

★ Python 3 para impacientes ★



"Simple es mejor que complejo" (Tim Peters)

Python	IPython	EasyGUI	Tkinter	JupyterLab	Numpy
--------	---------	---------	---------	------------	-------

sábado, 1 de febrero de 2014

Salida Estándar: print()

La función **print()** se utiliza para mostrar información en la salida estándar que, normalmente, se corresponde con la pantalla de un ordenador.

```
print('Python') # muestra: Python
luna = 385000 # asigna valor a variable
print('D.M. Tierra-Luna', luna, 'km') # D.M. Tierra-Luna 385000 km
print('D.M. Tierra-Luna ' + str(luna)+' km')
# D.M. Tierra-Luna 385000 km

print('\nHola\npythonis\n') # muestra cadena en varias líneas
print('Continuará...', end=' ') # ejecuta varios print() en misma...
print('otro día') # ...línea. Muestra: Continuará... otro día
print() # inserta una línea en blanco

lu = 'Quito (Ecuador)' # asigna cadena a variable
la = -0.216979 # asigna número flotante a variable
lo = -78.51627 # asigna número flotante a variable

# concatena con formato de intérprete
co = repr(lu)+' : '+repr(la) + ',' + repr(lo)
print(co) # 'Quito (Ecuador)': -0.216979,-78.51627

alt = 110 # asigna entero
dis = 550 # asigna entero
edi = 'La Giralda' # asigna cadena
est = 'Antares' # asigna cadena
print('{} mide {} metros'.format(edi, alt))
# La Giralda mide 110 metros

print('{1} metros: {}'.format(edi, alt)) # 110 metros: La Giralda
print('{c}:{p}'.format(c='Lima', p='Perú')) # Lima:Perú

val1 = 8.56767 # asigna flotante
val2 = 9.45548 # asigna flotante

# muestra redondeo con 2 y 3 decimales
print('{0:.3} {1:.4}'.format(val1, val2))

# rellena con guiones bajos
print('{0:_^30}'.format('Sevilla'))

codpais = 34 # asigna número

# rellena con ceros a la izquierda:
print(str(codpais).zfill(4)) # 0034

valor = 2.34565676 # asigna flotante

# muestra: Valor aproximado
print('Valor aprox. {:.3f}'.format(valor)) # 2.346

# inserta salto de línea antes de imprimir
print('\nCódigos Postales')

# declara diccionario
correos = {'SJ' : 300, 'LR': 309, 'B': 310}
for loc, cp in correos.items(): # recorre diccionario
    # muestra lista de pares con formato
    print('{0:5}:{1:4d}'.format(loc, cp))

print('\nTabla de Multiplicar') # muestra tablas de multiplicar
for x in range(1, 11): # recorre los números del 1 al 10
    print(repr(x)+' :')
```

Buscar

Python para impacientes

[Python](#)
[IPython](#)
[EasyGUI](#)
[Tkinter](#)
[JupyterLab](#)
[Numpy](#)

Anexos

[Guía urgente de MySQL](#)
[Guía rápida de SQLite3](#)

Entradas + populares

[Dar color a las salidas en la consola](#)

En Python para dar color a las salidas en la consola (o en la terminal de texto) existen varias posibilidades. Hay un método basado ...

[Instalación de Python, paso a paso](#)

Instalación de Python 3.6 A finales de 2016 se produjo el lanzamiento de Python 3.6 . El propósito de esta entrada es mostrar, pas...

[Añadir, consultar, modificar y suprimir elementos en Numpy](#)

Acceder a los elementos de un array. [], [,]. ... Acceder a un elemento de un array. Para acceder a un elemento se utiliz...

[Variables de control en Tkinter](#)

Variables de control Las variables de control son objetos especiales que se asocian a los widgets para almacenar sus valore...

[Cálculo con arrays Numpy](#)

Numpy ofrece todo lo necesario para obtener un buen rendimiento cuando se trata de hacer cálculos con arrays. Por como está concebido...

[Tkinter: interfaces gráficas en Python](#)

Introducción Con Python hay muchas posibilidades para programar una interfaz gráfica de usuario (GUI) pero Tkinter es fácil d...

[Operaciones con fechas y horas. Calendarios](#)

Los módulos datetime y calendar amplían las posibilidades del módulo time que provee funciones para manipular expresiones de ti...

[Convertir, copiar, ordenar, unir y dividir arrays Numpy](#)

Esta entrada trata sobre algunos métodos que se utilizan en Numpy para convertir listas en arrays y viceversa; para copiar arrays d...

[Tkinter: Tipos de ventanas](#)

Ventanas de aplicación y de diálogos En la entrada anterior tratamos los distintos gestores de geometría que se utilizan para di...

[El módulo random](#)

El módulo random de la librería estándar de Python incluye un conjunto de funciones

```

for y in range(1, 11): # recorre los números del 1 al 10
    print(repr(x).ljust(2), '*', end='') # muestra operadores y ...
    print(repr(y).rjust(3), end='') # ... resultado en una línea
    print(' = ' + repr(x*y).center(5))

# Utilizando comodines:
# %s (cadena), %i (entero), %f (número con decimales)
#
# Los datos también se pueden tabular reservando un número
# determinado de caracteres:
# Ejemplo: %10s reserva 10 posiciones y alinea a la izquierda
#          %-10s reserva 10 posiciones y alinea a la derecha

nombre = 'Claudio'
edad = 35
altura = 1.82

print('Tiene %i años' % edad) # Tiene 35 años
print('%s tiene %i años y mide %f' % (nombre, edad, altura))
# Claudio tiene 35 años y mide 1.820000

# Tabulando datos:

personas = [('Claudio', 35, 1.826),
            ('Elena', 24, 1.84),
            ('Manuel', 9, 1.449),
            ('Isabel', 34, 1.72)]

for persona in personas:
    nombre = persona[0]
    edad = persona[1]
    altura = persona[2]
    print('%-8s tiene %2i años y mide %1.2f' % (nombre, edad, altura))

# Salida:
# Claudio tiene 35 años y mide 1.83
# Elena tiene 24 años y mide 1.84
# Manuel tiene 9 años y mide 1.45
# Isabel tiene 34 años y mide 1.72

```

Relacionado:

- [Las cadenas f](#)
- [Dar color a las salidas en la consola](#)

[Ir al índice del tutorial de Python](#)

Publicado por Pherkad en [2:53](#)



Etiquetas: [Python3](#)

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)

que permiten obtener de distintos modos números a...

Archivo

febrero 2014 (17) ▾

python.org**pypi.org****Sitios**

- [ActivePython](#)
- [Anaconda](#)
- [Bpython](#)
- [Django](#)
- [Flask](#)
- [Ipython](#)
- [IronPython](#)
- [Matplotlib](#)
- [MicroPython](#)
- [Numpy](#)
- [Pandas](#)
- [Pillow](#)
- [PortablePython](#)
- [PyBrain](#)
- [PyCharm](#)
- [PyDev](#)
- [PyGame](#)
- [Pypi](#)
- [PyPy](#)
- [Pyramid](#)
- [Python.org](#)
- [PyTorch](#)
- [SciPy.org](#)
- [Spyder](#)
- [Tensorflow](#)
- [TurboGears](#)