

Strings and regular expression

UNIVERSITY of
HOUSTON

DIVISION OF RESEARCH
HEWLETT PACKARD ENTERPRISE DATA SCIENCE INSTITUTE

Matching simple patterns with string

String class has methods for counting and finding position of simple patterns

```
S1 = "I do not see a pattern here, I repeat I do not see the pattern"
```

```
print(S1.count('pattern'))
```

2

```
print(S1.find('pattern'))
```

15

```
print(S1.rfind('pattern'))
```

55

Matching simple patterns with string

String pattern matches are limited to simple patterns

Alternative is to use Regular Expressions

Provided by “re” library

```
import re
```

Regular Expression



A regular expression (or RE) specifies a set of strings that matches it.
Useful for creating search patterns and finding/counting matches

The functions in “re” module lets you:

- check if a particular string matches a given regular expression

Regular Expression, findall method

```
import re
```

```
pattern='G...T.'
```

```
DNA_SAMPLE="ATATATGGTGGTGGAAAAGATCAACAATTAGGAAGATCTTATAGAGAAGTTATGAATACTAA  
ATACAATAATAAGAAGAGCGCATTATTCTGAAAATTTTAAATTTAAAGATAGCAA"
```

```
search_result = re.findall (pattern, DNA_SAMPLE)
```

```
print (search_result)
```

```
['GTGGTG', 'GATCTT', 'GAAGTT', 'GCATTA']
```

Python Regular Expression Quick Guide

<code>^</code>	Matches the beginning of a line
<code>\$</code>	Matches the end of the line
<code>.</code>	Matches any character
<code>\s</code>	Matches whitespace
<code>\S</code>	Matches any non-whitespace character
<code>*</code>	Repeats a character zero or more times
<code>*?</code>	Repeats a character zero or more times (non-greedy)
<code>+</code>	Repeats a character one or more times
<code>+?</code>	Repeats a character one or more times (non-greedy)
<code>[aeiou]</code>	Matches a single character in the listed set
<code>[^XYZ]</code>	Matches a single character not in the listed set
<code>[a-z0-9]</code>	The set of characters can include a range
<code>(</code>	Indicates where string extraction is to start
<code>)</code>	Indicates where string extraction is to end



Regular Expression, findall method

```
import re
```

```
DNA_SAMPLE="ATATATGGTGGTGGAAAAGATCAACAATTAGGAAGATCTTATAGAGAAGTTATGAATACTAA  
ATACAATAATAAGAAGAGCGCATTATTCTGAAAATTTTAAATTTAAAGATAGCAA"
```

```
search_result = re.findall('^A..TA',DNA_SAMPLE)
```

```
print (search_result)
```

```
['ATATA']
```


Free Tool(s) for generating and verifying regex



<https://regex101.com/>

<https://www.regextester.com/>

<https://regexr.com/>

UNIVERSITY of
HOUSTON

DIVISION OF RESEARCH
HEWLETT PACKARD ENTERPRISE DATA SCIENCE INSTITUTE

Regex 101 <https://regex101.com/>

regular expressions 101 @regex101 donate sponsor contact bug reports & feedback wiki whats new?

REGULAR EXPRESSION 7 matches, 29 steps (~0ms)

`/T{2,}/gm`

TEST STRING

ATATATGGTGGTGGAAAAGATCAACAATTAGGAAGATCTTATAGAGAAGTT
ATGAATACTAAATACAATAATAAGAAGAGCGCATTTATTCTGAAAATTTTAA
ATTTAAAGATAGCAA

EXPLANATION

- `/T{2,}/gm`
 - `T{2,}` matches the character `T` literally (case sensitive)

MATCH INFORMATION

Match 1 Export Matches

Full match 27-29 TT

QUICK REFERENCE

Search reference

All Tokens

A singl... `[abc]`
A cha... `^[abc]`

Regex 101



UNIVERSITY of
HOUSTON

DIVISION OF RESEARCH
HEWLETT PACKARD ENTERPRISE DATA SCIENCE INSTITUTE