

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

Módulo 1. Introducción a la programación en Python y MATLAB

1.1. Introducción

1.1.1. Salida por pantalla

Ejercicio 01_01. Salida por pantalla en Python.

```
print("Hola Mundo")
```

Hola Mundo

Ejercicio 01_02. Salida por pantalla en MATLAB.

```
disp("Ejemplo 001- Salida por pantalla en MATLAB");  
disp("Hola Mundo");
```

Ejemplo 001- Salida por pantalla en MATLAB
Hola Mundo

Ejercicio 02_01. Salida por pantalla en Python con salto de línea \n.

```
print("salida", "de", "datos", "en", "Python", sep="\n") # si no se indica sep="\n"  
se usa espacio en blanco como separador.
```

salida
de
datos
en
Python

Ejercicio 02_02. Salida por pantalla en MATLAB con salto de línea \n.

```
s1 = sprintf("Salida\nde\ndatos\nen\nMATLAB");  
disp(s1);
```

Salida
de
datos
en
MATLAB

Ejercicio 03_01. Mostrar la suma de dos números en un mensaje por pantalla.

```
print("El resultado de sumar 2+2 es", 2+2)
```

El resultado de sumar 2+2 es 4

Ejercicio 03_02. Mostrar la suma de dos números en un mensaje por pantalla.

```
sprintf("El resultado de sumar 2+2 es %d",2+2);
```

"El resultado de sumar 2+2 es 4"

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

Ejercicio 04_01. Mostrar la multiplicación de dos números en un mensaje por pantalla.

```
print("El resultado de multiplicar 2x3 es", 2*3)
```

El resultado de multiplicar 2x3 es 6

Ejercicio 04_02. Mostrar la multiplicación de dos números en un mensaje por pantalla.

```
sprintf('El resultado de multiplicar 2x3 es %d.',2*3);
```

El resultado de multiplicar 2x3 es 6.

Ejercicio 05_01. Construir el mensaje de salida por pantalla usando valores de diferentes tipos de variables.

```
a = 1.5  
print("El valor de la variable a es %f" %(a))
```

El valor de la variable a es 1.500000

Ejercicio 05_02. Construir el mensaje de salida por pantalla usando valores de diferentes tipos de variables.

```
a = 1.5 % valor float  
b = 890  
sprintf("El valor de la variable a es %f", a) % muestra a con todos los decimales por defecto  
sprintf("El valor de la variable a es %5.2f", a) % 5 es la parte entera y 2 los decimales  
sprintf("El valor de la variable b es %4.2f",b) % aunque b es entero, lo mostrará con 2 decimales
```

```
a = 1.5000  
b = 890  
ans = "El valor de la variable a es 1.500000"  
ans = "El valor de la variable a es 1.50"  
ans = "El valor de la variable b es 890.00"
```

Ejercicio 06_01. Construir el mensaje de salida por pantalla usando más de una variable.

```
a = 1.5  
b = 2.5  
print("El valor de la variable a es %d y el de b es %f" %(a,b))
```

El valor de la variable a es 1 y el de b es 2.500000

Ejercicio 06_02. Construir el mensaje de salida por pantalla usando más de una variable.

```
c = 1.5  
d = 890  
sprintf("El valor de la variable c es %d y de la variable d es %f", c, d)  
sprintf("El valor de la variable c es %4.2d y de la variable d es %5.2f", c, d)
```

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

```
c = 1.5000
d = 890
ans = "El valor de la variable c es 1.500000e+00 y de la variable d es 890.000000"
ans = "El valor de la variable c es 1.50e+00 y de la variable d es 890.00"
```

1.1.2. Tipos de datos y asignación de variables

Ejercicio 07_01. Construir el mensaje de salida por pantalla usando una cadena o *string* en Python.

```
cadena = "hola mundo"
print(cadena) # Imprime la variable cadena
print(type(cadena)) # Nos muestra el tipo de dato

hola mundo
<class 'str'>
```

Ejercicio 07_02. Construir el mensaje de salida por pantalla usando una cadena o *string* en MATLAB.

```
cadena = 'hola mundo'
sprintf('%s', 'hola mundo')
cadena = "hola mundo"
sprintf('%s', cadena)

cadena = "hola mundo"
ans = 'hola mundo'
```

Ejercicio 08_01. Dadas unas variables indicar su tipo por un mensaje en pantalla.

```
a = 10
print("El valor del número es: %d" %(a))
print(type(a))
b = 11.5
print("El valor del número con decimales es: %f" %(b))
print(type(b))

El valor del número es: 10
<class 'int'>
El valor del número con decimales es: 11.500000
<class 'float'>
```

Ejercicio 08_02. Dadas unas variables indicar su tipo por un mensaje en pantalla.

```
a = 10
b = 11.5
c = 'hello'

sprintf("El valor del número a es: %d", a)
tipo_de_variable = class(a); % Obtener el tipo de variable de a
disp(['El tipo de variable de a es: ', tipo_de_variable]);
sprintf("El valor del número b es: %f", b)
tipo_de_variable = class(b); % Obtener el tipo de variable de b
```

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

```
disp(['El tipo de variable de b es: ', tipo_de_variable]);
sprintf("El valor de c es: %s",c)
tipo_de_variable = class(c); % Obtener el tipo de variable de c
disp(['El tipo de variable de c es: ', tipo_de_variable]);

a = 10
b = 11.5000
c = 'hello'
ans = "El valor del número a es: 10"
El tipo de variable de a es: double
ans = "El valor del número b es: 11.500000"
El tipo de variable de b es: double
ans = "El valor de c es: hello"
El tipo de variable de c es: char
```

Ejercicio 09_01. Reasignación de variables.

```
nuevo_numero = b
print("El valor del nuevo número es: %d" %(nuevo_numero))

El valor del nuevo número es: 11.5000
```

Ejercicio 09_02. Reasignación de variables.

```
b = 11.5
nuevo_numero = b
sprintf("El valor del número es: %5.2d", nuevo_numero)
sprintf("El valor del número es: %s", nuevo_numero)

b = 11.5000
nuevo_numero = 11.5000
ans = "El valor del número es: 1.15e+01"
ans = "El valor del número es: 1.150000e+01"
```

Ejercicio 10_01. Asignación múltiple de variables en Python.

```
a, b, c = 1, 2, 3
print(a, b, c)

1, 2, 3
```

Ejercicio 10_02. Asignación múltiple de variables en MATLAB.

```
[a, b, c] = deal(1, 2, 3);
fprintf('%d, %d, %d\n', a, b, c);

1, 2, 3
```

Ejercicio 11_01. Asignación de no valor.

```
a = None
print(a)

None
```

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

Ejercicio 11_02. Asignación de no valor.

```
x = NaN; % Asigna NaN a la variable x
disp(x);
```

NaN

```
% Además se puede asignar no valor a vectores:
A = [1, 2, NaN, 4]; % Crea una matriz con NaN como elemento
y = []; % Asigna un valor vacío a la variable y
B = [1, 2, [], 4]; % Crea una matriz con un valor vacío en el medio
```

```
A = 1     2     NaN     4
```

Ejercicio 12_01. Tipos de variables *booleanas*.

```
a = True # Deben empezar con mayúsculas
print(a)
b = False # Deben empezar con mayúsculas
print(b)
```

True
False

Ejercicio 12_02. Tipos de variables *booleanas*.

```
a = true % En MATLAB la palabra reservada true es en minúsculas
b = false % En MATLAB la palabra reservada false es en minúsculas
str_a = num2str(a); % Convierte el valor booleano en una cadena
str_b = num2str(b);
fprintf("El valor booleano de a es %s\n", str_a); % Muestra 1
fprintf("El valor booleano de b es %s\n", str_b);

if a
    str_a = 'True'; % Si a es verdadero, asigna 'true' a str_a
else
    str_a = 'False'; % Si a es falso, asigna 'false' a str_a
end
if b
    str_b = 'True'; % Si b es verdadero, asigna 'true' a str_b
else
    str_b = 'False'; % Si b es falso, asigna 'false' a str_b
end
fprintf("El valor booleano de a es %s\n", str_a); % Muestra la cadena 'True'
fprintf("El valor booleano de b es %s\n", str_b); % Muestra la cadena 'False'

a = logical
    1
b = logical
    0
El valor booleano de a es 1
El valor booleano de b es 0
El valor booleano de a es True
El valor booleano de b es False
```

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

Ejercicio 13_01. Todas las variables son objetos. Podemos comprobar el tipo de objeto de una variable.

```
a = 5
isinstance(a,int) # Devuelve True
```

True

Ejercicio 13_02. Todas las variables son objetos. Podemos comprobar el tipo de objeto de una variable.

```
variable = 42; % Define una variable de ejemplo
tipo_de_objeto = class(variable); % Comprueba el tipo de objeto de la variable
fprintf('La variable es de tipo: %s\n', tipo_de_objeto); % Muestra el resultado
```

La variable es de tipo: double

1.1.3. Entrada de datos.

Ejercicio 14_01. Leer por teclado datos usando el comando `input()`. Este comando devuelve una cadena de caracteres.

```
# Lee un primer número entero
numero1 = input("Por favor inserte el primer número entero: ")
# Lee un segundo número con decimales
numero2 = input("Por favor inserte el segundo número con decimales: ")
# En este punto tanto numero1 como numero2 son string
# numero1 será entero, así que usamos int()
numero1 = int(numero1)
# numero2 será un real, así que usamos float()
numero2 = float(numero2)
# mostramos el resultado de la suma
print("La suma de %d + %f es: %f" %(numero1,numero2, numero1 + numero2))
```

Por favor inserte el primer número entero: 1
Por favor inserte el segundo número con decimales: 2.4
La suma de 1 + 2.400000 es: 3.400000

```
# Podemos hacer la suma directamente
# Leemos un tercer número entero
numero3 = int(input("Por favor inserte el tercer número entero: "))
# Leemos un cuarto número con decimales
numero4 = float(input("Por favor inserte el cuarto número con decimales: "))
# Mostramos el resultado de la suma
print("La suma de %d + %f es: %f" %(numero3,numero4, numero3 + numero4))
```

Por favor inserte el tercer número entero: 1
Por favor inserte el cuarto número con decimales: 1.44
La suma de 1 + 1.400000 es: 2.440000

Ejercicio 14_02. Leer por teclado datos usando el comando `input()`. Este comando devuelve una cadena de caracteres.

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

```
num1 = input("Introduzca número entero: ")
num2 = input("Introduzca número decimal: ")
sprintf('El resultado de sumar %d+%f es: %f', num1+num2);
resultado = num1 + num2; % Calcular la suma
% sprintf para formatear la cadena
cadena_formateada = sprintf('El resultado de sumar %d + %f es: %f', num1, num2,
resultado);
disp(cadena_formateada);
```

```
Introduzca número entero: 4
num1 = 4
Introduzca número decimal: 2.3
num2 = 2.3000
El resultado de sumar 4 + 2.300000 es: 6.300000
```

1.1.4. Cadenas.

Ejercicio 15_01. Definir una cadena.

```
cad = "Hola"
cad2 = 'Hola2'
print(cad,cad2)
```

```
Hola
Hola2
```

Ejercicio 15_02. Definir una cadena.

```
cad = 'Hola';
cad2 = 'Hola2';
fprintf('%s %s\n', cad, cad2);
```

```
Hola Hola2
```

Ejercicio 16_01. Insertar valores de variables en una cadena.

```
variable = "Luis"
print("Hola %s" %variable)
# Otra forma
cad = f"Hola {variable}"
print(cad)
# Otra forma
print("Hola {}".format(variable))
```

```
Hola Luis
Hola Luis
Hola Luis
```

Ejercicio 16_02. Insertar valores de variables en una cadena.

```
variable = 'Luis';
fprintf('Hola %s\n', variable);
```

```
Hola Luis
```

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

Ejercicio 17_01. Concatenar cadenas con +

```
cad = "Esto" + " " + "es" + " " + "una" + " " + "prueba"  
print(cad)
```

```
nombre = "María"  
apellidos = "García"  
nombre_completo = nombre + " " + apellidos  
print("El nombre completo es %s." %nombre_completo)
```

```
Esto es una prueba  
El nombre completo es María García
```

Ejercicio 17_02. Insertar valores de variables en una cadena.

```
cad = 'Esto es una prueba';  
disp(cad);  
nombre = 'María';  
apellidos = 'García';  
nombre_completo = [nombre ' ' apellidos];  
fprintf('El nombre completo es %s.\n', nombre_completo);
```

```
Esto es una prueba  
El nombre completo es María García.
```

Ejercicio 18_01. En Python las cadenas son *arrays* de caracteres con el índice comenzando en 0.

```
# Mostramos Los elementos de La cadena  
palabra = "Hola"  
print(palabra[0])  
print(palabra[1])  
print(palabra[2])  
print(palabra[3])
```

```
H  
o  
l  
a
```

Ejercicio 18_02. En MATLAB Las cadenas son *arrays* de caracteres con el índice comenzando en 1.

```
palabra = 'Hola';  
fprintf('%c\n', palabra(1));  
fprintf('%c\n', palabra(2));  
fprintf('%c\n', palabra(3));  
fprintf('%c\n', palabra(4));
```

```
H  
o  
l  
a
```

Autores: María Inmaculada Rodríguez García , María Gema Carrasco García, Javier González Enrique, Juan Jesús Ruiz Aguilar, Ignacio J. Turias Domínguez. [Universidad de Cádiz](#)

Ejercicio 19_01. Podemos convertir otros tipos de datos a cadena.

```
a = 24
cad = "El número es " + str(a)
print(cad)
```

El número es 24

Ejercicio 19_02. Podemos convertir otros tipos de datos a cadena.

```
a = 23;
cad = sprintf('El número es %d', a);
disp(cad);
```

El número es 23