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:**

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

## 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 |  |  |

Every example of each metacharacters are given next

**Example of using []**

import re

txt = "The rain in Spain"

#Find all lower case characters alphabetically between "a" and "m":

x = re.findall("[a-m]", txt)

print(x)

**Output:**

['h', 'e', 'a', 'i', 'i', 'a', 'i']

**Example of using \d**

import re

txt = "That will be 59 dollars"

#Find all digit characters:

x = re.findall("\d", txt)

print(x)

**output:**

**['5', '9']**

**Example of using .**

import re

txt = "hello planet"

#Search for a sequence that starts with "he", followed by two (any) characters, and an "o":

x = re.findall("he..o", txt)

print(x)

**output:**

['hello']

**Example of using ^**

import re

txt = "hello planet"

#Check if the string starts with 'hello':

x = re.findall("^hello", txt)

print(x)

**output:**

['hello']

**Example of using $**

Import re

txt = "hello planet"

#Check if the string ends with 'planet':

x = re.findall("planet$", txt)

print(x)

**output:**

['planet']

**Example of using** \*

import re

txt = "hello planet"

#Search for a sequence that starts with "he", followed by 0 or more (any) characters, and an "o":

x = re.findall("he.\*o", txt)

print(x)

**output:** ['hello']

**Example of using +**

import re

txt = "hello planet"

#Search for a sequence that starts with "he", followed by 1 or more (any) characters, and an "o":

x = re.findall("he.+o", txt)

print(x)

**output:** ['hello']

**Example of using ?**

import re

txt = "This planet"

#Search for a sequence that starts with "Th", followed by 0 or 1 (any) character, and an "s":

x = re.findall("Th.?s", txt)

print(x)

#If the value of txt is “hello planet”. This time we got no match, because there were not zero, not one, but two characters between "he" and the "o"

**output:** ['This']

**Example of using {}**

import re

txt = "hello planet"

#Search for a sequence that starts with "he", followed excactly 2 (any) characters, and an "o":

x = re.findall("he.{2}o", txt)

print(x)

**output:**

['hello']

**Example of using |**

import re

txt = "The rain in Spain falls mainly in the plain!"

#Check if the string contains either "falls" or "stays" or “rain”:

x = re.findall("falls|stays|rain", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['rain', 'falls']

Yes, there is at least one match!

## 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" |  |
| \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" |  |
| \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" |  |
| \d | Returns a match where the string contains digits (numbers from 0-9) | "\d" |  |
| \D | Returns a match where the string DOES NOT contain digits | "\D" |  |
| \s | Returns a match where the string contains a white space character | "\s" |  |
| \S | Returns a match where the string DOES NOT contain a white space character | "\S" |  |
| \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" |  |
| \W | Returns a match where the string DOES NOT contain any word characters | "\W" |  |
| \Z | Returns a match if the specified characters are at the end of the string | "Spain\Z" |  |

**Every example of each special sequence is given below:**

**Example of using \A**

import re

txt = "The rain in Spain"

#Check if the string starts with "The":

x = re.findall("\AThe", txt)

print(x)

if x:

print("Yes, there is a match!")

else:

print("No match")

**output:**

['The']

Yes, there is a match!

**Example of using \b**

import re

txt = "The rain in Spain"

#Check if "The" is present at the beginning of a WORD:

x = re.findall(r"\bThe", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['The']

Yes, there is at least one match!

**Example of using \B**

import re

txt = "The rain in Spain"

#Check if "ain" is present, but NOT at the beginning of a word:

x = re.findall(r"\Bain", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['ain', 'ain']

Yes, there is at least one match!

**Example of using \B**

import re

txt = "The rain in Spain"

#Check if "ain" is present, but NOT at the end of a word:

x = re.findall(r"ain\B", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

[]

No match

**Example of using \d**

import re

txt = "Bangladesh has 6 divisions"

#Check if the string contains any digits (numbers from 0-9):

x = re.findall("\d", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['6']

Yes, there is at least one match!

**Example of using \D**

import re

txt = "MEHEDI"

#Return a match at every no-digit character:

x = re.findall("\D", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['M', 'E', 'H', 'E', 'D', 'I']

Yes, there is at least one match!

**Example of using \s**

import re

txt = "The rain in Spain"

#Return a match at every white-space character:

x = re.findall("\s", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:** **[' ', ' ', ' ']**

**Yes, there is at least one match!**

**Example of using \S**

import re

txt = "MEHEDI HASAN"

#Return a match at every NON white-space character:

x = re.findall("\S", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['M', 'E', 'H', 'E', 'D', 'I', 'H', 'A', 'S', 'A', 'N']

Yes, there is at least one match!

**Example of using \w**

import re

txt = "001 MEHEDI"

#Return a match at every word character (characters from a to Z, digits from 0-9, and the underscore \_ character):

x = re.findall("\w", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['0', '0', '1', 'M', 'E', 'H', 'E', 'D', 'I']

Yes, there is at least one match!

**Example of using \W**

import re

txt = "The rain in Spain"

#Return a match at every NON word character (characters NOT between a and Z. Like "!", "?" white-space etc.):

x = re.findall("\W", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

[' ', ' ', ' ']

Yes, there is at least one match!

**Example of using \Z**

import re

txt = "The rain in Spain"

#Check if the string ends with "Spain":

x = re.findall("Spain\Z", txt)

print(x)

if x:

print("Yes, there is a match!")

else:

print("No match")

**output:** ['Spain']

Yes, there is a match!

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

**Each example of each set of characters inside the square brackets is given below:**

**Example of using []:**

import re

txt = "The rain in Spain"

#Check if the string has any a, r, or n characters:

x = re.findall("[arn]", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['r', 'a', 'n', 'n', 'a', 'n']

Yes, there is at least one match!

**Example of using []:**

import re

txt = "The rain in Spain"

#Check if the string has any characters between a and n:

x = re.findall("[a-n]", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['h', 'e', 'a', 'i', 'n', 'i', 'n', 'a', 'i', 'n']

Yes, there is at least one match!

**Example of using []:**

import re

txt = "The rain in Spain"

#Check if the string has other characters than a, r, or n:

x = re.findall("[^arn]", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['T', 'h', 'e', ' ', 'i', ' ', 'i', ' ', 'S', 'p', 'i']

Yes, there is at least one match!

**Example of using []:**

import re

txt = "The rain in Spain 001"

#Check if the string has any 0, 1, 2, or 3 digits:

x = re.findall("[0123]", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['0', '0', '1']

Yes, there is at least one match!

**Example of using []:**

import re

txt = "8 times before 11:45 AM"

#Check if the string has any digits:

x = re.findall("[0-9]", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['8', '1', '1', '4', '5']

Yes, there is at least one match!

**Example of using []:**

import re

txt = "8 times before 11:45 AM"

#Check if the string has any two-digit numbers, from 00 to 59:

x = re.findall("[0-5][0-9]", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['11', '45']

Yes, there is at least one match!

**Example of using []:**

import re

txt = "8 times before 11:45 AM"

#Check if the string has any characters from a to z lower case, and A to Z upper case:

x = re.findall("[a-zA-Z]", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['t', 'i', 'm', 'e', 's', 'b', 'e', 'f', 'o', 'r', 'e', 'A', 'M']

Yes, there is at least one match!

**Example of using []:**

import re

txt = "8 times before 11:45 AM"

#Check if the string has any + characters:

x = re.findall("[+t8]", txt)

print(x)

if x:

print("Yes, there is at least one match!")

else:

print("No match")

**output:**

['8', 't']

Yes, there is at least one match!

## 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.

**Output:**

**[**'ai', 'ai']

**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)

**output:**

[]

No match

## 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#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())

**output:**

The first white-space character is located in position: 3

**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)

# when none type object .it will occur an error like attribute error

print(x.start()) # we should use start() method when list has value

**output:** None

## 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

#Split the string at multiple characters like [@ .]:

txt = "mehedia111ub@gmail.com"

x = re.split("[\@\.]", txt)

print(x)

**output:**

['mehedia111ub', 'gmail', 'com']

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

### Example

Split the string only at the first occurrence:

import re

#Split the string at the first white-space character:

txt = "The rain in Spain"

x = re.split("\s", txt, 1)

print(x)

**output:**

['The', 'rain in Spain']

## 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)

**output:**

The9rain9in9Spain

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

### Example

Replace the first 2 occurrences:

import re

#Replace the first two occurrences of a white-space character with the digit 9:

txt = "The rain in Spain"

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

print(x)

**output:**

The9rain9in Spain

## 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

#The search() function returns a Match object:

txt = "The rain in Spain"

x = re.search("ai", txt)

print(x)

**output:**

<\_sre.SRE\_Match object; span=(5, 7), match='ai'>

**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

#Search for an upper case "S" character in the beginning of a word, and print its position:

txt = "The rain in Spain"

x = re.search(r"\bS\w+", txt)

print(x)

print(x.span())

**output:**

<re.Match object; span=(12, 17), match='Spain'>

(12, 17)

### Example

Print the string passed into the function:

import re

#The string property returns the search string:

txt = "The rain in Spain"

x = re.search(r"\bS\w+", txt)

print(x.string)

**output:**

The rain in Spain

### 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

#Search for an upper case "S" character in the beginning of a word, and print the word:

txt = "The rain in Spain"

x = re.search(r"\bS\w+", txt)

print(x.group())

**output:**

Spain

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

NoneType' object has no attribute 'group'.