



## Computación Verano 2022

### Lab. 4.1. Procesamiento de String en Python. Parte II

Enero 24, 2022

**Cree una carpeta (folder) en el disco D, nómbrela con su apellido paterno seguido de su código. Ejemplo: LOPEZ12345**

#### El conjunto de los símbolos (caracteres) del código ASCII.

Todos los símbolos que se usan en computación normalmente están agrupados y ordenados bajo el sistema ASCII. A cada símbolo (carácter) le corresponde un número ordinal, es decir que cada símbolo del sistema (código) ASCII tiene asignado un número que lo representa.

El sistema (código) ASCII tiene 256 símbolos o caracteres.

Si queremos averiguar el código ASCII de la letra "A", usamos la función `ord()` de la siguiente manera:

```
>>> print(ord('A'))  
65
```

Si queremos averiguar el símbolo o carácter que le corresponde al número 64, usamos la función `chr()` de la siguiente manera:

```
>>> print(chr(64))  
@
```

#### Procesamiento de cadenas de caracteres (string)

Un **string** es un conjunto de caracteres ordenados en forma secuencial y que conforman una unidad. Eso quiere decir que cada elemento del **string** tiene asignado un número de orden o dicho de otra manera, tiene asignado su **índice**. La asignación de los índices lo realiza Python en forma automática.

Python asigna el índice empezando en **cero**, luego aumentando en 1 en forma secuencial de izquierda a derecha. Sin embargo, Python también permite asignar índices negativos de derecha a izquierda empezando en -1.

Índice	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Cadena:	H	o	l	a		c	o	m	o		e	s	t	a	s	,	J	o	s	e
Índice:	-20	-19	-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Todo **string** es **inmutable**, es decir que no se puede modificar su contenido. Si se desea modificar, lo que se hace es generar otro **string** con las modificaciones esperadas.

#### Cómo acceder a un elemento de una cadena de caracteres (string)

Para acceder a un elemento de un string basta con colocar el nombre del string y entre corchetes su índice:

```
>>> A="casa" >>> miString = "hola amigo"  
>>> print(A[2]) >>> print(miString[-4])  
's' 'm'
```

## Operaciones básicas

- El operador **+** (concatenación de cadenas): acepta dos cadenas como operadores y devuelve la cadena que resulta de unir la segunda a la primera.

```
>>> 'abc' + 'def'      >>> x="hola"
'abcdef'              >>> y="amigo"
                       >>> x+y
                       "holaamigo"
```

- ¿Qué sucede si queremos concatenar un string con un número? Pruebe lo siguiente:

```
>>> print(4 + "you" + 2 + "more")
```

- La función **str()** permite convertir un número a string.

```
>>> print(str(4) + "you" + str(2) + "more")
```

- Operador **\*** (repetición de cadena): acepta una cadena y un entero y devuelve la concatenación de la cadena consigo misma tantas veces como indica el entero.

```
>>> "N1"*5              >>> x="hola"
'N1N1N1N1N1'          >>> x*4
                       "holaholaholahola"
```

- La función **len()**, devuelve la cantidad de caracteres que contiene la cadena.

```
>>> len("hola! Dr.")
9
>>> s="mi mamá me mima"
>>> len(s)
15
```

**¡¡Cuidado!!**. El índice del último elemento no es 9

## Como imprimir los caracteres de una cadena uno a uno

Podemos imprimir uno a uno los caracteres de una cadena de dos maneras, a)-recorrido por índice, b)-recorrido por valor. En esta parte del curso usaremos principalmente el recorrido por índice.

a):

```
s= "mi mamá me mima"
x= len(s)
for i in range(0,x):
    print(s[i],end="")
```

b):

```
s= "mi mamá me mima"
for i in s:
    print(i,end="")
```

**Para cada uno de los siguientes ejercicios implemente el programa Python correspondiente. Guarde vuestros programas en vuestra carpeta de trabajo.**

- Generar letras minúsculas al azar mostrándolas en pantalla. Terminar cuando se hayan producido 20 ocurrencias de la letra 'w' indicando la cantidad de letras generadas.
- Un grupo de inteligencia militar desea codificar los mensajes secretos de tal forma que no puedan ser interpretados con una lectura directa; para lo cual han establecido las siguientes reglas:
  - Todo mensaje debe tener sus letras en mayúsculas.
  - Reemplazar cada letra por la que sigue según abecedario excepto la 'Z' que deberá reemplazarse con la letra 'A'.
  - Reemplazar cada dígito encontrado por el que le precede según su orden excepto el '0' que deberá reemplazarse por el '9'.

Ingresar un mensaje y mostrar su codificación secreta resultante.

3. Dada una cadena de caracteres, determinar la frecuencia de cada carácter, es decir la cantidad de cada carácter en la cadena.
4. Escriba un programa que lea una frase (las palabras están separadas por un espacio):
  - a. Imprimir en forma tabular cada palabra con su respectiva cantidad de letras.
  - b. Determinar la palabra con mayor longitud

```
Ingrese una frase: No necesito sustitutorio para Computacion
No                2
necesito          8
sustitutorio     12
para              4
computacion      11

La palabra de mayor longitud es: sustitutorio
```

5. Escribir un programa que reciba dos palabras y que las imprima verticalmente una al lado de la otra.

Ejemplo:

```
Ingrese la primera palabra: Hola
Ingrese la segunda palabra: Mundo

H    M
o    u
l    n
a    d
     o
```

## OPCIONAL

6. Escriba un programa que reciba dos cadenas de caracteres del teclado y realice lo siguiente:

Intercambiar los dos primeros caracteres de cada cadena y concatenar las cadenas resultantes en una sola cadena en la que cada cadena esté separada por un espacio.

Ingrese la primera cadena: "abcdefg"

Ingrese la segunda cadena: "vwxyz"

La nueva cadena es: "vwcddefg abxyz"

Guarde todos vuestros programas y vuestra hoja de respuestas en una carpeta con el nombre su **Apellido** paterno seguido de vuestro **DNI**, luego comprima esta carpeta. Envíe este archivo a [victor.melchor.e@upch.pe](mailto:victor.melchor.e@upch.pe) , especificando como asunto **Lab4.1**.