**Regular Expressions (REGEX)**

Regular expression (regex olarak kısaltılır), metinde bir arama düzenini belirten bir karakter dizisidir. Genellikle bu tür kalıplar, dizi arama algoritmaları tarafından dizilerdeki "bul" veya "bul ve değiştir" işlemleri veya girdi doğrulaması için kullanılır.

Regular expression, özel biçimleri belirtmek veya özel anlamlarını çağırmadan özel karakterlerin kullanılmasına izin vermek için ters eğik çizgi karakterini ('\') kullanır. Python'un Regular expression pattternleri için ham dize gösterimi, özel karakterlerin kullanılması sorununu çözer. r"\n", '\' ve 'n' içeren iki karakterli bir dizedir, "\n" ise yeni bir satır içeren tek karakterli bir dizedir.

**Common Patterns**

\d 0'dan 9'a kadar herhangi bir numeric digit

\D decimal digit olmayan herhangi bir karakterle eşleşir. Bu, \d'nin tersidir.

\w Herhangi bir harf, sayısal rakam veya alt çizgi karakteri. (Bunu eşleşen "word" karakterleri olarak düşünün.)

\W Harf, sayısal rakam veya alt çizgi karakteri olmayan herhangi bir karakter.

\s Herhangi bir boşluk, sekme veya yeni satır karakteri. (Bunu white-space karakterleriyle eşleşen olarak düşünün.)

\S Boşluk, sekme veya yeni satır olmayan herhangi bir karakter.

**Common Metacharacters**

"[]" Bir dizi karakter "[a-m]" gibi

"\" Özel bir diziyi işaret eder (özel karakterlerden kaçmak için de kullanılabilir)

"." Herhangi bir karakter (newline karakteri hariç)

"^" Starts with "^hello" gibi

"$" Ends with "world$" gibi

"\*" Zero or more occurrences

"+" One or more occurrences

"{}" Exactly the specified number of occurrences

"|" Either or "falls|stays" gibi

"()" Capture and group

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Raw String ("r / R")**

Python raw string, bir string literal önüne 'r' veya 'R' eklenerek oluşturulur.

Python raw string, backslah (\) literal bir karakter olarak ele alır. Bu, ters eğik çizgi içeren bir string sahip olmak istediğimizde ve bunun bir kaçış karakteri olarak ele alınmasını istemediğimizde kullanışlıdır.

print("Backslash: \\") ------ Backslash: \

print(r"Backslash: \\") ----- Backslash: \\

**Common Python RegEx Functions**

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

Patternle eşleşme arayarak dizeyi tarayın, bir Match nesnesi döndürün veya eşleşme bulunamazsa None dondurur

text = "A78L41K"

num = re.search("78", text)

num.group() ---->> '78'

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

Pattreni string başında uygulamayı deneyin, bir Match nesnesi döndürün veya eşleşme bulunmazsa None dondrur

Dize içinde herhangi bir yerde bir eşleşme bulmak istiyorsanız, match() yerine search() kullanın.

text = "A78L41K"

alpha = re.match("\D\d\d", text)

alpha.group() ----->> 'A78'

re.fullmatch(pattern, string, flags=0)

Pattreni tüm dizeye uygulamak için

alpnum = re.fullmatch("\D\d+\D\d+\D", text)

alpnum.group() --->> 'A78L41K'

re.findall(pattern, string, flags=0)

Dizedeki tüm örtüşmeyen eşleşmelerin bir listesini döndürür.

text = "O 1, t 10, o 100. 100000"

re.findall("\d{1}", text) ---------->>  ['1', '1', '0', '1', '0', '0', '1', '0', '0', '0', '0', '0']

re.sub(pattern, repl, string, count=0, flags=0)

Dizedeki desenin en soldaki örtüşmeyen tekrarlarını değiştirme repl ile değiştirerek elde edilen dizeyi döndürün. repl bir dize veya çağrılabilir olabilir; bir dize varsa, içindeki ters eğik çizgi işlenir. Çağrılabilirse, Match nesnesinden geçirilir ve kullanılacak yeni bir dize döndürmesi gerekir.

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

re.sub("\D", "", text) ----->> '2004959559'

re.split(pattern, string, maxsplit=0, flags=0)

Kaynak dizeyi, ortaya çıkan alt dizeleri içeren bir liste döndürerek, kalıbın oluşumlarına göre bölün.

text = "ab56cd78\_de fg3hıi49"

re.split("\D+", text) ------>> ['', '56', '78', '3', '49']

**Pandas Functions Accepting RegEx**

[jupyter notebook#1](file:///C:\Users\Arslan\Desktop\GitHub\daily_work_file\08.18.2022)

metin içeren bir resim

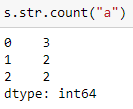
Açıklama otomatik olarak oluşturuldu

**count()**: Seri/İndeks'in her dizesindeki pattern oluşumlarını sayın

pandas.Series.str.count(pat, flags=0)

Seri/İndeks'in her dizesindeki pattern oluşumlarını sayın

Bu işlev, Serinin string öğelerinin her birinde belirli bir normal ifade modelinin tekrarlanma sayısını saymak için kullanılır.

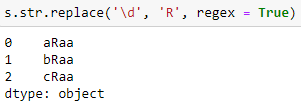


**replace()**: Arama dizesini veya patterni verilen değerle değiştirin

pandas.Series.str.replace(pat, repl, n=- 1, case=None, flags=0, regex=None)

Seri/Dizin içindeki her pattern/normal ifade oluşumunu değiştirin.

Normal ifade değerine bağlı olarak str.replace() veya re.sub() ile eşdeğerdir.



**contains()**: Bir Dizi veya Dizin dizesinde kalıp veya normal ifadenin içerilip içermediğini test edin. re.search()'i çağırır ve bir boole döndürür

pandas.Series.str.contains(pat, case=True, flags=0, na=None, regex=True)

Bir Dizi veya Dizin dizesinde pattern veya regex ifade olup olmadığını test edin.

Belirli bir patternin veya regex ifadenin bir series veya index dizesi içinde bulunup bulunmadığına bağlı olarak Boole Serisi veya index döndürür.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**findall():** Seri/Dizin içindeki tüm pattern veya regular expression oluşumlarını bulun. Tüm öğelere re.findall() uygulamasına eşdeğerdir

pandas.Series.str.findall(pat, flags=0)

Seri/İndeks'te tüm pattern veya egular expression oluşumlarını bulun.

Dizi/Dizin içindeki tüm öğelere re.findall() uygulamasına eşdeğerdir.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

match(): Her dizenin bir regular expression eşleşip eşleşmediğini belirleyin. re.match() işlevini çağırır ve bir boole değeri döndürür

pandas.Series.str.match(pat, case=True, flags=0, na=None)

Her dizenin bir regular expression başlayıp başlamadığını belirleyin.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

**split()**: Dizeleri verilen ayırıcı/sınırlayıcı çevresinde böler ve bölünecek string veya regular expression kabul eder

pandas.Series.str.split(pat=None, n=- 1, expand=False, \*, regex=None)

Dizeleri verilen ayırıcı/sınırlayıcı çevresinde bölün.

Seri/İndex içindeki dizeyi, belirtilen sınırlayıcı dizesinde baştan böler.

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

extract (): Normal ifadedeki yakalama gruplarını bir DataFrame'deki sütunlar olarak ayıklayın ve yakalanan grupları döndürür

pandas.Series.str.extract(pat, flags=0, expand=True)

Normal ifadedeki yakalama gruplarını bir DataFrame'de sütunlar olarak çıkarın.

Dizideki her konu dizesi için, düzenli ifade pat'ın ilk eşleşmesinden grupları çıkarın.

tablo içeren bir resim

Açıklama otomatik olarak oluşturuldu