

PROGRAMIRANJE 2

Regularni izrazi (regular expressions)

Robert Rozić, asis.

dr. sc. Marko Rosić, prof.

Creative Commons

- **Slobodno smijete:**
 - **dijeliti** - umnožavati, distribuirati i javnosti priopćavati djelo
 - **remiksirati** - prerađivati djelo
- **pod sljedećim uvjetima:**
 - **Imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
 - **Nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
 - **Dijeli pod istim uvjetima.** Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.

U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu. Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Tekst licence preuzet je s <https://creativecommons.org/>.



Regularni izrazi / Regular Expressions - RegEx(p)

- Slijed (niz) znakova koji formiraju **uzorak pretrage**
- Takvi uzorci (patterns) se koriste u algoritmima za pretragu teksta
- Najčešće se koriste za **pronalaženje** (find), **zamjenu** (replace) ili **validaciju**.
- Primjena:
 - pretraživači (search engines)
 - uređivači teksta (text editors)
 - alati za obradu teksta
 - leksička analiza...



```
/^([a-z\d\.-]+)@([a-z\d-]+)\.([a-z]{2,8})(\.[a-z]{2,8})?$/
```

Regularni izrazi / Regular Expressions - RegEx(p)

- Mnogi programski jezici omogućavaju regex mogućnosti kroz ugrađene metode ili kroz dodatne biblioteke (libraries)
- Python koristi ugrađeni paket “re” koji se koristi za rad s regularnim izrazima

```
import re
```



Regex funkcije

<code>findall(regex, text)</code>	Vraća listu svih podudaranja
<code>search(regex, text)</code>	Vraća tzv. “match objekt” ukoliko postoje podudaranja
<code>split(regex, text, br. pon)</code>	Vraća listu gdje je string podijeljen nakon svakog podudaranja
<code>sub(regex, zamjena, text, br. pon.)</code>	Mijenja jedan ili nekoliko podudaranja u stringu

Findall funkcija

```
import re
```

```
txt = "Ovo je predavanje programiranje2"
```

```
result = re.findall("programiranje2", txt)
```

```
print(result)
```


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

[]	set znakova	[abc]
.	svaki znak osim newline (/n)	prog..m
^	tekst počinje sa	^prog
\$	test završava sa	ram\$
*	0 ili više ponavljanja znaka lijevo od *	ma*n
+	1 ili više ponavljanja znaka lijevo od +	ma+n
?	0 ili 1 ponavljanje znaka lijevo od ?	ma?n
{m, n}	tačan broj ponavljanja / raspon od m,n	a{2,3}
	ili operator	a b
()	grupa	(a b c)xy
\	escape character ili posebni slijed	\\$ ili \A

Posebni sljedovi - special sequences

\A	Podudaranje ako su znakovi na početku stringa	\AOvo
\b	Podudaranje ukoliko su znakovi na početku ili na kraju rijeci	\bprog ili ram\b
\B	Obrnuto od \b. Podudaranje ukoliko znakovi nisu na početku ili na kraju rijeci.	\Bred ili nje\B
\d	Podudaranje se ukoliko string sadrži brojeve. [0-9]	
\D	Obrnuto od \d. Podudara se sa svim znakovima koji nisu brojevi.	
\s	Podudaranja na whitespace znakovima	
\S	Podudaranja na svim znakovima koji nisu whitespace	
\w	Podudaranje na bilo kojem alfanumerickom znaku (i _) [a-zA-Z0-9_]	
\W	Podudaranje na bilo kojem ne-alfanumerickom znaku [^a-zA-Z0-9_]	
\Z	Podudara se ako su znakovi na kraju stringa	

Setovi

[abc]	Podudaranje ukoliko postoji jedan od znakova u setu
[a-r]	Podudaranje za sva mala slova abecede između a i r
[^abc]	Podudaranje za sve znakove osim a, b i c
[123]	Podudaranje ukoliko postoji broj 1,2 ili 3
[0-9]	Podudaranje za bilo koji broj
[0-3][0-9]	Podudaranja za svaki dvoznamenkasti broj od 00 do 39
[a-zA-Z]	Podudaranje za bilo koje slovo abecede (veliko ili malo)
[+]	U setovima posebni znakovi nemaju posebno značenje

Split funkcija

```
import re
```

```
txt = "Ovo je predavanje programiranje2"
```

```
result = re.split("\s", txt)
```

```
print(result)
```

Sub funkcija

```
import re
```

```
txt = "Ovo je predavanje programiranje2"
```

```
result = re.sub("\s", "5", txt)
```

```
print(result)
```

Search funkcija i Match objekt

```
import re

txt = "Ovo je predavanje programiranje2"

result = re.search("predavanje", txt)

# Tuple pocetka i kraja, pocetak i kraj
print(result.span(), result.start(), result.end())

# Dio stringa koji se podudara
print(result.group())

# String i regularni izraz
print(result.string, result.re)
```

Search funkcija i Match objekt

```
import re

string = '39801 356, 2102 1111'
pattern = '(\d{3}) (\d{2})'
match = re.search(pattern, string)

if match:
    print(match.group())
else:
    print("pattern not found")
```

Raw string

Kada se prije regularnog izraza koristi “r” ili “R” prefix to nam govori da se radi o raw stringu.

Npr. “\n” je izraz za novi red, dok r“\n” znače dva znaka “\” backslash nakon kojeg je slovo n

Backslash se koristi za “escape” raznih znakova uključujući i metakaraktere.

```
import re

string = '\n and \r are escape sequences.'

result = re.findall(r'[\n\r]', string)

print(result)
```


Zadaća

Potrebno napisati regex koji vraća podudaranje ukoliko se unese string koji počinje kao prvo slovo vašeg imena, a završava kao prvo slovo prezimena. String mora sadržavati bar jedan broj između 0 i 5 i razmak.



[Regex101](#)

[W3Schools](#)

[Programiz](#)

[Python docs](#)



sumarum.sum.ba



marko.rosic@pmfst.hr

robert.rozic@fpmoz.sum.ba



HVALA NA POZORNOSTI !

FPMOZ