `[^aeiou]` : karakteri koji nisu samoglasnici
  - `[[\]]`: karakter [\]

Klase karaktera

- `\d`: cifra [0-9]
- `\D`: ne-cifra ([^0-9])
- `\s`: beli znaci (razmak, novi red, tab) ([\t\n\r\f\v])
- `\S`: ne-beli znaci ([^\t\n\r\f\v])
- `\w`: alfanumerici i donja crta ([a-zA-Z0-9_])
- `.` : odgovara bilo kom karakteru

Escape karatker

- `\` — `escape karakter` : menja značenje karakteru koji je neposredno iza njega
- `\d`: nije „d - slovo“, nego cifra
- `\.`: nije „bilo koji znak“, nego karakter koji odgovara tački
- `\?`: ne znači „opciono“, već upitnik

Operatori

- `+`: od 1 do beskonačno ponavljanja znaka koji je levo
- `*`: od 0 do beskonačno ponavljanja znaka koji je levo
- `?`: opciono pojavljivanje znaka koji je levo
- `{n},{a,b}`: pojavljivanje nekog znaka n puta (dato je n ili interval pa $n \in [a, b]$)
- `()`: grupisanje izraza sa vraćanjem
- `^`: očekuje se da se reč počne validnim izrazom
- `$`: očekuje se da se reč završi validnim izrazom

Funkcije

- `re.search`: vraća objekte koje zadovoljavaju izraz ukoliko ih ima (uz njih se koristi group() da bismo pročitali rezultat)
- `re.finditer`: vraća po jedno resenje za svaki šablon
- `re.findall`: vraca listu svih izraza koji zadovoljavaju šablon
- `re.compile`: čuva šablon koji kasnije mozemo da koristimo za prethodne funkcije
- `re.spilt`: deli string po na listu stingova (po razmaku, zarezu, tački ili nekom oznakom koju mi zadamo)
- `re.sub`: menja deo stringa koji zadovoljava izraz sa nekim datim stringom

Poznati primeri

- e-mail
  - `^([a-z0-9_ .-]+)@([\d a-z.-]+).([a-z.]{2,6})$`
- Hex broj
  - `^(-|+)?([a-f0-9]+)$`

re.search

```
import re

line = "Ova nedelja informatike je najbolja!"
searchObj = re.search(r'(.*) je (.*)', line)
if searchObj:
    print("searchObj.group() : ", searchObj.group())
    print("searchObj.group(1) : ", searchObj.group(1))
    print("searchObj.group(2) : ", searchObj.group(2))
else:
    print("Nothing found!")
```

re.sub

```
import re

phone_num = "061/606-1998 # Moj broj telefona"

# Brise sve posle komentara '#'
num = re.sub(r'#.*', "", phone_num)
print("Broj tel : ", num)

# Brise sve ne-brojeve
num = re.sub(r'\D', "", phone_num)
print("Broj tel : ", num)
```

Broj telefona

```
import re

telefoni = ['011/1234-56789', '011~1234~567',
            'fiksni: 011/7654-321 mob: 0601234567',
            '3218456', "Nemam telefon :'("]

for telefon in telefoni:
  if re.findall(r'0\d{2}/\d+', telefon) != []:
    print (re.findall(r'0\d{2}/\d+', telefon))

for telefon in telefoni:
  if re.findall(r'0\d{2}/?\d+\-?\d+', telefon) != []:
    print (re.findall(r'0\d{2}/?\d+\-?\d+', telefon))
```

Broj telefona

```
for telefon in telefoni:
  if re.findall(r'0\d{2}/?\d{4}\-?\d{3}', telefon) != []:
    print (re.findall(r'0\d{2}/?\d{4}\-?\d{3}', telefon))

for telefon in telefoni:
  if re.findall(r'(?:0\d{2}/?)?\d{4}\-?\d{3}', telefon) != []:
    print (re.findall(r'(?:0\d{2}/?)?\d{4}\-?\d{3}', telefon))

for telefon in telefoni:
  if re.findall(r'(?:0\d{2}/?)?(\d{4}\-?\d{3})', telefon) != []:
    print (re.findall(r'(?:0\d{2}/?)?(\d{4}\-?\d{3})', telefon))

for telefon in telefoni:
 if re.findall(r'(0\d{2}/?)?(\d{4}\-?\d{3})', telefon) != []:
    print (re.findall(r'(0\d{2}/?)?(\d{4}\-?\d{3})', telefon))
```

Mail

```
import re

mails = ["ni2018@gmail.com", "csweek <csweek@gmail.com>",
    "cs@week.com", "~ni@week4.com", "csweek@mg.edu.rs",
   "nedelja_informatike@mg.com", "nedeljainfo@mg.edu.co.rs" ,
   "csweek@mojmail123.rs", "Zvanicni mail je: csweek@mg.edu.rs"]


def check_mails(mail_pattern, mails):
    print (mail_pattern)
    for mail in mails:
        matched_mail =re.findall(mail_pattern, mail)
        print (matched_mail)
```

Mail

```
check_mails(r'[a-z][a-z0-9.]*@[a-z]+[\.a-z]*
                          \.(?:com|org|[a-z]{2})', mails)

check_mails(r'^[a-z][a-z0-9.]*@(?:[a-z]+
                        \.)+(?:com|org|[a-z]{2})$', mails)

check_mails(r'(?:^|<|\s)[a-z][a-z0-9.]*@(?:[a-z]+\.)+
                        (?:com|org|[a-z]{2})(?:$|>|\s)', mails)

check_mails(r'(?:^|<|\s)([a-z][a-z0-9.]*@(?:[a-z]+\.)+
                        (?:com|org|[a-z]{2}))(?:$|>|\s)', mails)
```

Web scraping

- Tehnika izvlačenja informacija sa web stranica
- **PAŽNJA**: Ne dozvoljavaju sve stranica da im se pristupa radi prikupljanja podataka!
- `urllib` i `BeautifulSoup` biblioteke
- Pomocu `RegEx` dohvatamo željene delove

urllib

- Biblioteka koja olakčava rad sa protokolima za dohvatanje resursa sa Interneta

- `urllib.request.urlopen(url)`: vraća podatke sa url (csv, txt, HTML stranicu, ...)

Primer urllib

```
import urllib.request
page = urllib.request.urlopen('http://info.cern.ch/')
for line in page:
    print(line.strip())

b'<html><head></head><body><header>'
b'<title>http://info.cern.ch</title>'
...

for line in page:
    line.strip().decode("utf-8")
    print(line.strip())

<html><head></head><body><header>
<title>http://info.cern.ch</title>
...
```

RegEx


WHENEVER I LEARN A NEW SKILL I CONCOCT ELABORATE FANTASY SCENARIOS WHERE IT LETS ME SAVE THE DAY.
OH NO! THE KILLER MUST HAVE FOLLOWED HER ON VACATION!
BUT TO FIND THEM WE'D HAVE TO SEARCH THROUGH 200 MB OF EMAILS LOOKING FOR SOMETHING FORMATTED LIKE AN ADDRESS!
IT'S HOPELESS!
EVERYBODY STAND BACK.
I KNOW REGULAR EXPRESSIONS.
PERL!
TAP TAP