

# **Python from Scratch**

## **Python RegEx**

### **Lesson 28**

- **RegEx Module**
- **RegEx in Python**
- **RegEx Functions**
- **Metacharacters**
- **Special Sequences**
- **Sets**
- **The findall() Function**
- **The search() Function**
- **The split() Function**
- **The sub() Function**
- **Match Object**

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

## Metacharacters

Metacharacters are characters with a special meaning:

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

## 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 ( <code>a</code> , <code>r</code> , or <code>n</code> ) is present
<code>[a-n]</code>	Returns a match for any lower case character, alphabetically between <code>a</code> and <code>n</code>
<code>[^arn]</code>	Returns a match for any character EXCEPT <code>a</code> , <code>r</code> , and <code>n</code>
<code>[0123]</code>	Returns a match where any of the specified digits ( <code>0</code> , <code>1</code> , <code>2</code> , or <code>3</code> ) are present
<code>[0-9]</code>	Returns a match for any digit between <code>0</code> and <code>9</code>
<code>[0-5][0-9]</code>	Returns a match for any two-digit numbers from <code>00</code> and <code>59</code>
<code>[a-zA-Z]</code>	Returns a match for any character alphabetically between <code>a</code> and <code>z</code> , 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)
```

## 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)
```

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)
```

---

## 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)
```



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)
```

---

## 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
```

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())
```

## 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)
```

## 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())
```

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