

# Python RegEx

⟨ Previous

Next >

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 <a href="re">re</a>, which can be used to work with Regular Expressions.

Import the re module:

import re

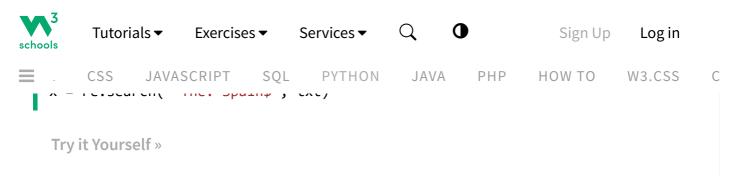
## RegEx in Python

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

#### Example

Get your own Python Server

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



## **RegEx Functions**

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

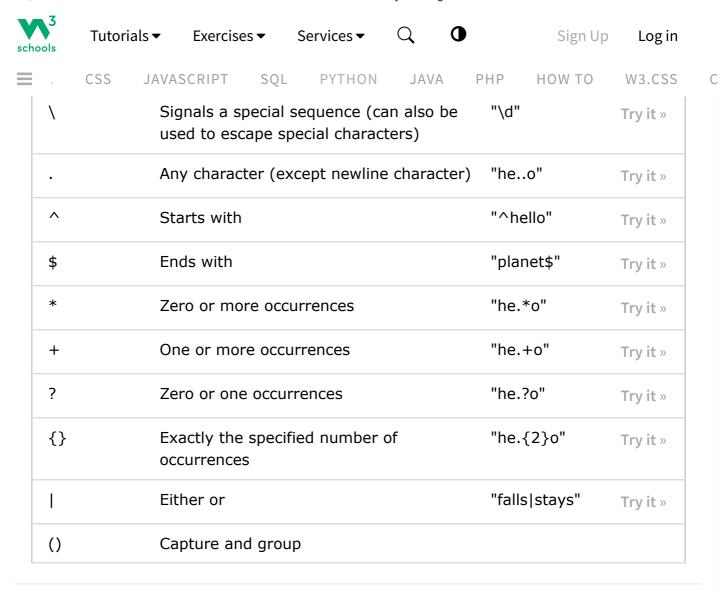
Function	Description
findall	Returns a list containing all matches
<u>search</u>	Returns a Match object if there is a match anywhere in the string
<u>split</u>	Returns a list where the string has been split at each match
<u>sub</u>	Replaces one or many matches with a string

**ADVERTISEMENT** 

Ad removed. Details

### Metacharacters

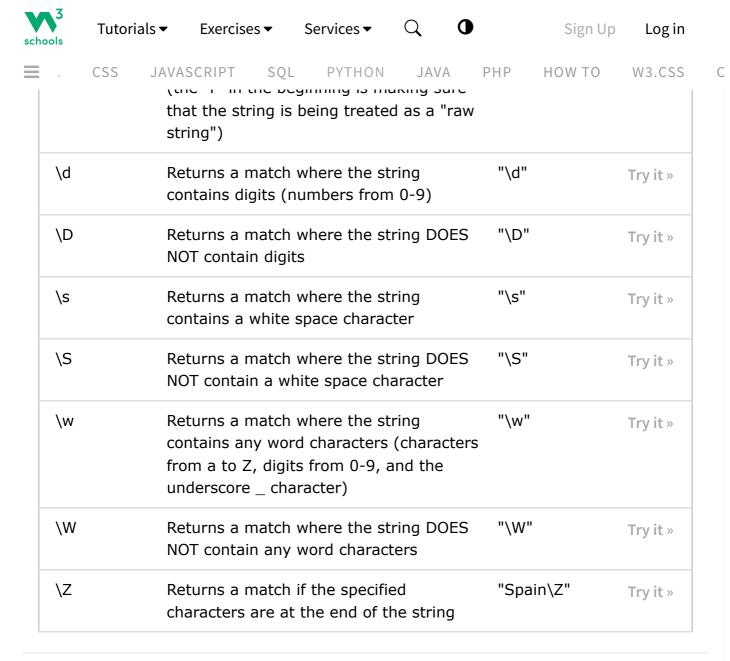
Metacharacters are characters with a special meaning:



## **Special Sequences**

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

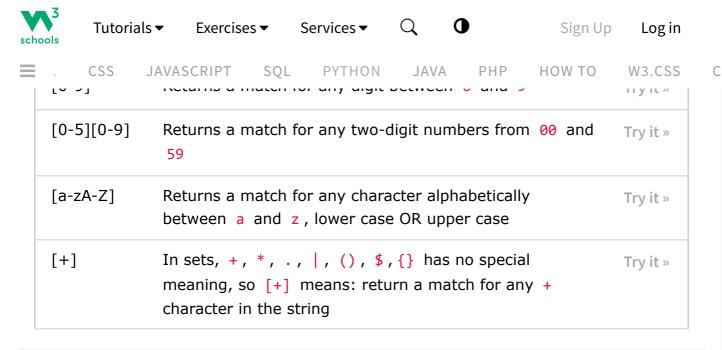
Character	Description	Example	Try it
\A	Returns a match if the specified characters are at the beginning of the string	"\AThe"	Try it »
\b	Returns a match where the specified characters are at the beginning or at the	r"\bain"	Try it »
	end of a word (the "r" in the beginning is making sure that the string is being treated as a "raw string")	r"ain\b"	Try it »



#### Sets

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

Set	Description	Try it
[arn]	Returns a match where one of the specified characters ( $a$ , $r$ , or $n$ ) is present	Try it »
[a-n]	Returns a match for any lower case character, alphabetically between $\frac{1}{2}$ and $\frac{1}{2}$	Try it »
[^arn]	Returns a match for any character EXCEPT $a$ , $r$ , and $n$	Try it »



## 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)
```

Try it Yourself »

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:

### The search() Function

The search() function searches the string for a match, and returns a <u>Match object</u> 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())

Try it Yourself »
```

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"
```



## The split() Function

The split() function returns a list where the string has been split at each match:

### Example

Split at each white-space character:

```
import re

txt = "The rain in Spain"
x = re.split("\s", txt)
print(x)
```

Try it Yourself »

You can control the number of occurrences by specifying the maxsplit parameter:

### Example

Split the string only at the first occurrence:

```
import re

txt = "The rain in Spain"
x = re.split("\s", txt, 1)
print(x)
```

Try it Yourself »



### Example

Replace every white-space character with the number 9:

```
import re

txt = "The rain in Spain"

x = re.sub("\s", "9", txt)
print(x)
```

You can control the number of replacements by specifying the count parameter:

### Example

Try it Yourself »

Replace the first 2 occurrences:

```
import re

txt = "The rain in Spain"

x = re.sub("\s", "9", txt, 2)
print(x)
```

Try it Yourself »

## Match Object

A Match Object is an object containing information about the search and the result.



#### Example

Do a search that will return a Match Object:

```
import re

txt = "The rain in Spain"
x = re.search("ai", txt)
print(x) #this will print an object

Try it Yourself »
```

The Match object has properties and methods used to retrieve information about the search, and the result:

- .span() returns a tuple containing the start-, and end positions of the match.
- .string returns the string passed into the function
- .group() returns the part of the string where there was a match

### Example

Print the position (start- and end-position) of the first match occurrence.

The regular expression looks for any words that starts with an upper case "S":

```
ADVERTISEMENT
import re

txt = "The rain in Spain"
x = re.search(r"\bS\w+", txt)
print(x.span())
```

Try it Yourself »

#### Example



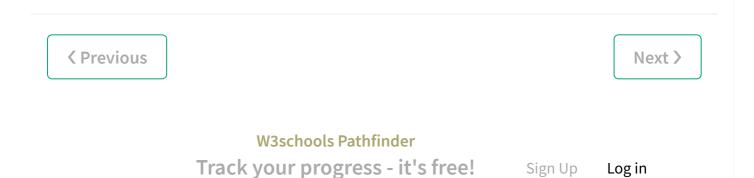


e string where there was a match.

sion looks for any words that starts with an upper case "S":

```
in Spain"
"\bS\w+", txt)
)
```

**Note:** If there is no match, the value None will be returned, instead of the Match Object.







**COLOR PICKER** 





**ADVERTISEMENT** 

**ADVERTISEMENT** 

schools

SPACES UPGRADE AD-FREE



Tutorials ▼ Exercises ▼

Services **▼** 

Q

•

Sign Up

Log in

= .

CSS

JAVASCRIPT

SQL

PYTHON

JAVA

PHP

HOW TO

W3.CSS

C.

#### **Top Tutorials**

HTML Tutorial
CSS Tutorial
JavaScript Tutorial
How To Tutorial
SQL Tutorial
Python Tutorial
W3.CSS Tutorial
Bootstrap Tutorial
PHP Tutorial
Java Tutorial
C++ Tutorial
jQuery Tutorial

#### **Top References**

HTML Reference
CSS Reference
JavaScript Reference
SQL Reference
Python Reference
W3.CSS Reference
Bootstrap Reference
PHP Reference
HTML Colors
Java Reference
Angular Reference
jQuery Reference

#### **Top Examples**

HTML Examples
CSS Examples
JavaScript Examples
How To Examples
SQL Examples
Python Examples
W3.CSS Examples
Bootstrap Examples
PHP Examples
Java Examples
XML Examples
jQuery Examples

#### **Get Certified**

HTML Certificate
CSS Certificate
JavaScript Certificate
Front End Certificate
SQL Certificate
Python Certificate
PHP Certificate
jQuery Certificate
Java Certificate
C++ Certificate
C# Certificate
XML Certificate











#### FORUM ABOUT CLASSROOM

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning.

Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness

of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.

Copyright 1999-2024 by Refsnes Data. All Rights Reserved. W3Schools is Powered by