

Python Regular Expressions

SINGLE CHARACTERS

Use	To match any character
[<i>set</i>]	In that set
[^ <i>set</i>]	Not in that set
[<i>a-z</i>]	In the <i>a-z</i> range
[^ <i>a-z</i>]	Not in the <i>a-z</i> range
.	Any except \n (new line)
\char	Escaped special character

CONTROL CHARACTERS

Use	To match	Unicode
\t	Horizontal tab	\u0009
\v	Vertical tab	\u000B
\b	Backspace	\u0008
\e	Escape	\u001B
\r	Carriage return	\u000D
\f	Form feed	\u000C
\n	New line	\u000A
\a	Bell (alarm)	\u0007

NON-ASCII CODES

Use	To match character with
\octal	First digit 0 followed by 2 octal digits or 3 octal digits
\x hex	2-digit hex character code
\u hex	4-digit hex character code

CHARACTER CLASSES

Use	To match character
\w	Word character. [0-9_a-zA-Z] and Unicode word characters
\W	Non-word character
\d	Decimal digit and Unicode digits
\D	Not a decimal digit
\s	White-space character [\t\n\r\f\v] and Unicode spaces
\S	Non-white-space char

QUANTIFIERS

Greedy	Lazy	Matches
*	*?	0 or more times
+	+?	1 or more times
?	??	0 or 1 time
{ <i>n</i> }	{ <i>n</i> }?	Exactly <i>n</i> times
{ <i>n</i> ,}	{ <i>n</i> ,}?	At least <i>n</i> times
{ <i>n</i> , <i>m</i> }	{ <i>n</i> , <i>m</i> }?	From <i>n</i> to <i>m</i> times

ANCHORS

Use	To specify position
^	At start of string or line
\A	At start of string
\Z	At end of string
\$	At end of string or line
\b	On word boundary
\B	Not on word boundary

GROUPS

Use	To define
(<i>exp</i>)	Indexed group
(?P< <i>name</i> > <i>exp</i>)	Named group
(?: <i>exp</i>)	Noncapturing group
(?= <i>exp</i>)	Zero-width positive lookahead
(?! <i>exp</i>)	Zero-width negative lookahead
(?<= <i>exp</i>)	Zero-width positive lookbehind. <i>exp</i> is fixed width
(?<! <i>exp</i>)	Zero-width negative lookbehind. <i>exp</i> is fixed width

FLAGS / INLINE OPTIONS

Option	Effect on match
i	Case-insensitive
m	Multiline mode
L	Locale specific
u	Unicode dependent
s	Single-line mode
x	Ignore white space

Chandra Lingam, Compute With Cloud Inc

<https://github.com/ChandraLingam/PyRegex>

Template: Microsoft/MSDN .NET Regular Expressions

Python Reference: re module documentation

BACKREFERENCES

Use	To match
<code>\n</code>	Indexed group
<code>(?P=name)</code>	Named group

ALTERNATION

Use	To match
<code>a b</code>	Either <i>a</i> or <i>b</i>
<code>(?n)</code> <i>yes no</i>	<i>yes</i> if <i>group n</i> is matched <i>no</i> if <i>group n</i> isn't matched
<code>(?name)</code> <i>yes no</i>	<i>yes</i> if <i>name</i> is matched <i>no</i> if <i>name</i> isn't matched

SUBSTITUTION

Use	To substitute
<code>\g<n></code>	Substring matched by group number <i>n</i>
<code>\g<name></code>	Substring matched by group <i>name</i>

COMMENTS

Use	To
<code>(?# comment)</code>	Add inline comment
<code>#</code>	Add x-mode comment to end

REGULAR EXPRESSION OPERATIONS

Module: re

Pattern matching with Compiled objects

To initialize with	Use constructor
Pattern	<code>re.compile(pattern)</code>
+ flags	<code>re.compile(pattern, flags)</code>

Finding and replacing matched patterns. Use compiled object methods for additional options and fine-tuning parameters

Use method	To
<code>re.match</code>	Find match at start of string
<code>re.search</code>	Find the first match
<code>re.findall</code>	Retrieve all matching strings
<code>re.finditer</code>	Retrieve all matches
<code>re.sub</code>	Replace a matching string
<code>re.split</code>	Split text based on match

Getting info about regular expression patterns

Use compiled object API	To get
<code>groupindex</code>	Dictionary of Group names and group number
<code>groups</code>	Capturing Group Count
<code>pattern</code>	Pattern for compiled object

Processing a match

Use method	To
<code>expand</code>	Replace a match
<code>group</code>	Retrieve value of a group by number or name
<code>groups</code>	Retrieve all subgroups as a tuple
<code>groupdict</code>	Retrieve dictionary of named groups and values
<code>start</code>	Find starting index position of a group
<code>end</code>	Find ending index position of a group

Chandra Lingam, Compute With Cloud Inc

<https://github.com/ChandraLingam/PyRegex>

Template: Microsoft/MSDN .NET Regular Expressions

Python Reference: re module documentation