برنامه نویسی به زبان

پایتون

مرجع کامل

تالیف
مهندس هادی کیامرثی

تمام مثال های موجود در این کتاب با کامپیوتر تست شده اند تا از هر گونه خطا مبرا باشند با این حال ممکن است باز هم خطاهایی در آن وجود داشته باشد از کلیه خوانندگان این کتاب ، اساتید و دانشجویان محترم خواهشمندم برای مطلع کردن مولف از این خطا ها لطفا با ایمیل آدرس زیر تماس بگیرید

hadikiamarsi@gmail.com

فصل
دوازدهم

در این فصل مطالب زیر را خواهید آموخت

عبارات منظم (Regular Expression) در زبان برنامه نویسی پایتون (python)

تابع تطبیق (match) در زبان برنامه نویسی پایتون (python)

تابع جستجو (search)

تطبیق (Matching) در مقایسه با جستجو (Searching)

جستجو (Search) و جایگزینی (Replace)

تنظیمات برای عبارات منظم (Regular Expression)

الگوهای عبارات منظم (Regular Expression)

مثال های عبارات منظم (Regular Expression)

عبارات منظم (Regular Expression) در زبان برنامه نویسی پایتون (python)

عبارات منظم شامل الگوهایی برای توصیف رشته های زبان طبیعی می باشند . با عبارات منظم می توان هم تطبیق رشته ها را انجام داد ، هم جستجوی رشته ها را انجام داد و هم تولید رشته را می توان انجام داد . عبارات منظم در بیشتر زبان های مفسری مانند perl ، ruby و پایتون (python) وجود دارد .

تابع تطبیق (match) در زبان برنامه نویسی پایتون (python)

برای انجام عمل تطبیق رشته ها با عبارات منظم در زبان برنامه نویسی پایتون (python) از کلاس re به همراه متد match استفاده می گردد

لازم به ذکر است برای استفاده از این کلاس ابتدا باید مانند دستور زیر آن را در کد خود import نمایید

```
import re
```

ساختار نحوی آن در زیر آورده شده است

```
re.match(pattern, string, flags=0)
```

آرگومان های این متد در زیر آورده شده است

آرگومان ها به همراه توضیحات
Pattern

الگو یا همان عبارت منظم

String

رشته ای که عمل تطبیق در آن انجام می پذیرد به عبارتی رشته ای است که الگو در آن جستجو می گردد

Flags

تنظیمات

اگر متد `match` کلمه `none` را بر گرداند یعنی هیچ رشته ای را مطابق با الگو نیافته است

متدها به همراه توضیحات
group(num=0) تمام نتایج را به صورت لست بر می گرداند
groups() تمام نتایج را به صورت تاپل بر می گرداند

برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python
import re

line = "Cats are smarter than dogs"

matchObj = re.match( r'(.*) are (.*?) .*', line, re.M|re.I)

if matchObj:
   print "matchObj.group() : ", matchObj.group()
   print "matchObj.group(1) : ", matchObj.group(1)
   print "matchObj.group(2) : ", matchObj.group(2)
else:
   print "No match!!"
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
matchObj.group() :  Cats are smarter than dogs
matchObj.group(1) :  Cats
matchObj.group(2) :  smarter
```

تابع جستجو (search)

ایت تابع یک الگو را در تمام یک رشته جستجو می نماید و زیر رشته های مطابق با آن را بر می گرداند راهنمای دستور نحوی آن در زیر آورده شده است

```
re.search(pattern, string, flags=0)
```

آرگومان های ای متد در جدول زیر آورده شده است

آرگومان ها به همراه توضیحات
Pattern الگو یا همان عبارت منظم
String رشته ای که جستجو در آن صورت می پذیرد
Flags تنظیمات

اگر متد `search` کلمه `none` را بر گرداند یعنی هیچ رشته ای را مطابق با الگو نیافته است

تنظیمات به همراه توضیحات
group(num=0) تمام نتایج را به صورت لست بر می گرداند
groups() تمام نتایج را به صورت تاپل بر می گرداند

برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python
import re

line = "Cats are smarter than dogs";

searchObj = re.search( r'(.*) are (.*?) .*', line, re.M|re.I)

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!!"
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
searchObj.group() :  Cats are smarter than dogs
searchObj.group(1) :  Cats
searchObj.group(2) :  smarter
```

تطبیق (Matching) در مقایسه با جستجو (Searching)

تفاوت تطبیق و جستجو در عبارات منظم در این است که تطبیق فقط در ابتدای رشته انجام می گیرد و جستجو در نمام رشته انجام می پذیرد

برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python
import re

line = "Cats are smarter than dogs";

matchObj = re.match( r'dogs', line, re.M|re.I)
if matchObj:
   print "match --> matchObj.group() : ", matchObj.group()
else:
   print "No match!!"

searchObj = re.search( r'dogs', line, re.M|re.I)
if searchObj:
   print "search --> searchObj.group() : ", searchObj.group()
else:
   print "Nothing found!!"
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
No match!!
search --> matchObj.group() :  dogs
```

جستجو (Search) و جایگزینی (Replace)

عمل جایگزینی با استفاده از عبارات منظم در زبان برنامه نویسی پایتون (python) با استفاده از متد sub از کلاس re انجام می پذیرد

راهنمای نحوی

```
re.sub(pattern, repl, string, max=0)
```

این متد هر رشته ای که با الگو مطابق باشد را با رشته ای که برای آن در نظر گرفته شده باشد را جایگزین می نماید

برای آشنایی بیشتر با این درس به مثال زیر توجه نمایید

```
#!/usr/bin/python
import re

phone = "2004-959-559 # This is Phone Number"

# Delete Python-style comments
num = re.sub(r'#.*$', "", phone)
print "Phone Num : ", num

# Remove anything other than digits
num = re.sub(r'\D', "", phone)
print "Phone Num : ", num
```

اجرای کد بالا نتیجه زیر را در صفحه خروجی ظاهر خواهد نمود

```
Phone Num :  2004-959-559
Phone Num :  2004959559
```

تنظیمات برای عبارات منظم (Regular Expression)

کلاس re دارای تنظیماتی می باشد که این تنظیمات در جدول زیر آورده شده است

تنظیمات به همراه توضیحات
re.I حساسیت به حروف کوچک و بزرگ را فعال می کند
re.M اگر در رشته کاراکتر خط جدید (\n) و خط حامل (\r) وجود داشته باشد آن ها را با کاراکترهای ^ و $ جایگزین می نماید
re.S کاراکتر نقطه . (dot) را جایگزین کاراکتر خط جدید (\n) می کند
re.X برای خوانایی بیشتر اجازه استفاده از فضای خالی را می دهد

الگوهای عبارات منظم (Regular Expression)

الگوهای عبارات منظم از علامت هایی تشکیل شده است که معانی این علامت ها را در جدول زیر مشاهده می نمایید

الگوها به همراه توضیحات
^ برای بررسی مطابقت با ابتدای رشته بکار می رود
$ برای بررسی مطابقت با انتهای رشته بکار می رود
. مطابق با هر کاراکتری به جز کاراکتر خط جدید
[...] مطابق با هر کاراکتر تکی در براکتز
[^...] مطابق با هر کاراکتر تکی که در براکتز نیست
re*

جایگزین صفر یا بیشتر کاراکتر می باشد

re+

جایگزین یک یا بیشتر کاراکتر می باشد

re?

جایگزین صفر یا یک کاراکتر می باشد

re{ n}

مطابق با `n` عدد

re{ n,}

مطابق با `n` یا بیشتر

re{ n, m}

مطابق با حداقل `n` و حداکثر `m`

a| b

مطابق با `a` یا `b`

\w

مطابق با کلمه ای از کاراکترها

\W

مطابق با کاراکترهایی که کلمه نیستند

\s

برای فضای خالی مانند علامت های `\t\n\r\f` بکار می رود

\S

برای بدون فضای خالی بکار می رود

\d

برای اعداد `0` تا `9` بکار می رود

\D

برای کاراکترهایی بجز اعداد بکار می رود

\A

مطابق با شروع رشته

\Z

برای انتهای رشته در صورتی که کاراکتر خط جدید وجود داشته باشد در واقع برای کارکتر یکی مونده به آخر رشته بکار می رود
\z برای انتهای رشته بکار می رود
\G مطابق با نقطه ای که آخرین مطابقت رخ داده
\b مطابق با کاراکتر backspace در صورتی که درون براکت باشد
\n, \t, etc. مطابق با کارکترهای خط جدید و تب

مثال های عبارات منظم (Regular Expression)

مثال با توضیحات
Python مطابق با کلمه "python"

کاراکترها

مثال با توضیحات
[Pp]ython مطابق با کلمه Python یا python
rub[ye] مطابق با کلمه ruby یا rube
[aeiou] مطابق با یک حرف صدا دار انگلیسی
[0-9] مطابق با هر یک از اعداد 0 تا 9

[a-z] مطابق با هر کاراکتر اسکی حروف کوچک انگلیسی
[A-Z] مطابق با هر کاراکتر اسکی حروف بزرگ انگلیسی
[a-zA-Z0-9] مطابق با کاراکتری که شامل هر یک از حروف انگلیسی بزرگ و کوچک و اعداد است
[^aeiou] مطابق با هر کاراکتری به غیر از حروف صدادار انگلیسی
[^0-9] مطابق با هر کاراکتری به جز اعداد 0 تا 9

کاراکترهای خاص

مثال با توضیحات
. مطابق با هر کاراکتری به جز کاراکتر خط جدید
\d مطابق با عددها
\D مطابق با هر چیزی غیر از عدد
\s مطابق با کاراکتر فضای خالی
\S مطابق با هر چیزی غیر از کاراکتر فضای خالی
\w مطابق با یک کلمه که شامل حروف و اعداد باشد
\W مطابق با یک کلمه که شامل حروف و اعداد نباشد

حالت های تکرار

مثال با توضیحات
Ruby? این الگو مطابق با کلمه rub یا ruby و حرف y اختیاری می باشد
Ruby* این الگو مطابقت می کند با کلمه rub و هیچ یا بیشتر حرف y
Ruby+ این الگو مطابقت می کند با کلمه rub و یک یا بیشتر حرف y
\d{3} این الگو دقیقا با 3 عدد مطابق هست
\d{3,} این الگو مطابق با 3 رقم یا بیشتر
\d{3,5} این الگو با 3 و 4 یا 5 عدد

گروه بندی با پرانتزها (Parentheses)

مثال با توضیحات
\D\d+ گروهی وجود ندارد پس فقط علامت \d تکرار می گردد
(\D\d)+ گروه وجود دارد پس جفت علامت \D\d با هم تکرار می گردند
([Pp]ython(,)?)+ ابن الگو تطابق داره با ,"Python", "Python, python, python"

گروه بندی با براکتز

مثال با توضیحات
([Pp])ython&\1ails این الگو با python&pails یا Python&Pails مطابق است

جایگزین ها (Alternatives)

مثال با توضیحات
python|perl این الگو با python یا perl مطابق است
rub(y|le)) این الگو با ruby یا ruble مطابق است
Python(!+|\?) این الگو مطابق هست با شروع با Python و در ادامه یک یا بیشتر علامت ! یا یک علامت ؟

لنگرها (Anchors)

مثال با توضیحات
^Python کلمه ابتدایی رشته با کلمه Python شروع شود
Python$ کلمه انتهایی رشته با کلمه Python شروع شود
\APython

کلمه ابتدایی رشته با کلمه Python شروع شود

Python\Z

کلمه انتهایی رشته با کلمه Python شروع شود