

# Python Básico II

**Alumno 1: Celic Gabriel Hernández Archundia**

**Alumno 2: Diego Iván García Zarza**

**Matrícula 1: 2877240**

**Matrícula 2: 2897709**

## Uso de Strings

```
In [1]: nombre = "Celic Gabriel Hernández Archundia"
```

```
In [2]: len(nombre)
```

```
Out[2]: 33
```

```
In [4]: print(" ".join(["Celic", "Gabriel", "Hernández", "Archundia"]))  
print("-".join(["Celic", "Gabriel", "Hernández", "Archundia"]))  
print("".join(["Celic", "Gabriel", "Hernández", "Archundia"]))
```

```
Celic Gabriel Hernández Archundia  
Celic-Gabriel-Hernández-Archundia  
CelicGabrielHernándezArchundia
```

*Función **split** sirve para separar cadenas de caracteres a partir de un caracter en particular*

```
In [6]: nombre.split(" ")
```

```
Out[6]: ['Celic', 'Gabriel', 'Hernández', 'Archundia']
```

```
In [7]: nombre.split('a')
```

```
Out[7]: ['Celic G', 'briel Hernández Archundi', '']
```

Sirve **replace** sirve para sustituir caracteres por otros

```
In [10]: nombre.replace('Celic', 'el guapo')
```

```
Out[10]: 'el guapo Gabriel Hernández Archundia'
```

```
In [12]: nombre
```

```
Out[12]: 'Celic Gabriel Hernández Archundia'
```

Función **find** sirve para buscar una cadena de caracteres dentro de otra. Encuentra solo la primera coincidencia

```
In [14]: nombre.find("Gabriel")
```

```
Out[14]: 6
```

```
In [17]: nombre.find("Celic")
```

```
Out[17]: 0
```

```
In [19]: nombre.find("a")
```

```
Out[19]: 7
```

Función **rfind** sirve para buscar una cadena de caracteres dentro de otra buscándola desde el final. Pero sigue contando desde 0

```
In [20]: nombre.rfind("H")
```

```
Out[20]: 14
```

Concatenación

```
In [21]: "Celic" + "Gabriel"
```

```
Out[21]: 'CelicGabriel'
```

Uso de listas

```
In [11]: milista = [1,2,3,4,5,6]
milista2 = ["Celic","Gabriel"]
milista3 = [ [1,2,3,4], ["a","b","c", "d"]] # Listas anidadas
```

```
In [12]: milista4 = [True, "asda", 1, 123.2131, []]
```

```
In [13]: len(milista)
```

```
Out[13]: 6
```

```
In [14]: milista2[0]
```

```
Out[14]: 'Celic'
```

```
In [15]: milista2[-1]
```

```
Out[15]: 'Gabriel'
```

```
In [31]: milista + milista2
```

```
Out[31]: [1, 2, 3, 4, 5, 6, 'Celic', 'Gabriel']
```

```
In [43]: milista.append(7)
```

```
In [44]: milista
```

```
Out[44]: [1, 2, 3, 4, 5, 6, 7]
```

```
In [45]: del milista[-1]
```

```
In [46]: milista
```

```
Out[46]: [1, 2, 3, 4, 5, 6]
```

```
In [47]: milista.remove(6)
milista
```

```
Out[47]: [1, 2, 3, 4, 5]
```

La función **index** sirve para encontrar un elemento dentro de una lista

```
In [48]: milista.index(3)
```

```
Out[48]: 2
```

La función **insert** sirve para agregar un elemento a una lista en una posición específica

```
In [49]: milista.insert(1,66) # posición, valor o elemento
```

```
In [20]: milista
```

```
Out[20]: [1, 2, 3, 4, 5, 6]
```

Uso de tuplas

```
In [139]: miTupla = (1, True, "Celic")
print(miTupla)
```

```
(1, True, 'Celic')
```

```
In [52]: len(miTupla)
```

```
Out[52]: 3
```

```
In [53]: miTupla[2]
```

```
Out[53]: 'Celic'
```

```
In [58]: miTupla2 = 1,4,7,3,8,9,12,43,7
```

La función **count** devuelve el número de elementos repetidos de acuerdo a un elemento de búsqueda

```
In [59]: miTupla2.count(7)
```

```
Out[59]: 2
```

```
In [60]: nombre
```

```
Out[60]: 'Celic Gabriel Hernández Archundia'
```

```
In [61]: nombre[7]
```

```
Out[61]: 'a'
```

Ciclo For

```
In [64]: for elemento in nombre:  
         print(elemento)
```

```
C  
e  
l  
i  
c  
  
G  
a  
b  
r  
i  
e  
l  
  
H  
e  
r  
n  
á  
n  
d  
e  
z  
  
A  
r  
c  
h  
u  
n  
d  
i  
a
```

## Funciones en Python

Las funciones se definen con una palabra clave **def** y se tienen la sintaxis: `def nombre_funcion(parametros):` Para devolver un valor utilizando la palabra clave **return**

Ejemplo 1: Escribe una función en Python que encuentre el valor más grande de una lista de números

```
In [17]: def busca_mayor(milista):
          resultado = milista[0]
          for numero in milista:
              if numero > resultado:
                  resultado = numero
          return resultado
num_mayor = busca_mayor(milista)
num_mayor
```

Out[17]: 6

Ejemplo 2: Escribe una función en Python que lea un string y lo devuelva al revés

```
In [41]: [x for x in range(5, 1, -1)]
```

Out[41]: [5, 4, 3, 2]

```
In [50]: def revierteCadena(cadena):
          resultado = ""
          for index in range(len(cadena) - 1, -1, -1):
              resultado = resultado + cadena[index]
          print(resultado)

          revierteCadena("Celic")

          cileC
```

## Ejercicios II

### Ejercicios de Diego Iván García Zarza

**1. Escribe una función que reciba un número positivo entero "n" y muestre un cuadrado de tamaño  $n \times n$  usando el número positivo.**

```
In [108]: def cuadrado():
          n = int(input("Ingresa el valor de un lado:  "))
          for i in range(n):
              print("--" * (n))

          print("El cuadrado se ve asi")
          cuadrado()
```

Ingresa el valor de un lado: 5

```
-----
-----
-----
-----
-----
```

El cuadrado se ve asi

**2. Escribe una función en Python que encuentre el palíndromo más cercano al valor de un número. Si hay dos palíndromos cuya distancia absoluta es la misma, muestra el valor más pequeño.**

```
In [111]: def encontrar_palindromo(numero):

    palindromo_inf = numero
    palindromo_sup = numero

    while (str(palindromo_inf)[::-1]) != str(palindromo_inf):
        palindromo_inf -= 1

    while (str(palindromo_sup)[::-1]) != str(palindromo_sup):
        palindromo_sup += 1

    if (abs(numero - palindromo_sup) < abs(numero - palindromo_inf)):
        palindromo = palindromo_sup

    else:
        palindromo = palindromo_inf

    return palindromo

numero = int(input("Ingresa un numero para conocer el palindromo mas cercano:
resultado = encontrar_palindromo(numero)

print(resultado)
```

Ingresa un numero para conocer el palindromo mas cercano: 10  
9

**3. Escribe una función en Python que encuentre los dígitos que no aparecen en un dado número de teléfono.**

```
In [112]: def encontrar_digitos_faltantes(numero):
    digitos_faltantes = set("0123456789") - set(str(numero))
    return digitos_faltantes

numero_telefono = input("Ingresa el numero de telefono: ")
digitos_faltantes = encontrar_digitos_faltantes(numero_telefono)

print("Los dígitos faltantes del numero ", numero_telefono, " son:", digitos_f
```

Ingresa el numero de telefono: 5540046660  
Los dígitos faltantes del numero 5540046660 son: {'9', '7', '1', '8', '2', '3'}

```
In [113]: def digitos_faltantes(numero):
            if len(numero) < 10:
                print("El numero que ingreso le faltan: ", 10-len(numero), " digitos")
            elif len(numero) > 10:
                print("El numero que ingreso es invalido ya que tiene mas de 10 numeros")
            else:
                print("El numero que ingreso esta correcto, felcidades")

            numero = input("Ingrese un numero telefonico: ")
            digitos_faltantes(numero)
```

Ingrese un numero telefonico: 456  
El numero que ingreso le faltan: 7 digitos

#### 4. Escribe una función en Python que calcule la suma de los dígitos de un número.

```
In [115]: from numpy import double
            def suma_numeros():
                numero =int(input("Ingrese algun nunero que desee sumar entre todos sus dig

                suma = 0

                while numero > 0:
                    digito = numero % 10
                    suma += digito
                    numero //= 10
                print ("La suma es igual a:", suma)
            suma_numeros()
```

Ingrese algun nunero que desee sumar entre todos sus digitos: 3456  
La suma es igual a: 18

#### 5. Escribe una función en Python que reciba una oración y la convierta a una lista de palabras.

```
In [116]: def separador_palabras():
            oracion = input("Ingresa una oracion o frase: ")
            palabras = oracion.split()

            for palabra in palabras:
                print (palabra)
            separador_palabras()
```

Ingresa una oracion o frase: Hola soy Homelo  
Hola  
soy  
Homelo

#### 6. Escribe una función en Python que encuentre el dígito más grande y el más pequeño de un número.



```
In [118]: def encontrar_max_min_digito():
    numero = int(input("Ingrese el numero que quiere resolver: "))
    digito_min = 9
    digito_max = 0

    while numero != 0:
        digito = numero % 10
        if digito > digito_max:
            digito_max = digito
        if digito < digito_min:
            digito_min = digito

        numero //= 10

    print("\nEl digito mas grande es :", digito_max, "\nEl digito mas pequeño es :", digito_min)

encontrar_max_min_digito()
```

Ingrese el numero que quiere resolver: 10

El digito mas grande es : 1

El digito mas pequeño es: 0

**7. Escribe una función en Python que indique si en una cadena de caracteres aparecen dos símbolos iguales consecutivos.**

```
In [122]: def cadena_simbolos(cadena):

    simbolo_anterior = ""

    for simbolo in range(len(cadena)-1):
        if cadena[simbolo] == cadena[simbolo+1]:
            if cadena[simbolo] != simbolo_anterior:
                print(f"\nLos símbolos consecutivos son: {cadena[simbolo]}{cadena[simbolo+1]}")
                simbolo_anterior = cadena[simbolo]

    cadena = input("Ingrese una cadena de caracteres: ")

    cadena_simbolos(cadena)
```

Ingrese una cadena de caracteres: oola

Los símbolos consecutivos son: oo

**8. Escribe una función en Python que reciba una lista de palabras e indique la palabra más larga y su tamaño.**

```
In [123]: def lista_palabras (cadena):

    palabras = cadena.split()
    palabra_mas_larga = ""
    tamaño_max = 0

    for palabra in palabras:
        tamaño_palabra = len(palabra)
        if tamaño_palabra > tamaño_max:
            palabra_mas_larga = palabra
            tamaño_max = tamaño_palabra

    print(f"La palabra más larga es '{palabra_mas_larga}' con un tamaño de {tamaño_max}")

cadena = input("Ingrese varias palabras: ")
lista_palabras(cadena)
```

Ingrese varias palabras: El deber de un soldado es proteger a la humanidad  
La palabra más larga es 'humanidad' con un tamaño de 9

**9. Escribe una función en Python que reciba una cadena de caracteres y elimine aquellos cuyo índice sea un número impar.**

```
In [125]: def eliminar_numero_impar(cadena):
    nueva_cadena = ""
    for numero, caracter in enumerate(cadena):
        if numero % 2 == 0:
            nueva_cadena += caracter
    return nueva_cadena

cadena= input("Ingresa una cadena de caracteres: ")
print(eliminar_numero_impar(cadena))
```

Ingresa una cadena de caracteres: Halo es el mejor juego  
Hl se eo ug

**10. Escribe una función en Python que ordene una cadena de caracteres de manera lexicográfica.**

```
In [127]: def orden_alfabetico():
    cadena = input("Ingresa un texto u oracion a ordenar: ")
    cadena_ordenada = " ".join(sorted(cadena))
    print(cadena_ordenada)
orden_alfabetico()
```

Ingresa un texto u oracion a ordenar: Titanfall es mejor que Apex  
A T a a e e e e f i j l l m n o p q r s t u x

## Ejercicios de Celic Gabriel Hernández Archundia

**11. Escribe una función en Python que recibe un conjunto de valores RGB y los convierta a código Hexadecimal.**

```

In [130]: # Diccionario de conversión de decimal a hexadecimal
tablaConversion = {0: '0', 1: '1', 2: '2', 3: '3', 4: '4',
                   5: '5', 6: '6', 7: '7',
                   8: '8', 9: '9', 10: 'A', 11: 'B', 12: 'C',
                   13: 'D', 14: 'E', 15: 'F'}

# Función para convertir de decimal a Hexadecimal
def convertirAHexadecimal(decimal):
    hexadecimal = ''
    while (decimal > 0):
        residuo = decimal % 16
        hexadecimal = tablaConversion[residuo] + hexadecimal
        decimal = decimal // 16

    return hexadecimal

# Función para pasar de valores RGB a código hexadecimal
def RGB_CHexadecimal():
    print("Bienvenido. Por favor, introduce los valores RGB\n")
    R = input("R:")
    R = int(R)
    G = input("G:")
    G = int(G)
    B = input("B:")
    B = int(B)

    if (R > 255): R = 255
    if (G > 255): G = 255
    if (B > 255): B = 255
    if (R < 0): R = 0
    if (G < 0): G = 0
    if (B < 0): B = 0

    print("\nResultados Forma 1 (Función)")
    print("\nCódigo Hexadecimal:", convertirAHexadecimal(R)
                                                , convertirAHexadecimal(G)
                                                , convertirAHexadecimal(B))

    print("\nResultado Forma 2 (Método hex)")
    print("\nCódigo Hexadecimal:", hex(R)
                                                , hex(G)
                                                , hex(B))

RGB_CHexadecimal()

```

Bienvenido. Por favor, introduce los valores RGB

R:255

G:100

B:34

Resultados Forma 1 (Función)

Código Hexadecimal: FF 64 22

Resultado Forma 2 (Método hex)

Código Hexadecimal: 0xff 0x64 0x22

## 12. Escribe una función en Python que convierta los valores de una tupla a string

```
In [150]: def convertirString(nuevaTupla):
            y = list(nuevaTupla)
            i = 0

            for elementos in nuevaTupla:
                y[i] = str(elementos)
                i+=1
            x = tuple(y)

            return x

myTupla = (1, 45.3, True, "Holaaa")

print(convertirString(myTupla))

('1', '45.3', 'True', 'Holaaa')
```

## 13. Escribe una función en Python que sume todos los valores numéricos de una lista

```
In [151]: def sumaValoresLista(listaValores):
            x = 1
            y = 1.1
            suma = 0
            for elemento in listaValores:
                if type(elemento) == type(x) or type(elemento) == type(y):
                    suma += elemento

            return suma

Lista = [1,2,3,4,5.5, "Celic", True, 10]
print(sumaValoresLista(Lista))

25.5
```

## 14. Escribe una función en Python que remueva valores duplicados de una lista

```
In [152]: def removerDuplicado(thisList):
            for elemento in thisList:
                if thisList.count(elemento) > 1:
                    thisList.remove(elemento)

            return thisList

ListaD = [1, 1, 23, "Celic", "Celic", "Carlos", 45, 45.5]
print(removerDuplicado(ListaD))
```

[1, 23, 'Celic', 'Carlos', 45, 45.5]

**15. Escribe una función en Python que revise si una lista está vacía o no.**

```
In [1]: def estaVacíaQ(estaLista):
        j = len(estaLista)

        if j == 0:

            return "Esta lista está vacía"
        else:

            return estaLista

ListaV1 = [1, True, "Celic"]
ListaV2 = []

print ("Lista 1:", estaVacíaQ(ListaV1))
print ("Lista 2:", estaVacíaQ(ListaV2))
```

Lista 1: [1, True, 'Celic']  
Lista 2: Esta lista está vacía

**16. Escribe una función en Python que genere todas las permutaciones de una lista.**

```
In [1]: def permutaciones(empezar, fin=[]):
        if len(empezar) == 0:
            print(fin)
        else:
            for i in range(len(empezar)):
                permutaciones(empezar[:i] + empezar[i+1:], fin + empezar[i:i+1])

listaPe = [1,2,3]
permutaciones(listaPe)
```

[1, 2, 3]  
[1, 3, 2]  
[2, 1, 3]  
[2, 3, 1]  
[3, 1, 2]  
[3, 2, 1]

**17. Escribe una función en Python que encuentre el segundo valor más grande de una lista de números.**

```
In [35]: def segundoMayor(listaM):  
  
    return (sorted(listaM, reverse=True)[1])  
  
listaMayor = [4,7,8,2,9,5]  
print("El segundo número más alto es:",segundoMayor(listaMayor))
```

El segundo número más alto es: 8

**18. Escribe una función en Python que imprima todos los valores únicos de una lista**

```
In [105]: from collections import Counter  
  
def valoresUnicos(listaAñadida):  
    x = 0  
    listaR = [k for k,v in Counter(listaAñadida).items() if v>1]  
    print(listaR)  
    nuevaLista = listaAñadida  
    for j in listaR:  
        for i in listaAñadida:  
            if j == i:  
                nuevaLista.remove(j)  
  
            if j == i+1:  
                nuevaLista.remove(j)  
  
            if j+1 == i:  
                nuevaLista.remove(j)  
  
    return nuevaLista  
  
listaRepetida = [3,3,5,6,9,8,8,1,2,1]  
print("Valores únicos de la lista:",valoresUnicos(listaRepetida))
```

[3, 8, 1]  
Valores únicos de la lista: [5, 6, 9, 2]

**19. Escribe una función en Python que convierta una cadena a una lista.**

```
In [43]: def convertirALista(cadena):  
         return list(cadena)  
  
texto = "Hola, soy Celic. Tengo 20 años"  
print("Nueva lista:",convertirALista(texto))
```

```
Nueva lista: ['H', 'o', 'l', 'a', ',', ' ', 's', 'o', 'y', ' ', 'C', 'e',  
'l', 'i', 'c', '.', ' ', 'T', 'e', 'n', 'g', 'o', ' ', '2', '0', ' ', 'a',  
'ñ', 'o', 's']
```

**20. Escribe una función en Python que revise si un string es palíndromo o no.**

```
In [46]: def esPalindromo(palabra):  
         if str(palabra) == str(palabra[::-1]):  
             respuesta = "es un palíndromo"  
         else:  
             respuesta = "no es un palíndromo"  
  
         return respuesta  
  
palabra = input()  
print("Resultado:",palabra,esPalindromo(palabra))
```

```
madam  
Resultado: madam es un palíndromo
```