


7 Entrada y salida estándar.

Objetivo. ...

Funciones de Python: ...

[MACTI-Algebra_Lineal_01](#) by [Luis M. de la Cruz](#) is licensed under [Attribution-ShareAlike 4.0 International](#) 

8 Entrada estándar: `input()`

La entrada estándar para proporcionar información a un programa se realiza mediante la función `input()`.

Veamos algunos ejemplos:

```
# Cuando se ejecuta, se espera a que el usuario teclee algo y luego de <enter>
input()
```

34

'34'

```
# Se puede asignar el valor que se teclea a una variable
entrada = input()
```

34

```
# Ahora imprimimos el valor guardado
print(entrada)
```

34

```
# Se puede poner un mensaje para que el usuario sepa
# lo que se espera:
entrada = input('Teclea un valor entero :')
```

Teclea un valor entero : 5

```
# Lo que se lee siempre se transforma en una cadena de texto:
print(type(entrada))
print(entrada)
```

<class 'str'>

5

```
# Podemos hacer el 'casting' para transformar lo que teclea el usuario
# en el tipo de dato requerido:
entrada = int(input('Teclea un valor entero :'))
```

Teclea un valor entero : 5

```
print(type(entrada))
print(entrada)
```

```
<class 'int'>
5
```

Lo anterior se debe realizar con más cuidado e incluso usando declaraciones para capturar posibles errores del usuario al teclear un valor.

9 Salida estándar: `print()`

Existen varias formas de presentar la salida de un programa al usuario, la más común es en la pantalla (la salida estándar).

En todos los casos se desea un control adecuado sobre el formato de la salida. Para ello se tienen varias maneras de controlar esta salida:

9.1 Cadenas con formato `f` o `F`.

Este tipo de cadenas se forman anteponiendo una `f` o `F` al principio de la misma. De esta forma, es posible poner variables entre llaves `{}` dentro de la definición de la cadena.

```
nombre = 'LUIS MIGUEL'
edad = 25
f'Hola mi nombre es {nombre} y tengo {edad} años'
```

'Hola mi nombre es LUIS MIGUEL y tengo 25 años'

En el caso de números es posible agregar un formato:

```
import math
print(f'El valor de PI es aproximadamente {math.pi:.10f}.') # {valor:formato}
```

El valor de PI es aproximadamente 3.1415926536.

También es posible alinear el texto de la salida.

```
na1 = 'Fulano'; n1 = 5_521_345_678
na2 = 'Sutano'; n2 = 7_712_932_143
```

```
print(f'{na1:10} ==> {n1:15d}') # alineación del texto
print(f'{na2:10} ==> {n2:15d}') # alineación del texto
```

```
Fulano      ==>      5521345678
Sutano      ==>      7712932143
```

Para más información véase [Format Specification Mini-Language](#)

9.2 Método `format()`

Las cadenas tienen un método llamado `format()` que permite darle formato a la misma.

Veamos unos ejemplos:

```
print('El curso se llama "{}" y tenemos {} alumnos'.format('Python de cero a exper
```

El curso se llama "Python de cero a experto" y tenemos 100 alumnos

```
votos_a_favor = 42_572_654 # Este es un formato de número entero
votos_en_contra = 43_132_495 # que usa _ para separar los miles
total_de_votos = votos_a_favor + votos_en_contra
porcentaje = votos_a_favor / total_de_votos

# El primer dato se alinea al centro usando ^
# El segundo dato tendrá dos valores antes del punto y dos valores después.
print('{:^20} votos a favor ({:2.2%})'.format(votos_a_favor, porcentaje))
```

```
42572654          votos a favor (49.67%)
```

```
# Se pueden usar números para identificar los argumentos de format()
print('{0} y {1}'.format('el huevo', 'la gallina'))
print('{1} y {0:^20}'.format('el huevo', 'la gallina'))
```

```
el huevo y la gallina
la gallina y          el huevo
```

```
# Se le puede dar nombre a los argumentos para que
# sea más fácil entender la salida
print('Esta {sustantivo} es {adjetivo}'.format(adjetivo='exquisita', sustantivo=
```

Esta comida es exquisita.

```
# Se pueden combinar números con nombres de argumentos
print('El {0}, el {1}, y el {otro}'.format('Bueno', 'Malo', otro='Feo'))
```

El Bueno, el Malo, y el Feo.

```
gatos = {'Siamés': 5, 'Siberiano': 4, 'Sphynx': 0}

# Podemos usar las características de los diccionarios
# para imprimir la salida.
print('Sphynx: {g[Sphynx]:d}; Siamés: {g[Siamés]:d}; Siberiano: {g[Siberiano]:d}')
```

Sphynx: 0; Siamés: 5; Siberiano: 4

```
# Una manera más entendible:
print('Sphynx: {Sphynx:d}; Siamés: {Siamés:d}; Siberiano: {Siberiano:d}'.format(*
```

Sphynx: 0; Siamés: 5; Siberiano: 4

```
for x in range(1, 11):
    print('{0:2d} {1:3d} {2:4d}'.format(x, x*x, x*x*x))
```

```
1   1   1
2   4   8
3   9  27
4  16  64
5  25 125
6  36 216
7  49 343
8  64 512
9  81 729
10 100 1000
```

Para más información véase [Format String Syntax](#)

9.3 Forma antigua de formatear la salida

Se puede seguir usando la forma antigua de la salida.

```
import math
print('El valor aproximado de pi es %5.6f.' % math.pi)
```

El valor aproximado de pi es 3.141593.