

String definitions

String S is a finite sequence of characters

Characters are drawn from alphabet Σ

Usually, $\Sigma = \{ \text{A, C, G, T} \}$

$|S|$ = number of characters in S

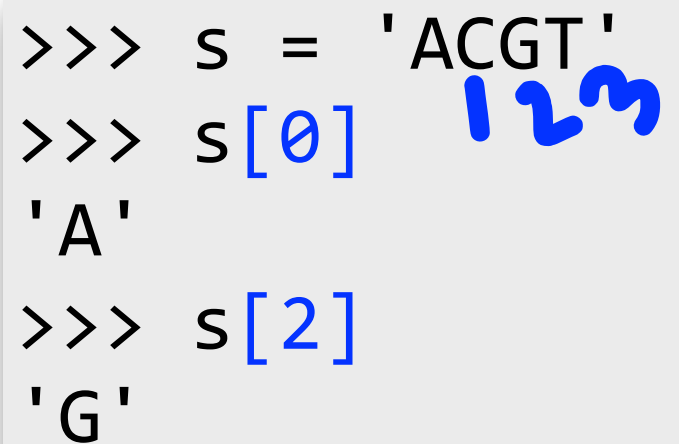
```
>>> s = 'ACGT'  
>>> len(s)  
4
```

ε is “empty string” $|\varepsilon| = 0$

```
>>> len('')  
0
```

String definitions

Positions within a string S are referred to with offsets



A code snippet in a light gray box with a shadow. It contains three lines of Python code. The first line is `>>> s = 'ACGT'`. The second line is `>>> s[0]` followed by the output `'A'`. The third line is `>>> s[2]` followed by the output `'G'`. Blue handwritten annotations include a curved arrow pointing from the word 'offsets' in the text above to the first character 'A' in the string, and the numbers 1, 2, and 3 written below the characters 'A', 'C', and 'G' respectively.

```
>>> s = 'ACGT'
>>> s[0]
'A'
>>> s[2]
'G'
```

Leftmost offset = 0

String definitions

Concatenation of S and T = characters of S followed by characters of T

```
>>> s = 'AACCC'  
>>> t = 'GGTT'  
>>> s + t  
'AACCCGGTT'
```

Substring of S is a string occurring inside S

```
>>> s = 'AACCCGGTT'  
>>> s[2:6]  
'CCGG' # substring of seq
```

String definitions

Prefix of S is a substring starting at the beginning of S

```
>>> s = 'AACCGGTT'  
>>> s[0:6]  
'AACCGG' # prefix  
>>> s[:6] # same as above  
'AACCGG'
```

Suffix is substring ending at end of S

```
>>> s = 'AACCGGTT'  
>>> s[4:8]  
'GGTT' # suffix  
>>> s[4:] # like s[4:len(s)]  
'GGTT'  
>>> s[-4:] # like s[len(s)-4:len(s)]  
'GGTT'
```