



EJERCICIOS DE PYTHON III

Funciones

- Definición y llamada
- Funciones sin parámetros
- Parámetros y argumentos
- Parámetros opcionales
- Parámetros posicionales
- Devolución de un valor
- Devolución de varios valores
- Funciones que usan funciones

Codo a Codo
2024

- 1) **Ejercicio 1 (introducción):** La universidad desea un programa que permita mostrar a través de una función un mensaje que permita darle la bienvenida a los alumnos, similar al siguiente:

```
=====
Bienvenidos a la Universidad de Python!
=====
```

Además, se requiere mostrar los números de aulas disponibles de los 5 pisos con los que cuenta uno de los edificios. Las aulas se numeran desde el número 100 y hay 5 por piso. Mostrar de la siguiente manera:

```
Piso    Aulas
1        100 a 110
2        200 a 210
3        300 a 310
4        400 a 410
5        500 a 510
```

- 2) **Ejercicio 2 (parámetros):** Mostrar a través de una función un mensaje de bienvenida a los alumnos, pero indicando el cuatrimestre y el año. Validar en el programa principal que se haya ingresado un valor en ambas variables, considerando que el año ingresado debe ser mayor o igual a 2000. El mensaje será similar al siguiente:

```
Bienvenidos estudiantes!
1er cuatrimestre 2024
```

Además, desarrollar una función que permita calcular el valor de un curso dependiendo de la forma de pago. El importe y la forma de pago (contado, débito o crédito) se ingresan por teclado.

- Si la forma de pago es contado se hace un descuento del 10%
- Si la forma de pago es débito no se hace descuento
- Si la forma de pago es crédito el interés por pagar en 3 cuotas es del 15% y se van agregando 15% para 6, 9 o 12 cuotas.
- Si la forma de pago no es correcta se deberá informar a través de un mensaje.

Calcular el valor de la cuota y el total financiado y mostrar lo siguiente, según el caso:

```
Ingrese un importe: 24000
Ingrese una forma de pago: Contado, Débito, Crédito: Contado
Forma de pago: CONTADO
Valor: 21600.00
```

```
Ingrese un importe: 24000
Ingrese una forma de pago: Contado, Débito, Crédito: Débito
Forma de pago: DÉBITO
Valor: 24000.00
```

```
Ingrese un importe: 24000
Ingrese una forma de pago: Contado, Débito, Crédito: Crédito
Forma de pago: CRÉDITO
Cuotas  Valor cuota    Total financiado
3        9200.00      27600.00
```

6	5200.00	31200.00
9	3866.67	34800.03
12	3200.00	38400.00

```
Ingrese un importe: 24000
Ingrese una forma de pago: Contado, Débito, Crédito: Otro
Forma de pago: OTRO
Forma de pago errónea
```

- 3) **Ejercicio 3 (parámetros opcionales y parámetros posicionales):** La universidad requiere una función que permita registrar datos de los alumnos. Esta función recibe su nombre completo, la sede y el año de ingreso. Dado que la mayoría de los alumnos son de la sede “Buenos Aires” y el año de ingreso es el actual¹ considerar estos valores por defecto, además mostrar el nombre con la primera letra de cada palabra en mayúsculas. Llamar a las funciones y mostrar los datos de la siguiente manera:

```
registrar_datos("agustina gonzález", "Córdoba", 2021)
registrar_datos("diego lópez", "Misiones")
registrar_datos("ANA FERNÁNDEZ")
```

```
Se ha inscripto a Agustina González en la sede Córdoba para el año
2021
Se ha inscripto a Diego López en la sede Misiones para el año 2024
Se ha inscripto a Ana Fernández en la sede Buenos Aires para el año
2024
```

Además, crear una función que reciba y muestre los siguientes datos de la sede: dirección, ciudad y provincia, considerando como valor por defecto “Buenos Aires”. Realizar llamadas a la función indicando una posición distinta en los parámetros, por ejemplo:

```
datos_sede("Av. Las Heras 3456", "Godoy Cruz", "Mendoza")
datos_sede(provincia='Chubut', ciudad='Rawson', direccion='Belgrano
312')
datos_sede(ciudad='Mar del plata', direccion='Av. Moreno 56')
```

La salida por pantalla esperada es similar a esta:

```
Universidad de Python - Av. Las Heras 3456 - Godoy Cruz - Mendoza
Universidad de Python - Belgrano 312 - Rawson - Chubut
Universidad de Python - Av. Moreno 56 - Mar del plata - Buenos
Aires
```

- 4) **Ejercicio 4 (devolución de valores):** La universidad requiere una función que reciba el valor de la cuota y un porcentaje de aumento (número entero). La función debe devolver la cuota con el aumento aplicado. Solicitar en el programa principal la cuota y el porcentaje y mostrar en el programa principal estos dos valores más la cuota aumentada, de la siguiente manera:

¹ Puedes traer el año actual de la siguiente manera:

```
from datetime import datetime
anio_actual = datetime.now().year
```

```
Cuota    $ 25000
Aum      15%
Total    $ 28750.0
```

Además, se requiere una función que valide si un alumno es mayor de edad. Recibe la edad y devuelve True o False.

En el programa principal ingresar la edad y llamar a la función hasta que la edad ingresada sea mayor a 18, luego mostrar la edad y el valor retornado por la función (debe ser *True*). Ejemplo:

```
Ingrese la edad del estudiante: 12
Debe ser mayor o igual a 18. Ingrese la edad del estudiante: 16
Debe ser mayor o igual a 18. Ingrese la edad del estudiante: 17
Debe ser mayor o igual a 18. Ingrese la edad del estudiante: 19
Edad: 19. Es mayor? True
```

Desarrollar una función que reciba el valor de un curso y retorne tres valores: pagado en 3 cuotas sin interés, en 6 cuotas con un 25% de interés o en 9 cuotas con un 50% de interés.

En el programa principal se debe mostrar el valor del curso y las tres opciones de financiación (recordar desempaquetar), como el ejemplo:

```
Ingrese el costo del curso: 10000
Costo del curso $ 10000
- Tres cuotas de $ 3333.33
- Seis cuotas de $ 2083.33
- Nueve cuotas de $ 1666.67
```

Finalmente la universidad desea una función que permita ingresar las materias que se dictan en un cuatrimestre. Recibe la cantidad de materias que se van a ingresar y devuelve una lista con las materias ingresadas.

En el programa principal mostrar cada uno de los elementos de la lista, como el ejemplo:

```
Listado de materias:
1- Sistemas Operativos
2- Programación I
3- Algoritmos
4- Robótica
```

5) **Ejercicio 5 (funciones que usan funciones):** La universidad requiere un sistema que realice varias tareas:

- Debe crear una lista y permitir la carga de N notas en ella hasta que se ingrese -1. No recibe parámetros y devuelve la lista.
- Debe validar que las notas ingresadas sean números entre 1 y 10 (considerar la excepción del -1).
- Obtendrá la cantidad de notas ingresadas y el promedio. Mostrará esos datos en pantalla, como el ejemplo:

```
Ingrese una nota: 8
Ingrese una nota: -2
Dato no válido! Ingrese una nota: -9
Dato no válido! Ingrese una nota: 11
Dato no válido! Ingrese una nota: 15
```

```
Dato no válido! Ingrese una nota: 6
Ingrese una nota: 5
Ingrese una nota: 10
Ingrese una nota: -1
Cantidad de notas: 4. Promedio: 7.25
```