

Introducción a Python

por Ricardo Cárdenes

Intérprete

- Lenguaje interpretado
- Originalmente orientado a “scripting”
- Perfecto para uso general
 - excepto cálculo intensivo (módulos específicos)

Intérprete interactivo

```
$ python
```

```
>>> 1 + 2
```

```
3
```

```
>>> print "hola mundo"
```

```
hola mundo
```

Tipos de datos básicos

- Numéricos
 - enteros, coma flotante, número complejo
- Secuencias
 - cadenas de texto
 - tuplas
 - listas
- Diccionario (dict)
- Booleano
- None

Otros tipos de datos

- Creados por el usuario
 - clases
 - nuevos tipos (en C, por ejemplo)
- Tipos agregados en Python 2.4
 - conjunto (set)
 - número de coma fija (decimal)

Introspección

- Los objetos los podemos “inspeccionar” usando `dir`

```
>>> dir("hola mundo")
```

```
['__add__', '__class__', ... ]
```

- y podemos saber su tipo con `type`

```
>>> type("hola mundo")
```

```
<type 'str'>
```

Estructuras de control

- Condicional

```
if CONDICION:  
    bloque  
de  
    instrucciones
```

- Atención: ¡los espacios al principio de cada línea determinan el bloque al que pertenece!
- `pass` nos permite crear un bloque nulo

Estructuras de control

- Bucle condicional

```
while CONDICION:  
    bloque de  
    instrucciones
```

- Instrucción de continuación: `continue`
- Instrucción de terminación: `break`

Estructuras de control

- Bucle iterativo

```
for VARIABLE in SECUENCIA:  
    bloque de  
    instrucciones
```

```
else:  
    bloque de  
    instrucciones
```

- También podemos usar `continue` y `break`

Función

- Declaración básica:

```
def nombreFuncion( par1, par2, ... ):
    bloque de
    instrucciones
```

- En cualquier momento podemos salir usando `return` o devolver un valor con `return valor`
- Por omisión, se devuelve `None`

Módulos

- La funcionalidad de Python se incrementa usando **módulos**.
- Se puede **importar** sólo un módulo o partes de él

```
>>> import os, sys
```

```
>>> from modulo.otromodulo import func, SYM
```

Módulos

- Un módulo básico tiene la forma de un fichero normal y corriente, escrito en Python y con extensión `.py`
- También puede ser un módulo escrito en C (`.so`, `.pyd`, etc.)
- Ha de estar situado en el `PYTHONPATH` para poder cargarlo

Módulos

- También podemos organizar los módulos en **paquetes**.
- Un paquete es un directorio que contiene un fichero `__init__.py`
- Un paquete puede, a su vez, contener otros paquetes.
- Al contenido de un paquete se accede como si éste fuera un módulo.

Temas avanzados

- Listas por comprensión

```
[EXPRESSION for VARIABLES in SECUENCIA  
if CONDICION]
```

- También existen las expresiones generadoras
(Python 2.4)