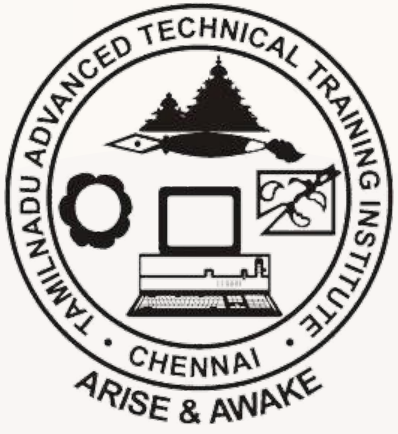

Regular Expressions with Python

Regular Expressions

- A sequence of characters that forms a search pattern.
- Used to check if a string contains the specified search pattern.
- Python has a dedicated module named RegEx.

Example of a Regular Expression

Expression	String	Matched?
^a...s$	abs	No match
	alias	Match
	abyss	Match
	Alias	No match
	An abacus	No match

RE Module

Syntax:

re.search(<regex>, <string>)

Metacharacters

Character(s)	Meaning
.	Matches any single character except newline
^	· Anchors a match at the start of a string · Complements a character class
$	Anchors a match at the end of a string
*	Matches zero or more repetitions
+	Matches one or more repetitions
?	· Matches zero or one repetition · Specifies the non-greedy versions of *, +, and ? · Introduces a lookahead or lookbehind assertion · Creates a named group

Metacharacters

Character(s)	Meaning
{ }	Matches an explicitly specified number of repetitions
\	· Escapes a metacharacter of its special meaning · Introduces a special character class · Introduces a grouping backreference
[ ]	Specifies a character class
|	Designates alternation
( )	Creates a group

[] - Square brackets

Square brackets specifies a set of characters you wish to match.

Expression	String	Matched?
[abc]	a	1 match
	ac	2 matches
	Hey Jude	No match
	abc de ca	5 matches

. - Period

A period matches any single character (except newline '\n').

Expression	String	Matched?
..	a	0 match
	ac	1 matches
	acd	1 match
	acde	2 matches

^ - Caret

A caret checks if a string starts with a certain character.

Expression	String	Matched?
^a	a	1 match
	abc	1 match
	bac	No match
^ab	abc	1 match
	acb	No match

$ - Dollar

The dollar symbol $ is used to check if a string ends with a certain character.

Expression	String	Matched?
a$	a	1 match
	formula	1 match
	cab	No match

* - Star

The star symbol * matches zero or more occurrences of the pattern left to it.

Expression	String	Matched?
ma*n	mn	1 match
	man	1 match
	maaan	1 match
	main	No match
	woman	1 match

+ - Plus

The plus symbol + matches one or more occurrences of the pattern left to it.

Expression	String	Matched?
ma+n	mn	No match
	man	1 match
	maaan	1 match
	main	No match
	woman	1 match

? - Question Mark

The question mark symbol ? matches zero or one occurrence of the pattern left to it.

Expression	String	Matched?
ma?n	mn	1 match
	man	1 match
	maaan	No match
	main	No match
	woman	1 match

{ } - Braces

Takes 2 Parameters: a{n,m}

where n = least repetitions and m = most repetitions of the pattern on left.

Expression	String	Matched?
a{2, 3}	abc dat	No match
	abc daat	1 match
	aabc daaat	2 matches
	aabc daaaat	2 matches

| - Alternation(OR)

Vertical bar | is used for alternation (or operator).

Expression	String	Matched?
a|b	cde	No match
	ade	1 match
	acdbea	3 matches

( ) - Group

Parentheses () is used to group sub-patterns.

Expression	String	Matched?
(a|b|c)xz	ab xz	No match
	abxz	1 match
	axz cabxz	2 matches

\ - Backslash

- Backlash \ is used to escape various characters including all metacharacters.

- Example: ('\.', 'hello.world')

Anchors in RegEx

Anchors

- Special characters that match positions.
- Helpful in extracting a particular data and searching patterns within a large text collection.

\A

Matches if the specified characters are at the start of a string.

Expression	String	Matched?
\Athe	the sun	Match
	In the sun	No match

\b

Matches if the specified characters are at the beginning or end of a word.

Expression	String	Matched?
\bfoo	football	Match
	a football	Match
	afootball	No match
foo\b	the foo	Match
	the afoo test	Match
	the afootest	No match

\B

Matches if the specified characters are **NOT** at the beginning or end of a word.

Expression	String	Matched?
\Bfoo	football	No match
	a football	No match
	afootball	Match
foo\B	the foo	No match
	the afoo test	No match
	the afootest	Match

\d

Matches any decimal digit. Equivalent to [0-9]

Expression	String	Matched?
\d	12abc3	3 matches
	Python	No match

\D

Matches any non-decimal digit.

Expression	String	Matched?
\D	1ab34"50	3 matches (at 1ab34"50)
	1345	No match

\s

Matches where a string contains any whitespace character.

Expression	String	Matched?
\s	Python RegEx	1 match
	PythonRegEx	No match

\S

Matches where a string DOES NOT contain any whitespace character.

Expression	String	Matched?
\S	a b	2 matches
		No match

\w

- Matches any alphanumeric character
- Equivalent to [a-zA-Z0-9_]

Expression	String	Matched?
\w	12&": ;c	3 matches
	%"> !	No match

\W

- Matches any non-alphanumeric character.
- Equivalent to [^a-zA-Z0-9_]

Expression	String	Matched?
\W	1a2%c	1 match
	Python	No match

\Z

Matches if the specified characters are at the end of a string.

Expression	String	Matched?
Python\Z	I like Python	1 match
	I like Python Programming	No match
	Python is fun.	No match

RegEx Functions in Python

Searching Functions in RegEx

Function	Description
re.search()	Scans a string for a regex match
re.match()	Looks for a regex match at the beginning of a string
re.fullmatch()	Looks for a regex match on an entire string
re.findall()	Returns a list of all regex matches in a string
re.finditer()	Returns an iterator that yields regex matches from a string

Substitution Functions

Function	Description
re.sub()	Scans a string for regex matches, replaces the matching portions of the string with the specified replacement string, and returns the result
re.subn()	Behaves just like re.sub() but also returns information regarding the number of substitutions made

Utility Functions

Function	Description
re.split()	Splits a string into substrings using a regex as a delimiter
re.escape()	Escapes characters in a regex

Regular Expression Flags

Flags

Flag	Long Syntax	Meaning
re.A	re.ASCII	Perform ASCII-only matching instead of full Unicode matching
re.I	re.IGNORECASE	Perform case-insensitive matching
re.M	re.MULTILINE	This flag is used with metacharacter ^ (caret) and $ (dollar). When this flag is specified, the metacharacter ^ matches the pattern at beginning of the string and each newline's beginning (\n). And the metacharacter $ matches pattern at the end of the string and the end of each new line (\n)

Flags

Flag	Long Syntax	Meaning
re.S	re.DOTALL	Make the DOT (.) special character match any character at all, including a newline. Without this flag, DOT(.) will match anything except a newline
re.X	re.VERBOSE	Allow comment in the regex. This flag is useful to make regex more readable by allowing comments in the regex.
re.L	re.LOCALE	Perform case-insensitive matching dependent on the current locale. Use only with bytes patterns

re.U	re.UNICODE	Enables unicode matching.