**Re in Python**

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)

[Try it Yourself »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex" \t "https://www.w3schools.com/python/_blank)

## RegEx Functions

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

|  |  |
| --- | --- |
| **Function** | **Description** |
| [findall](https://www.w3schools.com/python/python_regex.asp" \l "findall) | Returns a list containing all matches |
| [search](https://www.w3schools.com/python/python_regex.asp" \l "search) | Returns a [Match object](https://www.w3schools.com/python/python_regex.asp" \l "matchobject) if there is a match anywhere in the string |
| [split](https://www.w3schools.com/python/python_regex.asp" \l "split) | Returns a list where the string has been split at each match |
| [sub](https://www.w3schools.com/python/python_regex.asp" \l "sub) | Replaces one or many matches with a string |

## Metacharacters

Metacharacters are characters with a special meaning:

|  |  |  |  |
| --- | --- | --- | --- |
| **Character** | **Description** | **Example** | **Try it** |
| [] | A set of characters | "[a-m]" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_meta1" \t "https://www.w3schools.com/python/_blank) |
| \ | Signals a special sequence (can also be used to escape special characters) | "\d" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_meta2" \t "https://www.w3schools.com/python/_blank) |
| . | Any character (except newline character) | "he..o" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_meta3" \t "https://www.w3schools.com/python/_blank) |
| ^ | Starts with | "^hello" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_meta4" \t "https://www.w3schools.com/python/_blank) |
| $ | Ends with | "world$" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_meta5" \t "https://www.w3schools.com/python/_blank) |
| \* | Zero or more occurrences | "aix\*" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_meta6" \t "https://www.w3schools.com/python/_blank) |
| + | One or more occurrences | "aix+" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_meta7" \t "https://www.w3schools.com/python/_blank) |
| {} | Exactly the specified number of occurrences | "al{2}" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_meta8" \t "https://www.w3schools.com/python/_blank) |
| | | Either or | "falls|stays" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_meta9" \t "https://www.w3schools.com/python/_blank) |
| () | 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** | **Try it** |
| \A | Returns a match if the specified characters are at the beginning of the string | "\AThe" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_seq1" \t "https://www.w3schools.com/python/_blank) |
| \b | 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") | r"\bain" r"ain\b" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_seq2" \t "https://www.w3schools.com/python/_blank) [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_seq2-2" \t "https://www.w3schools.com/python/_blank) |
| \B | 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") | r"\Bain" r"ain\B" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_seq3" \t "https://www.w3schools.com/python/_blank) [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_seq3-2" \t "https://www.w3schools.com/python/_blank) |
| \d | Returns a match where the string contains digits (numbers from 0-9) | "\d" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_seq4" \t "https://www.w3schools.com/python/_blank) |
| \D | Returns a match where the string DOES NOT contain digits | "\D" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_seq5" \t "https://www.w3schools.com/python/_blank) |
| \s | Returns a match where the string contains a white space character | "\s" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_seq6" \t "https://www.w3schools.com/python/_blank) |
| \S | Returns a match where the string DOES NOT contain a white space character | "\S" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_seq7" \t "https://www.w3schools.com/python/_blank) |
| \w | Returns a match where the string contains any word characters (characters from a to Z, digits from 0-9, and the underscore \_ character) | "\w" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_seq8" \t "https://www.w3schools.com/python/_blank) |
| \W | Returns a match where the string DOES NOT contain any word characters | "\W" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_seq9" \t "https://www.w3schools.com/python/_blank) |
| \Z | Returns a match if the specified characters are at the end of the string | "Spain\Z" | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_seq10" \t "https://www.w3schools.com/python/_blank) |

## 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) are present | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_set1" \t "https://www.w3schools.com/python/_blank) |
| [a-n] | Returns a match for any lower case character, alphabetically between a and n | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_set2" \t "https://www.w3schools.com/python/_blank) |
| [^arn] | Returns a match for any character EXCEPT a, r, and n | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_set3" \t "https://www.w3schools.com/python/_blank) |
| [0123] | Returns a match where any of the specified digits (0, 1, 2, or 3) are present | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_set4" \t "https://www.w3schools.com/python/_blank) |
| [0-9] | Returns a match for any digit between 0 and 9 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_set5" \t "https://www.w3schools.com/python/_blank) |
| [0-5][0-9] | Returns a match for any two-digit numbers from 00 and 59 | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_set6" \t "https://www.w3schools.com/python/_blank) |
| [a-zA-Z] | Returns a match for any character alphabetically between a and z, lower case OR upper case | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_set7" \t "https://www.w3schools.com/python/_blank) |
| [+] | In sets, +, \*, ., |, (), $,{} has no special meaning, so [+] means: return a match for any + character in the string | [Try it »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_set8" \t "https://www.w3schools.com/python/_blank) |

## 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 »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_findall" \t "https://www.w3schools.com/python/_blank)

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)

[Try it Yourself »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_findall2" \t "https://www.w3schools.com/python/_blank)

## The search() Function

The search() function searches the string for a match, and returns a [Match object](https://www.w3schools.com/python/python_regex.asp" \l "matchobject) 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 »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_search" \t "https://www.w3schools.com/python/_blank)

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)

[Try it Yourself »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_search2" \t "https://www.w3schools.com/python/_blank)

## 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 »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_split" \t "https://www.w3schools.com/python/_blank)

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 »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_split2" \t "https://www.w3schools.com/python/_blank)

## The sub() Function

The sub() function replaces the matches with the text of your choice:

### 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)

[Try it Yourself »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_sub" \t "https://www.w3schools.com/python/_blank)

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

### Example

Replace the first 2 occurrences:

import re  
  
txt = "The rain in Spain"  
x = re.sub("\s", "9", txt, 2)  
print(x)

[Try it Yourself »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_sub2" \t "https://www.w3schools.com/python/_blank)

## Match Object

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

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

### 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 »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_match" \t "https://www.w3schools.com/python/_blank)

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":

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

[Try it Yourself »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_match_span" \t "https://www.w3schools.com/python/_blank)

### Example

Print the string passed into the function:

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

[Try it Yourself »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_match_string" \t "https://www.w3schools.com/python/_blank)

### Example

Print the part of the string where there was a match.

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

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

[Try it Yourself »](https://www.w3schools.com/python/trypython.asp?filename=demo_regex_match_group" \t "https://www.w3schools.com/python/_blank)

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