Regex در پایتون

عطیه سادات طباطبایی
دانشجو علوم کامپیوتر
دانشگاه تهران
atiehtaba82@gmail.com

Regex، مخفف عبارت Regular Expression، دنباله‌ای از کاراکترها است که یک الگوی جستجو را تعریف می‌کنند. ریجکس یک ابزار قدرتمند برای پردازش متن و تطبیق الگو است.

زبان‌های برنامه نویسی مانند java Python و Perl از ریجکس پشتیبانی می‌کنند. این کتابخانه معمولاً در اعتبارسنجی داده‌ها، تجزیه متن و عملیات جستجو و جایگزینی استفاده می‌شود.

ریجکس در واقع ترکیبی از کاراکترهای خاص است که الگویی را تعریف می‌کنند که با مجموعه خاصی از کاراکترها مطابقت دارد.

در ادامه ابتدا به معرفی قواعد الگوسازی ریجکس می‌پردازیم و سپس چند مثال را با هم می‌بینیم.

در جدول زیر برخی از پرکاربرد ترین قواعد این کتابخانه را مشاهده می‌کنید:

Quantifiers	
a|b	Match either "a" or "b"
?	Match either "a" or "b"
+	One or more
*	Zero or more
*?	Zero or more, but stop after first match
{N}	Exactly N number of times (Where N is number)
{N, M}	From N to M number of times (Where N and M are numbers)

General Tokens	
.	Any character
\n	Newline character
\t	Tab character
\s	Any whitespace character (Including \t, \n, etc)
\S	Any non-whitespace character
\w	Any word character (Upper/lowercase letters, 0-9, _)
\W	Any non-word character
\b	Word boundary (Matches between characters)
\B	Non-word boundary
^	The start of a line
$	The end of a line
\\	The literal character "\"

برای مثال "[A-Za-z]+" یک الگو است که با هر رشته‌ای که فقط از حروف تشکیل شده مطابقت دارد.

برای استفاده از ریجکس در زبان پایتون از دستور زیر استفاده می‌کنیم.

```
import re
```

همان طور که گفته شد از ریجکس می توان برای جستجو، جایگزینی و تطابق الگو استفاده کرد. در این راستا به معرفی سه تابع این کتابخانه می پردازیم:

تابع search :

```
txt="some text"
x=re.search("your pattern",txt)
```

خروجی قطعه کد بالا شامل یک رشته و یک تاپل است که به ترتیب عبارت تطابق داده شده و محل شروع و پایان عبارت مذکور را در صورت وجود بر می‌گرداند . اگر جستجو نتیجه‌ای نداشته باشد و عبارت مورد نظر در متن نباشد خروجی None خواهد بود.

حال فرض کنیم می‌خواهیم وجود یک عبارت یا یک الگو را در رشته دلخواه بررسی کنیم. برای مثال فرض کنیم می‌خواهیم ببینیم آیا کلمه "Python" در یک رشته وجود دارد یا خیر:

```
import re

string = "Python is fun"

match = re.search('Python', string)

if match:
  print("pattern found inside the string")
else:
  print("pattern not found")

# Output: pattern found inside the string
```

تابع find all :

```
txt="some text"
x=re.findall("your pattern",txt)
```

دستور بالا یک لیست شامل تمامی عبارت‌هایی که در متن با الگو تطابق دارند را خروجی می‌دهد.

برای مثال فرض کنیم می خواهیم همه اعداد استفاده شده در یک رشته را

استخراج کنیم ؛ با توجه جدول بالا الگوی مورد نظر ما "+[۰-۹]" خواهد بود بنابراین طبق دستور زیر عمل می‌کنیم:

```
import re

string = 'The first release candidate of Python 3.12 was offered on 6 August 2023.'
pattern = '[0-9]+'

result = re.findall(pattern, string)
print(result)

# Output: ['3', '12', '6',2023]
```

تابع sub :

```
txt="some text"
x=re.sub("your pattern","new string",txt)
```

با استفاده از دستور بالا می‌توانید یک عبارت جدید را جایگزین الگوی تعریف شده کنید.

برای مثال فرض کنیم می‌خواهیم همه فاصله‌های یک رشته را حذف کنیم. برای این کار باید همه فاصله‌ها را با یک رشته خالی جایگزین کنیم:

```
import re

string = 'H e l l o W o r l d !'
# matches all whitespace characters
pattern = '\s+'
# empty string
replace = ''

new_string = re.sub(pattern, replace, string)
print(new_string)

# Output: HelloWorld!
```

حال برای درک بهتر کاربردregex چند مثال را با هم بررسی می‌کنیم

مطابقت فرمت تاریخ (yyyy/mm/dd):

فرض کنیم از کاربر خواسته‌ایم تا تاریخ تولدش را مطابق فرمت بالا وارد کند. فرض می‌کنیم فقط فرمت بالا را به عنوان تاریخ می‌پذیریم (چهارعدد که نشان دهنده سال است و دو جفت دوعدد که به ترتیب نشان دهنده ماه و روز هستند و اعداد هر قسمت با کاراکتر "/" از هم جداشده اند مانند ۲۰۲۳/۰۵/۱۰). می‌خواهیم اگر کاربر تاریخ تولدش را مطابق الگوی خواسته شده وارد نکند ارور دریافت کند. خواهیم داشت :

```
import re
user_birthdate=input()
x=re.search("[0-9]{4}/[0-9]{2}/[0-9]{2}",user_birthdate)

if not x:
    print("pleas enter your birthdate correctly")
```

اعتبار سنجی آدرس ایمیل:

فرض کنید از کاربر خواسته‌ایم تا آدرس ایمیل خود را وارد کند. می‌خواهیم ببینیم آیا عبارتی که کاربر وارد کرده می‌تواند یک آدرس ایمیل باشد؟

برای این کار ابتدا یک الگو تعریف می‌کنیم. اگر الگوی تعریف شده با عبارت وارد شده توسط کاربر مطابقت داشت رشته ورودی می‌تواند یک آدرس ایمیل باشد در غیر این صورت عبارت وارد شده معتبر نخواهد بود.

برای تعریف الگو ابتدا به ساختار یک آدرس ایمیل دقت می‌کنیم . در یک آدرس ایمیل ابتدا مجموعه‌ای از کاراکترها که می‌توانند شامل عدد، حروف و یا کاراکترهای دیگر باشند کنار هم قرار می گیرند سپس علامت "@" می آید و پس از آن نیز مجددا مجوعه‌ای از کاراکترها می آید و در آخر نیز دامنه سایت که با "." شروع می‌شود :

```
import re
addres=input()
x=re.search("[A-Za-z0-9.+_-]+@[A-Za-z0-9.+_-]+\.[A-Za-z0-9.+_-]+",addres)
if x:
 print("valid email addres")
else:
 print("invalid email addres")
```

دقت کنید چون "." جزء قواعد ریجکس است و برای تطابق هر کاراکتر دلخواه به شمار می‌رود، اگر بخواهیم از آن به عنوان خود کاراکتر "." استفاده کنیم قبل از آن از علامت "\" استفاده می‌کنیم.

سخن پایانی :

کار با ریجکس در نگاه اول می‌تواند دشوار به نظر برسد اما با تمرین می‌توانید از قابلیت‌های این کتابخانه قدرتمند بهره ببرید.

برای مطالعه بیشتر می تانید به آدرس https://www.w۳schools.com/python/python_regex.asp رجوع کنید.