

# ★ Python 3 para impacientes ★



"Simple es mejor que complejo" (Tim Peters)

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

sábado, 1 de febrero de 2014

## Variables locales y variables globales

Una variable **local** se declara en su ámbito de uso (en el programa principal y dentro de una función) y una **global** fuera de su ámbito para que se pueda utilizar en cualquier función que la declare como global.

```
# Define función
def acelerar():
    # Declara la variable 'km' como global
    # Ahora se podrá utilizar dentro de la función
    global km

    # Declara variable Local (ámbito función)
    tiempo = 1

    # Se incrementa La velocidad en 5 km
    km+= 5

# Define variable Local (ámbito programa principal)
km = 10

# Muestra variable 'km'
print('Velocidad:', km) # velocidad: 10

# Llama a la función acelerar()
acelerar()

# Muestra variable 'km'
print('Velocidad:', km) # velocidad: 15

# Intenta mostrar la variable 'tiempo'
# Se produce una excepción (error) de tipo NameError
# porque la variable no pertenece a este ámbito:
# NameError: name 'tiempo' is not defined
print('Tiempo:', tiempo)
```

### Relacionado:

[Diccionarios de variables locales y globales](#)

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

Publicado por Pherkad en 7:22



Etiquetas: [Python3](#)

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)

### 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

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](#)