

Python RegEx

A RegEx, or Regular Expression, is a sequence of characters that forms a search pattern.

RegEx can be used to check if a string contains the specified search pattern.

RegEx Module

Python has a built-in package called re, which can be used to work with Regular Expressions.

Import the re module:

```
import re
```

RegEx in Python

When you have imported the re module, you can start using regular expressions:

Example

Search the string to see if it starts with "The" and ends with "Spain":

```
import re
```

```
txt = "The rain in Spain"
```

```
x = re.search("^The.*Spain$", txt)
```

RegEx Functions

The re module offers a set of functions that allows us to search a string for a match:

Function	Description
findall	Returns a list containing all matches
search	Returns a Match object if there is a match anywhere in the string

split	Returns a list where the string has been split at each match
sub	Replaces one or many matches with a string

ADVERTISEMENT

Metacharacters

Metacharacters are characters with a special meaning:

Character	Description	Example
[]	A set of characters	"[a-m]"
\	Signals a special sequence (can also be used to escape special characters)	"\d"
.	Any character (except newline character)	"he..o"
^	Starts with	"^hello"
\$	Ends with	"planet\$"
*	Zero or more occurrences	"he.*o"
+	One or more occurrences	"he.+o"
?	Zero or one occurrences	"he.?o"
{}	Exactly the specified number of occurrences	"he.{2}o"
	Either or	"falls stays"
()	Capture and group	

Special Sequences

A special sequence is a `\` followed by one of the characters in the list below, and has a special meaning:

Character	Description	Example
<code>\A</code>	Returns a match if the specified characters are at the beginning of the string	<code>"\AThe"</code>
<code>\b</code>	Returns a match where the specified characters are at the beginning or at the end of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	<code>r"\bain"</code> <code>r"ain\b"</code>
<code>\B</code>	Returns a match where the specified characters are present, but NOT at the beginning (or at the end) of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	<code>r"\Bain"</code> <code>r"ain\B"</code>
<code>\d</code>	Returns a match where the string contains digits (numbers from 0-9)	<code>"\d"</code>
<code>\D</code>	Returns a match where the string DOES NOT contain digits	<code>"\D"</code>
<code>\s</code>	Returns a match where the string contains a white space character	<code>"\s"</code>
<code>\S</code>	Returns a match where the string DOES NOT contain a white space character	<code>"\S"</code>
<code>\w</code>	Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore <code>_</code> character)	<code>"\w"</code>
<code>\W</code>	Returns a match where the string DOES NOT contain any word characters	<code>"\W"</code>

<code>\Z</code>	Returns a match if the specified characters are at the end of the string	"Spain\Z"
-----------------	--	-----------

Sets

A set is a set of characters inside a pair of square brackets `[]` with a special meaning:

Set	Description
<code>[arn]</code>	Returns a match where one of the specified characters (a, r, or n) is present
<code>[a-n]</code>	Returns a match for any lower case character, alphabetically between a and n
<code>[^arn]</code>	Returns a match for any character EXCEPT a, r, and n
<code>[0123]</code>	Returns a match where any of the specified digits (0, 1, 2, or 3) are present
<code>[0-9]</code>	Returns a match for any digit between 0 and 9
<code>[0-5][0-9]</code>	Returns a match for any two-digit numbers from 00 and 59
<code>[a-zA-Z]</code>	Returns a match for any character alphabetically between a and z, lower case OR upper case
<code>[+]</code>	In sets, <code>+</code> , <code>*</code> , <code>.</code> , <code> </code> , <code>()</code> , <code>\$</code> , <code>{}</code> has no special meaning, so <code>[+]</code> means: return a match for any <code>+</code> character in the string

The findall() Function

The `findall()` function returns a list containing all matches.

Example

Print a list of all matches:

```
import re
```

```
txt = "The rain in Spain"
```

```
x = re.findall("ai", txt)
print(x)
```

The list contains the matches in the order they are found.

If no matches are found, an empty list is returned:

Example

Return an empty list if no match was found:

```
import re

txt = "The rain in Spain"
x = re.findall("Portugal", txt)
print(x)
```

The search() Function

The search() function searches the string for a match, and returns a [Match object](#) if there is a match.

If there is more than one match, only the first occurrence of the match will be returned:

Example

Search for the first white-space character in the string:

```
import re

txt = "The rain in Spain"
x = re.search("\s", txt)

print("The first white-space character is located in position:", x.start())
```

If no matches are found, the value None is returned:

Example

Make a search that returns no match:

```
import re
```

```
txt = "The rain in Spain"
```

```
x = re.search("Portugal", txt)
```

```
print(x)
```