Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ciencias de la Computación

CC3501:
Computación Gráfica, Visualización y Modelación para Ingenieros

Profesora Nancy Hitschfeld K.

Otoño 2012

Daniel Calderon S.

PYTHON

Artículo | Discusión | Leer | Editar | Ver historial | Buscar

Python

Este artículo trata sobre el lenguaje de programación. Para el grupo de humoristas, véase Monty Python.

Para otros usos de este término, véase Pitón.

Python es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico, es fuertemente tipado y multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License,[1] que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

Contenido [ocultar]

Python

Desarrollador(es)	
Python Software Foundation *http://www.python.org/*	
Información general	
Extensiones comunes	.py, .pyc, .pyd, .pyo, .pyw
Paradigma	multiparadigma: orientado a objetos, imperativo, funcional, reflexivo

python

- Lenguaje interpretado.
- Sintaxis extremadamente simple.
- Multiparadigma (Orientado a Objetos, Estructurado, etc…)
- No se especifican tipos de datos.
  - Se determinan automáticamente.
- Es necesario identar (o utilizar 4 espacios) para anidar código. No hay llaves {} !

Hola Mundo

- Basta con una línea ☺

```
print "Hola Mundo"
```

- Ejemplos en otros lenguajes
  - http://www.roesler-ac.de/wolfram/hello.htm

Variables

```
a = 12

b = "palabra"

c = 4.543

print a," ",b," ",c,"\t"
```

- No se especifican tipos, se determinan automáticamente.

Tipos numéricos

```
a = int("354")
# a = 354

b = 13
# b --> int

z =  float(13)
# z = 13.0 --> float

c = int(3.1415)
# c = 3

d = float("3.1415")
# d = 3.1415

e = a+b
# e --> int

f = c*d
# f --> float
```

- Operadores
  - +,-,*,/,**
  - a**b : ‘a’ elevado a ‘b’
- Cuidado!

```
g = 124/60
# g = 2

h = 124/60.0
# h = 2.0666666...
```

Cadenas de texto

- Similar a un arreglo.
- Caracteres especiales:
  - “\t”: tabulación
  - “\n”: salto de línea
  - “\r”: retorno de carro
  - “\\”: \
  - …

```
a = "python"

b = a[0]
#b = "p"

c = a[1:4]
#c = "yth"

d = c+b
# d = "ythp"

e = "cañón"
# e --> caÃ±Ã³n

f = u"cañón"
# f --> cañón

g = str(12.3)
# g --> "12.3"
```

Listas

- No se especifica el largo.
- Con “+” se unen 2 listas.
- Con “*” se amplifica la lista repitiendo elementos.
- La lista puede contener distintos tipos de elementos.
- len(b): entrega el largo de la lista b.

```
a = [1,2,3.1415,"hola"]

i = a[2]
# i = 3.1415

b = [1,2] + ["lista"]
# b = [1,2,"lista"]

b.append(815)
# b = [1,2,"lista",815]

c = [12,[1,2]]*2
# c = [12,[1,2],12,[1,2]]

n = len(b)

d = c.pop()
# c = [12,[1,2],12]
# d = [1,2]
```

Sintaxis básica: If/Else

```
if a < 50:
    print "menor a 50"
elif a < 100:
    print "entre 50 y 100"
else:
    print "mayor a 100"
```

Sintaxis básica: For y while

```
a = ["H","H","O"]

for ai in a:
    print "elemento: ",ai

for i in range(len(a)):
    print "elemento: ",a[i]

b = range(0,10,2)
#b = [0,2,4,6,8]

i = 0
while i < len(a):
    print "elemento: ",a[i]

    if i>100:
        break

    i += 1
```

- En el ‘for’ se irá recorriendo cada elemento en el arreglo.
- Se puede utilizar ‘break’ para salir del loop.

Funciones

```
def masMenos(a,b):
    c = a+b
    d = a-b

    return [c,d]

[v1,v2] = masMenos(23,12)
[u,v] = masMenos(1.2,0.5)
[a,b] = masMenos(1.2,13)
```

- El código se ejecuta con variables que sean válidas para las operaciones involucradas.
- Si la instrucción no se lee, no se arrojará error.

Clases y Objetos

- Los 3 ingredientes:
  - Variables de instancia o atributos
  - Constructor
  - Métodos
- Objeto: Instancia de una clase
- Objeto ~ Elemento
- Clase ~ Conjunto

```
class Vector:
    def __init__(self,x,y):
        self.a = [0,0]
        self.a[0] = x
        self.a[1] = y

    def x(self):
        return self.a[0]

    def y(self):
        return self.a[1]

    def setX(self,x):
        self.a[0] = x

    def setY(self,y):
        self.a[1] = y
```

```
v = Vector(12,23)
v = Vector(12,3.1415)

print "v = (",v.x(),",",v.y(),")"

v.setX(88)
```

Librerías!!

Math	• Operaciones matemáticas
PyGame	• SDL, manejo de eventos
PyOpenGL	• Wrapper d eOpenGL
PIL	• Imágenes
Numpy	• Manejo numérico (~Matlab)
SciPy	• Computación científica (~Matlab)
Matplotlib	• Gráficos (~Matlab)
PyOpenCV	• Visión Computacional
PyODE	• Dinámica
Etc…	

Documentación oficial

- http://www.python.org/doc/
- http://docs.python.org/tutorial/