

★ Python 3 para impacientes ★

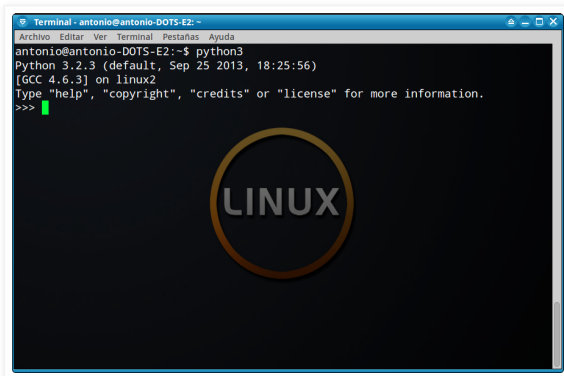


"Simple es mejor que complejo" (Tim Peters)

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

miércoles, 29 de enero de 2014

Sesión interactiva Python



En una sesión interactiva con la consola predeterminada de Python3 podemos ejecutar una sentencia o un bloque de sentencias. Este modo de trabajo resulta de mucha utilidad para ejecutar programas pequeños, realizar cálculos y leer la documentación del lenguaje Python3.

(Existe una consola llamada [IPython](#) que incorpora algunas mejoras que pueden ser consideradas más adelante).

Iniciar una sesión interactiva

\$ python3

Una sesión interactiva comienza con la siguiente información:

```
Python 3.2.3 (default, Sep 25 2013, 18:22:43)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Los signos >>> indican el lugar donde introducir las sentencias.

Finalizar una sesión interactiva

En Linux:

Ctrl+D o exit()

En Windows:

Ctrl+Z o exit()

Ejecutar sentencias

Ejecutar sentencias, una a una:

En una sesión interactiva después de escribir una sentencia la ejecutamos pulsando la tecla [Enter]:

```
print("Hola Pythonisos")
```

Con la tecla **[Flecha Arriba]** podemos regresar a sentencias ejecutadas con anterioridad, modificarlas y/o volverlas a ejecutar.

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

Ejecutar bloques o secuencias de sentencias:

En ocasiones por exigencia de la sintaxis Python es necesario indentar o sangrar las líneas de código con la tecla **[Tabulador]**, o bien, precediendo la sentencia con cuatro espacios en blanco. Para indicar el fin de un grupo de líneas sangradas pulsaremos **[Enter]** y para ejecutar el bloque pulsaremos nuevamente la tecla **[Enter]**.

```
tengo_euros=10 [Enter]
if tengo_euros>7: [Enter]
[Tab ó cuatro espacios] print("me voy al cine") [Enter]
[Enter]
```

El sangrado del código es una característica de Python que hace posible una lectura cómoda de los programas.

Realizar cálculos

En el modo interactivo de Python podemos realizar cálculos matemáticos escribiendo expresiones del tipo:

```
2 * 5
→ 10
```

```
(4 + 3 * 10) / 2
→ 17.0
```

```
10 // 3
→ 3 (al dividir devuelve la parte entera del cociente)
```

```
5 ** 3
→ 125
```

El signo "=" se utiliza para asignar valores a variables. En adelante, las variables podrán incluirse en cualquier expresión que se escriba. Si una variable no está definida e intentamos emplearla el sistema producirá una excepción o error.

Para acceder al valor de una variable, simplemente, hay que escribir su nombre:

```
precio_tapa = 3
numero_tapas = 5
a_pagar = precio_tapa * numero_tapas
a_pagar
```

Resultado: 15

Para asignar un mismo valor a varias variables:

```
altura = ancho = 10
```

El resultado final de cualquier operación se asigna siempre a "_":

Escribir dos operaciones seguidas.

```
altura * anchura
_ * 5
```

Es equivalente a:

```
10 * 10
100 * 5
```

Resultado: 500

Sesiones de Ayuda

Una sesión de ayuda da acceso a la documentación de Python.

Iniciar sesión de ayuda:

```
>>> help()
```

Una sesión de ayuda comienza con el siguiente texto:

```
help>
```

Finalizar sesión de ayuda:

```
help> quit
```

que permiten obtener de distintos modos números a...

Archivo

enero 2014 (10) ▼

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](#)

En una sesión de ayuda se puede consultar la documentación disponible de los módulos, funciones y sentencias de Python. Ejemplos:

Mostrar lista de módulos disponibles:

```
help> modules
```

Mostrar palabras reservadas del lenguaje Python:

```
help> keywords
```

Mostrar documentación del módulo os:

```
help> os
```

Muestra documentación de la sentencia if:

```
help> if
```

Cuando se muestra la documentación de un módulo, una función o una sentencia, podemos avanzar línea a línea con la tecla [Enter] o con [Flecha Abajo] y retroceder con [Flecha Arriba]. Para avanzar o retroceder, pantalla a pantalla, lo haremos con [Flecha Derecha] y [Flecha Izquierda], respectivamente. Cuando se alcance el final de la ayuda aparecerá el indicador (END). Para terminar la lectura, en cualquier momento, pulsar la tecla [q].

Ayuda directa:

En vez de iniciar una sesión de ayuda es posible pedir ayuda directa sobre una sentencia o un módulo concreto de Python con:

```
help("sentencia|función|módulo")
```

```
>>> help("print")
```

```
>>> help("os")
```

En el modo interactivo podemos obtener la lista de nombres de los atributos, de los métodos de una clase, un módulo o de cualquier objeto con `dir()`:

```
>>> dir("math")
```

Errores de ejecución

Si el sistema no puede ejecutar un comando producirá una excepción:

```
caramelos_de_ayer = 10
caramelos_de_hoy = "cuatro"
mis_caramelos = caramelos_de_ayer + caramelos_de_hoy
print(mis_caramelos)
```

Excepción:

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: unsupported operand type(s) for +: 'int' and 'str'

El error se produce al intentar sumar datos de distinto tipo. En estas circunstancias no se puede hacer uso del operador suma "+".

Modos para ejecutar código directamente:

Ejecutar un programa sin iniciar el entorno interactivo:

```
$ python3 ejemplo.py
```

Ejecutar un programa iniciando el entorno interactivo:

```
$ python3 -i ejemplo.py
```

Ejecutar el código Python contenido en una cadena de texto:

```
$ python3 -c 'for num in range(10): print(num)'
```

Relacionado:

- [Tutorial de lpython](#)
- [Ptpython: efectivo, cómodo y funcional](#)
- [Tutorial de JupyterLab](#)

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

Publicado por Pherkad en [11:55](#)



Etiquetas: [Python3](#), [Shell](#)

[Entrada más reciente](#)

[Inicio](#)

[Entrada antigua](#)

2014-2020 | Alejandro Suárez Lamadrid y Antonio Suárez Jiménez, Andalucía - España
. Tema Sencillo. Con la tecnología de [Blogger](#).