



UNIVERSIDAD TECNICA  
FEDERICO SANTA MARIA

DEPARTAMENTO  
DE INFORMÁTICA

# Programación – IWI 131

## Paralelo 18

UVA 5 : Strings

Lunes 03 de Marzo 2023

Profesora: Raquel Pezoa

[raquel.pezoa@usm.cl](mailto:raquel.pezoa@usm.cl)

# UVA 5

## Strings

|     |     |    |    |    |    |    |    |    |    |    |
|-----|-----|----|----|----|----|----|----|----|----|----|
| 0   | 1   | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |
| H   | o   | l  | a  |    | M  | u  | n  | d  | o  | !  |
| -11 | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 |

- Operadores

- +, \*

- Métodos

- len()
  - in, not in
  - upper(), lower()

# UVA 5

## Ciclo for, Ejercicio 1

- Escriba un programa que solicite su nombre y apellido a un usuario e imprima por pantalla las letras en posición impar en mayúscula y las letras en la posición par en minúscula.

Ingrese su nombre y apellido: Juanita Parra

```
0 : j
1 : U
2 : a
3 : N
4 : i
5 : T
6 : a
7 :
8 : p
9 : A
10 : r
11 : R
12 : a
```

# UVA 5

## Ciclo for , Ejercicio 1

- Escriba un programa que solicite su nombre y apellido a un usuario e imprima por pantalla las letras en posición impar en mayúscula y las letras en la posición par en minúscula.

```
nombre = input("Ingrese su nombre y apellido: ")
posicion = 0
for i in nombre: # i toma el valor de cada caracter
    if posicion%2 != 0:
        print(posicion, ":", i.upper())
    else:
        print(posicion, ":", i.lower())
    posicion += 1 # para manejar el índice
```

```
Ingrese su nombre y apellido: Juanita Parra
0 : j
1 : U
2 : a
3 : N
4 : i
5 : T
6 : a
7 :
8 : p
9 : A
10 : r
11 : R
12 : a
```

# UVA 5

## Ejercicio 2

- El siguiente programa pretende separar una línea compuesta por varios campos que están separados por el caracter de punto y coma. Complete las partes faltantes y agregue lo necesario para que funcione según lo descrito.

```
linea = "rut;rol;apellido1;apellido2;nombres"
i = 0
pos_puntuacion = 0
while i < __:
    if __ == ";":
        print(__)
        pos_puntuacion = __
    i+=1
```



Salida

rut  
rol  
apellido1  
apellido2

# UVA 5

## Ejercicio 3

- Considere frase = 'Solo vuela el que se atreve a hacerlo'
  - ¿Cuál es la longitud de este string?
  - ¿Cuáles son los índices válidos?
  - ¿Cómo se expresa siempre el mayor índice en términos de la longitud?
  - ¿Cómo completaría el siguiente while para recorrer todos los índices del string?

# UVA 5

## Ejercicio 4

### A) Casting

- ¿Podemos convertir el string "23515" a un int? ¿cómo?
- ¿Podemos hacer lo mismo con "veintitrés mil quinientos quince"?

### B) Tabla ASCII y orden lexicográfico. ¿Cuál es el resultado de estas comparaciones?

- 'perro' < 'gato'
- 'perro' == 'Perro'    ¿por qué son distintos?
- 'perro' < 'Perro'

# UVA 5

## Ejercicio 5

### Inmutabilidad de strings

- ¿Qué imprime el siguiente programa?

```
nombre = 'Juan Carlos Bodoque'  
nombre.upper()  
print(nombre)
```

- ¿qué hace esto?

```
print(nombre[3])
```

- ¿y esto?

```
nombre[3] = 'A'
```



# UVA 5

## Ejercicio 6

Suponga

```
texto = 'gato grande, negro y gordo'
```

- a) ¿Qué retorna `texto[4]`?
- b) ¿`len(texto[5:8])`?
- c) ¿`texto[:4]`?
- d) ¿`texto[-5:]`?

# UVA 5

## Ejercicio 7

En el básquetbol existen tres diferentes tipos de anotaciones:

- tiro libre (L), que vale 1 punto
- el doble (D), que vale 2 puntos, y
- el triple (T), que vale 3 puntos.

Un partido de básquetbol está dividido en varios períodos.

Escriba un programa que reciba como **entrada una única línea**, que contenga todas las anotaciones realizadas por un equipo de básquetbol durante un partido. Las anotaciones de períodos van separadas por un espacio. Como **salida, debe mostrar la cantidad de puntos obtenidos en cada período y los puntos totales**, como el ejemplo.

```
Anotaciones: DDTDLLDD DDLDT TDTLLD DDDDD
15 puntos en el periodo 1
10 puntos en el periodo 2
12 puntos en el periodo 3
10 puntos en el periodo 4
Total: 47 puntos
```