Regex

(regular expression)

Content:

Introduction to Regular Expression

Regex
(regular expression)

- Regular expressions are **patterns** used to match character combinations in String.
  - **Pattern** is composed of simple characters.
    - Eg: /abc/
  - Combination of **simple** and **special** characters.
    - Eg: /ab*c/

Regex

(regular expression)

Meta Characters

- Special Characters
- Search for a match something more than a direct match.

Special Sequence

- Escaping Characters
- These characters search for a match something specific characters

Meta Characters

Characters	Description	Example	Match
|	Either or	“cat|hello	
.	Any character (one) except new line	“he..lo”	Match the characters between e and l in the example
{ }	Exactly the specified number of occurrence	“he.{2}o”	
*	Zero or more occurrence	“he.*o”	Find the word that starts with 'he,' followed by zero or more characters, and ends with ‘o. Eg: hello, hero, **Helico**
+	one or more occurrence	“he.+o”	Find the word that starts with 'he,' followed by one or more characters, and ends with ‘o. Eg: hello, hero, **Helico**
?	Zero or one occurrence	“he.?o”	Eg. Hero
( )	Capture and group		
[ ]	A set of characters	“[a-z]”	abcdefghijklmnopqrstuvwxyz
^	Starts with	“^hello”	Starts with hello
$	Ends with	“world$”	Sentence ends with World
\	Signals a special sequence (Can also be used to escape special characters)	“\d”	Match all integers in string

Special Sequences

Characters	Description	Example
\w	Returns a match where the string contains any word characters (characters from a to z, digits from 0-9, and the underscore _ characters)	"\w"
\W	Returns a match where the string **DOES NOT** contain any word characters	"\W"
\b	Returns a match where the specified characters are at the beginning or at the end of a word.	r"\bello" r"ello\b"
\B	Returns a match where the specified characters are present, but **NOT** as the beginning of a word (or at the end)	r"\Bello" r"ello\B"
\d	Returns a match where the string contains digits (number from 0-9)	"\d"
\D	Returns a match where the string **DOES NOT** contains digits	"\D"
\s	Returns a match where the string contains a white space characters	"\s"
\S	Return a match where the string **DOES NOT** contains a white space characters	"\S"
\A	Returns a match if specified characters are at the beginning of the string.	r"\AThe"
\Z	Returns a match if the specified characters are at the end of the string	"Spain\Z"

Example String

Once upon a time in a colorful town, a young coder started learning Regex. He practiced on words like apple, Banana, and cherry. His teacher said, ‘Numbers like 12345 and 42 are easy to match using special patterns.’ So, he tried a regex that found any digit. One day, he saw a sign that said Start here and wondered if regex could match words that start with something. Later, he found a note that ended with End here. and thought about matching the end of a sentence. He also learned that . could match any character, like in ‘cat’ or ‘cet’. When testing, he noticed some words like hero, helicopter, he, and heo, which made him curious about zero or more characters in between letters. He wrote a regex that could match one or more characters after a pattern. Another tricky test was checking if color and colour could be matched with an optional letter. He then practiced with words like ‘hellooo’ to match exactly two ‘o’s. Finally, he saw two people falling and staying, so he tested an OR condition. His final challenge was repeating words like ‘repeated repeated’ and checking if regex could capture and group them correctly. In the end, he mastered Regex and built powerful search tools!

/<pattern>/

In python

r"<pattern>"

/cat/

match the pattern “cat”
Return which range this pattern exists

Example

The quick brown fox jumps over the lazy dog. This is outside (this is inside)

SNO	QUESTION	ANSWER
1	Match the string **“fox”** and provide its range	16-19
2	How many times does **“is”** appear in above string ?	4
3	Match the pattern **“(this is inside)”** and provide its range	61-77

Meta Characters

Characters	Description	Example	Match
|	Either or	"cat|hello	
.	Any character (one) except new line	"he..lo"	Match the characters between e and l in the example
{ }	Exactly the specified number of occurrence	"he.{2}o"	
*	Zero or more occurrence	"he.*o"	Find the word that starts with 'he,' followed by zero or more characters, and ends with 'o. Eg: hello, hero, **Helico**
+	one or more occurrence	"he.+o"	Find the word that starts with 'he,' followed by one or more characters, and ends with 'o. Eg: hello, hero, **Helico**
?	Zero or one occurrence	"he.?o"	Eg. Hero
( )	Capture and group		
[ ]	A set of characters	"[a-z]"	abcdefghijklmnopqrstuvwxyz
^	Starts with	"^hello"	Starts with hello
$	Ends with	"world$"	Sentence ends with World
\	Signals a special sequence (Can also be used to escape special characters)	"\d"	Match all integers in string

| (pipe)

/cat|cherry/

match the pattern “**cat**” or “**cherry**”
Return which range this pattern exists

Exercise

The sun rises in the east and sets in the west. Birds sing in the morning or evening.

SNO	QUESTION	PATTERN
1	Write a regex pattern to match either **“sun”** or **“moon”** in the string	sun|moon
2	Find a regex pattern to match either **“east”** or **“west”**	east|west
3	Write a regex pattern to match either **“morning”** or **“evening”**	morning|evening
4	Create a regex pattern to match either **“rises”, “sets”,** or **“sing”**	rises|sets|sing
5	Find regex pattern to match either **“The”** or **“Birds”**	The|Birds

Meta Characters

Characters	Description	Example	Match
|	Either or	“cat|hello	
.	Any character (one) except new line	“he..lo”	Match the characters between e and l in the example
{ }	Exactly the specified number of occurrence	“he.{2}o”	
*	Zero or more occurrence	“he.*o”	Find the word that starts with 'he,' followed by zero or more characters, and ends with ‘o. Eg: hello, hero, **Helico**
+	one or more occurrence	“he.+o”	Find the word that starts with 'he,' followed by one or more characters, and ends with ‘o. Eg: hello, hero, **Helico**
?	Zero or one occurrence	“he.?o”	Eg. Hero
( )	Capture and group		
[ ]	A set of characters	“[a-z]”	abcdefghijklmnopqrstuvwxyz
^	Starts with	“^hello”	Starts with hello
$	Ends with	“world$”	Sentence ends with World
\	Signals a special sequence (Can also be used to escape special characters)	“\d”	Match all integers in string

● (dot)

Matches any character (one) expect new line

- Alphabets
- Special characters
- Numbers
- Escape characters (except new line [\n])

/c.t/

cat	cAt	c0t	c~t
cbt	cBt	c1t	c!t
cct	cCt	c2t	c@t
cdt	cDt	c3t	c#t
cet	cEt	c4t	c$t
cft	cFt	c5t	c%t
…	…	…	…
czt	cZt	c9t	

match the pattern “c<any character>t”

Return which range this pattern exists

Warning

- Dot (.) is very powerful metacharacter that can **create problem** if it is not use properly.

Data Science Anywhere | MARS Learnings

Meta Characters

Characters	Description	Example	Match
|	Either or	"cat|hello	
.	Any character (one) except new line	"he..lo"	Match the characters between e and l in the example
[ ]	A set of characters	"[a-z]"	abcdefghijklmnopqrstuvwxyz
{ }	Exactly the specified number of occurrence	"he.{2}o"	
*	Zero or more occurrence	"he.*o"	Find the word that starts with 'he,' followed by zero or more characters, and ends with 'o. Eg: hello, hero, **Helico**
+	one or more occurrence	"he.+o"	Find the word that starts with 'he,' followed by one or more characters, and ends with 'o. Eg: hello, hero, **Helico**
?	Zero or one occurrence	"he.?o"	Eg. Hero
( )	Capture and group		
^	Starts with	"^hello"	Starts with hello
$	Ends with	"world$"	Sentence ends with World
\	Signals a special sequence (Can also be used to escape special characters)	"\d"	Match all integers in string

[ ]

Character set or character classes

- Allows to define a character that will match if any of the defined characters on the set is present.

Element	Description
[cs]	/ licen[cs]e/ $\longrightarrow$ licence, license
[0-9]	Matches anything between 0 and 9 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
[a-z]	Matches anything between a and z (a, b, c, d, …, z)
[A-Z]	Matches anything between A and Z (A, B, C, D …, Z)
[0-9a-zA-Z]	Any lowercase or uppercase alphanumeric character

licen**c**e
licen**s**e

caret (^)

Negation of range

- Allows to define a character that will match if any of the defined characters on the set is present.

Element	Description
[^0-9]	Will match anything that is not a digit
[^a-z]	Will match anything that is not a lowercase alphabets
[^A-Z]	Will match anything that is not a uppercase alphabets
[^0-9a-zA-Z]	Will match anything that is not lowercase or uppercase alphanumeric character

Exercise Question

Contact Information:
John Doe - john.doe@example.com - (555) 123-4567
Mary Smith - mary_smith@email.net - 555.987.6543
Tom Johnson - tom-johnson@company.org - (555)246-8910
Sarah Brown - sarah@brown.co.uk - +1-555-369-7412
Mike Wilson - mike.wilson@subdomain.example.edu - 555 741 0258

SNO	QUESTION	PATTERN
1	Write a regular expression using character sets ([]) to match any single vowel (a, e, i, o, u) in the text.	[aeiou]
2	Write a regular expression using pipe (|) to match either "John" or "Tom" in the text.	John|Tom
3	Write a regular expression using caret (^) inside character sets to match any single character that is NOT a digit in the text.	[^0-9]
4	Write a regular expression to match either "com" or "net" in the email domains.	com|net
5	Write a regular expression using character sets to match any single digit in the phone numbers.	[0-9]
6	Write a regular expression using the dot (.) to match any character between 'T' and 'm' in "Tom".	T.m
7	Write a regular expression using character sets to match any single uppercase letter in the text.	[A-Z]
8	Write a regular expression using caret (^) inside character sets to match any single character that is not a letter or number in the text.	[^0-9a-zA-Z]

Quantifiers

- The mechanism to define how a character, metacharacter, or character set can be repeated.

Symbol	Name	Quantification of previous character
?	Question Mark	0 or 1 repetitions
*	Asterisk	Zero or more times
+	Plus sign	One or more times
{n,m}	Curly braces	Between n and m times

Example of Quantifiers

/hello*/

hell
hello
helloo
hellooo
helloooo
hellooooo
...

/hello+/

hello
helloo
hellooo
helloooo
hellooooo
...

/hello?/

hell
hello

/hello{2,5}/

helloo
hellooo
helloooo
hellooooo

Data Science Anywhere | MARS Learnings

Special Sequences – (pre defined characters)

Characters	Description	Similar to
\w	Returns a match where the string contains any word characters (characters from a to z, digits from 0-9, and the underscore _ characters)	[a-zA-Z0-9_]
\W	Returns a match where the string **DOES NOT** contain any word characters	[^a-zA-Z0-9_]
\d	Returns a match where the string contains digits (number from 0-9)	[0-9]
\D	Returns a match where the string **DOES NOT** contains digits	[^0-9]
\s	Returns a match where the string contains a white space characters	[\t\n\r\f\v]
\S	Return a match where the string **DOES NOT** contains a white space characters	[^ \t\n\r\f\v]

Boundary Matchers

- Boundary matchers are a identifiers that will correspond to a particular position inside of the input.

Matcher	Description
^	Matches at the beginning of a line
$	Matches at the end of a line
\b	Matches a word boundary. Matches both beginning and ending.
\B	Matches the opposite of \b. Anything that is not a word boundary
\A	Matches the beginning of the input
\Z	Matches the end of the input

Example String

```
Hello world! This is line one.
World, hello! This is line two.
HelloWorld is a single word.
The word "hello" appears in quotes.
This line ends with hello
hello starts this line and world ends it with world
com.example.domain is a domain name
user@example.com is an email address.
2023-05-15 is a date format.
The final line ends the entire text.
```

DATA
SCIENCE
ANYWHERE

MARS
Learnings

https://www.youtube.com/@datascienceanywhere/

https://www.udemy.com/user/freeai-space/

https://github.com/marslearnings