

Procesamiento de Texto

Programación

<http://progra.usm.cl>

UTFSM



Departamento de Informática
Universidad Técnica Federico Santa María

Saltos de línea

```
>>> a = 'casa\narbol\npatio'
>>> a
'casa\narbol\npatio'
>>> print a
casa
arbol
patio
>>> len('a\nb\nc')
5
```

Reemplazar secciones de un string

```
>>> palabra = 'cara'
>>> palabra.replace('r', 's')
'casa'
>>> palabra.replace('ca', 'pa')
'para'
>>> palabra.replace('a', 'e', 1)
'cera'
>>> palabra.replace('c', '').replace('a', 'o')
'oro'
```

Partir un string por palabras

```
>>> oracion = 'ex umbra in solem'  
>>> list(oracion)  
['e', 'x', ' ', 'u', 'm', 'b', 'r', 'a', ' ', ' ',  
 'i', 'n', ' ', 's', 'o', 'l', 'e', 'm']  
>>> oracion.split()  
['ex', 'umbra', 'in', 'solem']
```

Partir un string con separador

```
>>> s = 'taca taca'
>>> s.split()
['taca', 'taca']
>>> s.split('a')
['t', 'c', ' t', 'c', '']
>>> s.split('ac')
['t', 'a t', 'a']
```

Unir una secuencia de strings

```
>>> a = ['Ex', 'umbra', 'in', 'solem']
>>> ' '.join(a)
'Ex umbra in solem'
>>> ''.join(a)
'Exumbrainsolem'
>>> ' -> '.join(a)
'Ex -> umbra -> in -> solem'
```

Unir una secuencia de valores (no strings)

```
>>> numeros = range(5)
>>> numeros
[0, 1, 2, 3, 4]
>>> map(str, numeros)
['0', '1', '2', '3', '4']
>>> '-'.join(map(str, numeros))
'0-1-2-3-4'
```

Interpolación de valores por posición

```
>>> s = 'Soy {0} y vivo en {1}'  
>>> s.format('Perico', 'Valparaiso')  
'Soy Perico y vivo en Valparaiso'  
>>> s.format('Erika', 'Berlin')  
'Soy Erika y vivo en Berlin'  
>>> s.format('Wang Dawei', 'Beijing')  
'Soy Wang Dawei y vivo en Beijing'
```


Interpolación de valores por nombre

```
>>> s = '{nombre} estudia en la {u}'
>>> s.format(nombre='Perico', u='UTFSM')
'Perico estudia en la UTFSM'
>>> s.format(nombre='Fulana', u='PUCV')
'Fulana estudia en la PUCV'
>>> s.format(u='UPLA', nombre='Yayita')
'Yayita estudia en la UPLA'
```

Mayúsculas y minúsculas

```
>>> palabra = 'Progra'  
>>> palabra.upper()  
'PROGRA'  
>>> palabra.lower()  
'progra'  
>>> palabra.swapcase()  
'pROGRA'
```

Problema: ADN

Una cadena de ADN es una secuencia de bases nitrogenadas llamadas adenina, citosina, timina y guanina.

En un programa, una cadena se representa como un string de caracteres `'a'`, `'c'`, `'t'` y `'g'`.

A cada cadena, le corresponde una cadena complementaria, que se obtiene intercambiando las adeninas con las timinas, y las citosinas con las guaninas:

```
cadena          = 'cagcccatgaggcagggtg'  
complemento    = 'gtcgggtactccgtcccac'
```

Problema: ADN—Ejercicio 1

Escriba la función `cadena_al_azar(n)` que retorne una cadena aleatoria de ADN de largo `n`:

```
>>> cadena_al_azar(10)
'acgtccgcct'
>>> cadena_al_azar(10)
'tgttcgcatt'
```

Pista:

```
>>> from random import choice
>>> choice('atcg')
't'
>>> choice('atcg')
'g'
```

Problema: ADN—Ejercicio 2

Escriba la función complementaria (c) que entregue la cadena complementaria de c:

```
>>> cadena = 'cagcccatgagggcaggggtg'  
>>> complementaria(cadena)  
'gtcgggtactccgtcccac'
```

Problema: reporte de promedios

Escriba un programa que haga lo siguiente:

```
Numero de alumnos: 3
```

```
Nombre alumno 1: Isaac Newton
```

```
Ingrese las notas de Isaac: 98 94 77
```

```
Nombre alumno 2: Nikola Tesla
```

```
Ingrese las notas de Nikola: 100 68 94 88
```

```
Nombre alumno 3: Albert Einstein
```

```
Ingrese las notas de Albert: 83 85
```

```
El promedio de Isaac es 89.67
```

```
El promedio de Nikola es 87.50
```

```
El promedio de Albert es 84.00
```

Problema: cuenta de palabras

Escriba la función `contar_palabras(oracion)`, que retorne un diccionario indicando cuántas veces aparece cada palabra en la oración:

```
>>> t = 'El sobre, en el aula, esta sobre el pupitre.'  
>>> contar_palabras(t)  
{'el': 3, 'en': 1, 'esta': 1, 'aula': 1, 'sobre': 2,  
'pupitre': 1}
```