# Strings and regular expression

# 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

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

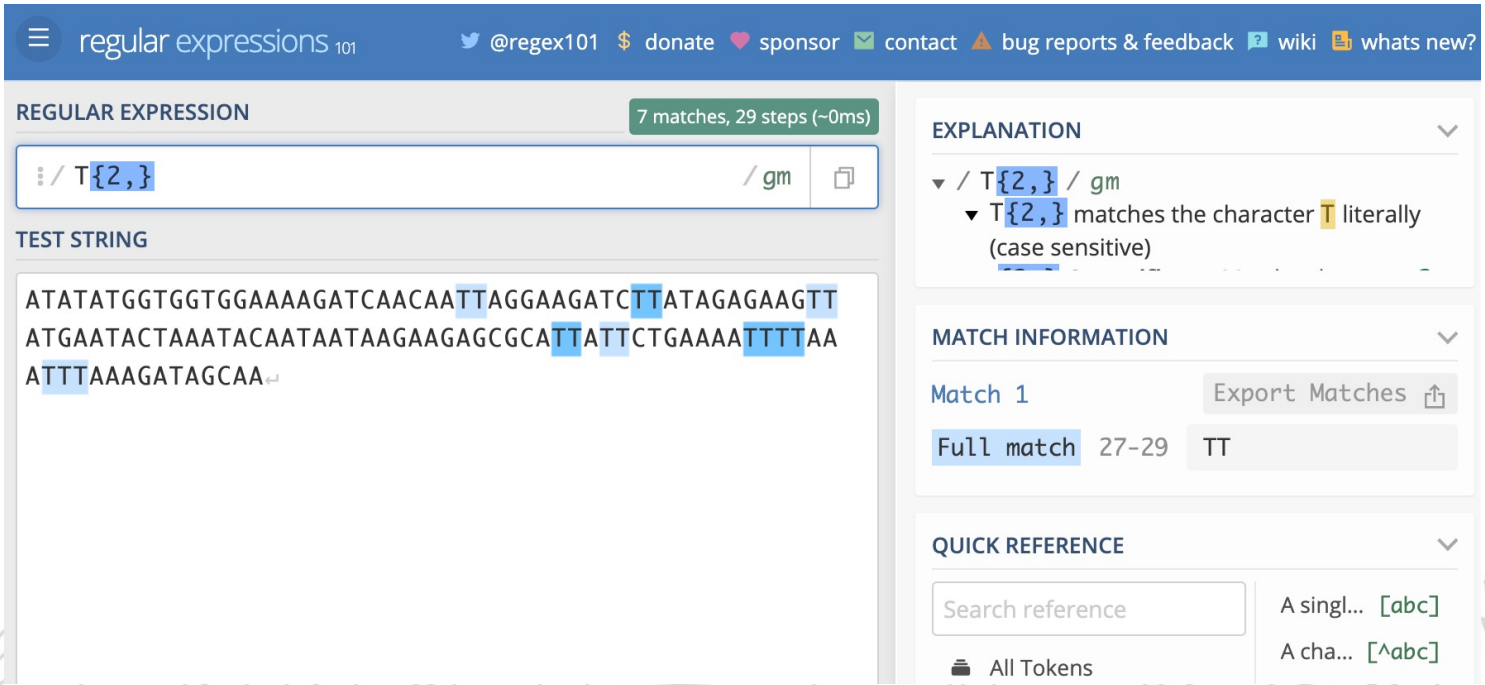
# Free Tool(s) for generating and verifying regex

https://regex101.com/

https://www.regextester.com/

https://regexr.com/

# Regex 101 https://regex101.com/



# Regex 101