

Reguliariosios išraiškos ir Django URL sandara

Albertas Gimbutas

¹Matematikos ir informatikos institutas
Vilniaus universitetas

2014 pavasaris

1 Reguliariosios išraiškos

2 Django URL sandara

Reguliariosios išraiškos

Reguliarioji išraiška nusako simbolių išsidėstymo taisyklę, pagal kurią galima:

- Patikrinti ar simbolių eilutė atitinka reguliariąją išraišką.
- Išskirti simbolių eilutės fragmentus (simbolių grupes).

```
>>> import re
>>> pattern = '10+'
>>> re.findall(pattern, '100')
['100']
>>> re.findall(pattern, '200')
[]
>>> pattern = '1(0+)'
>>> re.findall(pattern, '100')
['00']
```

- python.org: re

re bibliotekos metodai

```
>>> import re
>>> re.findall('du', 'du kart du')
['du', 'du']
>>> pattern = re.compile('du')
>>> pattern.findall('du kart du')
['du', 'du']
>>> pattern.search('vienas')      # None
>>> pattern.search('du')
<_sre.SRE_Match object at 0x7f1e8c8493d8>
```

Specialieji simboliai:

<code>\d</code>	skaičius (<code>[0-9]</code>)
<code>\D</code>	ne skaičius (<code>[^0-9]</code>)
<code>\s</code>	tuščias simbolis (<i>whitespace</i>) (<code>[\t\n\r\f\v]</code>)
<code>\S</code>	ne tuščias simbolis (<code>[^\t\n\r\f\v]</code>)
<code>\w</code>	simbolis (<code>[a-zA-Z0-9_]</code>)
<code>\W</code>	ne simbolis (<code>[^a-zA-Z0-9_]</code>)

Reguliarių išraiškų sintaksė

<code>^</code>	simbolių eilutės pradžia
<code>\$</code>	simbolių eilutės pabaiga
<code>*</code>	0 arba daugiau simbolių
<code>+</code>	1 arba daugiau simbolių
<code>?</code>	0 arba 1 simbolis
<code>{m}</code>	m simbolių
<code>{m, n}</code>	nuo m iki n simbolių (nenurodymas m arba n žymi begalybę)
<code>\</code>	suteikti ypatingą prasmę toliau einančiam simboliui
<code>[]</code>	simbolių aibės apibrėžimas, pvz. <code>[0-9a-zA-Z_-]</code>
<code>[^]</code>	simbolis ne aibėje, jeigu aibė prasideda <code>^</code> simboliu
<code>()</code>	simbolių grupė

```
>>> regexp = '^ (brum+) $|^ (brum) +$|^ ([brum] +) $'
>>> re.findall(regexp, 'brum')          # [ ('brum', '', '') ]
>>> re.findall(regexp, 'brumbrum')     # [ ('', 'brum', '') ]
>>> re.findall(regexp, 'burm')         # [ ('', '', 'burm') ]
```

Reguliarių išraiškų sintaksė

| | Reguliarių išraiškų apjungimas

```
>>> regexp = '(\w+) day|(\w+) dien'
>>> result = re.findall(regexp, 'penkios dienos')
[(' ', 'penkios')]
```

(?P<name>...) | Vardo priskyrimas simbolių grupei

```
>>> regexp = '^/(?P<metai>\d+)/(?P<menuo>\d+)/(\d+)/$'
>>> result = re.search(regexp, '/2014/04/07/')
>>> result.group('metai')
'2014'
>>> res.groupdict()
{'metai': '2014', 'menuo': '04'}
>>> res.groups()
('2014', '04', '07')
```

Django URL sandara

```
from django.conf.urls import patterns, url
```

```
urlpatterns = patterns('',  
    url(r'^blog/2003/$', 'news.views.special_case_2003'),  
    url(r'^blog/(\d{4})/$', 'news.views.year_archive'),  
    url(r'^blog/(\d{4})/(\d{2})/$', 'news.views.month_arch'),  
    url(r'^blog/(\d{4})/(\d{2})/(\d+)/$', 'news.views.blog'),  
)
```

```
urlpatterns += patterns('',  
    url(r'^blog/(?P<year>\d{4})/$', 'news.views.sp_cases'),  
    url(r'^contact/', include('news.contact.views.urls')),  
)
```

- Django dokumentacija
- Django: urls

Django URL sandara

```
from django.conf.urls import patterns, url

urlpatterns = patterns('blog.views',
    url(r'^blog/(?P<year>\d{4})/$', 'year_archive',
        {'foo': 'bar'}, name='year-blog-archive'),
)
```

views.py:

```
from django.core.urlresolvers import reverse
from django.http import HttpResponseRedirect

def year_archive(request, year, foo):
    return HttpResponseRedirect(reverse('home'))
```

templates/*.html:

```
{% url 'year-blog-archive' 2003 %}
```