

Auxiliar nº4:  
Introducción a

# Python

Maite González Mendoza  
30 de Agosto, 2012.

# En esta clase veremos:

Sintaxis:

- Variables
- Condicionales
- Ciclos
- Funciones
- Clases
- Objetos



# Variables:

- Las variables son dinámicas
- No hay que definir un tipo

**x=10**

**y=1.0**

**x=y**

**z="hola"**

**y=z**

# Condicionales:

```
if condicion:  
    instrucciones  
elif condicion:  
    instrucciones  
else:  
    instrucciones
```

La condición no  
va entre paréntesis,  
Y lleva los “ : ”  
Al final

## Indentación:

Las instrucciones siempre se encuentran  
un nivel más adentro

# Ciclos y listas:

```
while condicion:  
    instrucciones
```

```
a=[] #inicializa lista vacía  
a=[1,2,3] #inicializa lista  
        con elementos  
a.append(20) #agrega al final  
            de la lista  
for elemento in a:  
    instrucciones
```

# Ejemplo

```
edad=0
```

```
while edad<18:
```

```
    edad=edad+1
```

```
    if edad==18:
```

```
        print "Ya soy mayor de edad!"
```

```
    else:
```

```
        print "Tengo "+str(edad)+" años."
```

# Funciones:

```
def funcion(param1, param2, ...):  
    instrucciones  
    return var1, var2, ...
```

No es necesario definir previamente qué es lo que retorna la función, y puede retornar varios valores  
La cantidad de parámetros puede ser cero, uno o muchos.

# Ejemplos

```
def sumaCuadrados(x, y):  
    z=x**2+y**2  
    return z
```

```
def Cuadrados(x, y):  
    xc=x**2  
    yc=y**2  
    return xc, yc
```

# Clases Y Objetos

```
class ClaseNueva:  
    def __init__(self, arg1,...):  
        self.varDeInstancia=arg1  
        self.otraVarDeInstancia="yujuu\0/"  
        .  
        .  
    def metodo(self, arg2,...):  
        Instrucciones  
    def otroMetodo(self):  
        MasInstrucciones  
class OtraClaseNueva:  
    def __init__(self, arg3=3):  
        self.vardeInst=arg1  
        .
```

Todos los métodos, incluido el constructor, reciben como parámetro a self

# Clases Y Objetos

```
class ClaseNueva:
```

```
    def __init__(self, arg1, ...):
```

```
        self.varDeInstancia=arg1
```

```
        self.otraVarDeInstancia="yujuu\0/"
```

```
        .
```

```
        .
```

```
    def metodo(self, arg2, ...):
```

```
        Instrucciones
```

```
    def otroMetodo(self):
```

```
        MasInstrucciones
```

```
class OtraClaseNueva:
```

```
    def __init__(self, arg3=3):
```

```
        self.vardeInst=arg1
```

```
        .
```

Constructor

Métodos

El parámetro arg3=3  
Dice que su valor  
por defecto es 3

# Ejemplo

```
class Coche:
```

```
    #Abstraccion de los objetos coche.
```

```
    def __init__(self, gasolina):
```

```
        self.gasolina = gasolina
```

```
        print "Tenemos", gasolina, "litros"
```

```
    def arrancar(self):
```

```
        if self.gasolina > 0:
```

```
            print "Arranca"
```

```
        else:
```

```
            print "No arranca"
```

```
    def conducir(self):
```

```
        if self.gasolina > 0:
```

```
            self.gasolina -= 1
```

```
            print "Quedan", self.gasolina, "litros"
```

```
        else:
```

```
            print "No se mueve"
```

# Ejemplo de uso de Clases y Objetos

```
>mi_coche = Coche(3)
>mi_coche.arrancar()
>mi_coche.conducir()
>mi_coche.conducir()
>mi_coche.conducir()
>mi_coche.conducir()
>mi_coche.arrancar()
>print mi_coche.gasolina
```

- Tenemos 3 litros
- Arranca
- Quedan 2 litros
- Quedan 1 litros
- Quedan 0 litros
- No se mueve
- No arranca
- 0

# Instalación de Python

# • Instalacion de Python en Linux

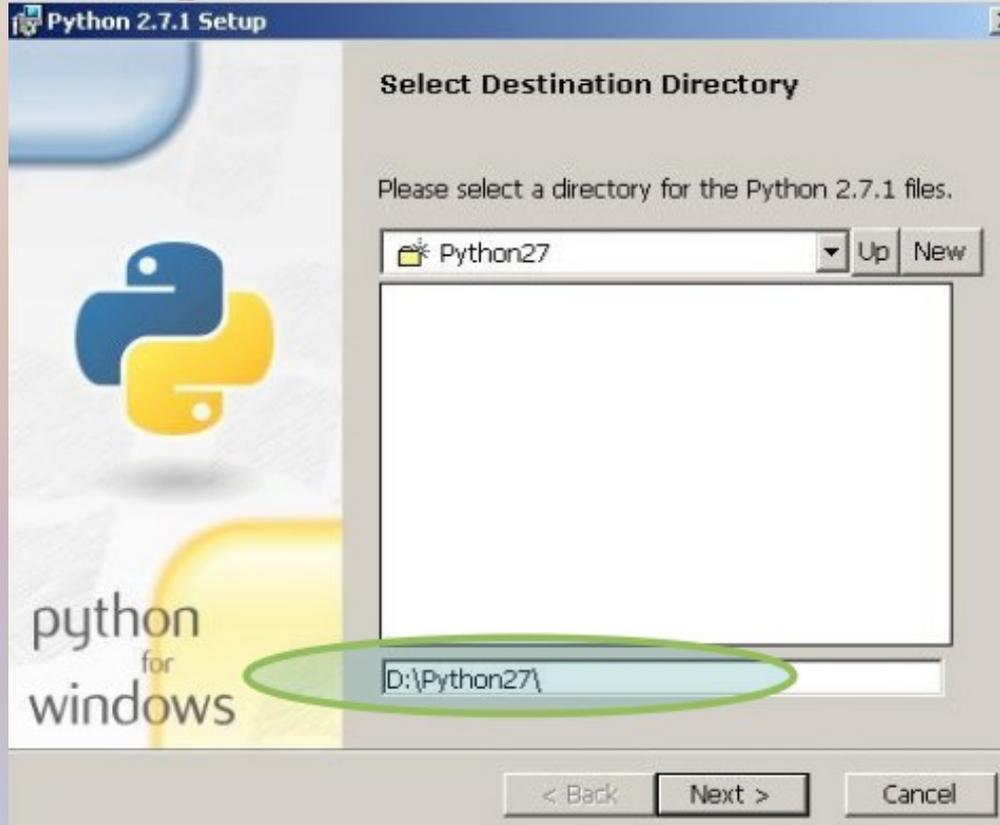
- Ubuntu: `$sudo apt-get install python`
- Archlinux: `$sudo pacman -S python2`
- Ejecución en Linux:
  - Ubuntu: `$python`
  - Archlinux: `$python2`

```
maite@maite-EasyNote-NM85: ~  
maite@maite-EasyNote-NM85:~$ sudo apt-get install python  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias  
Leyendo la información de estado... Hecho  
python ya está en su versión más reciente.  
0 actualizados, 0 se instalarán, 0 para eliminar y 0 no actualizados.  
maite@maite-EasyNote-NM85:~$
```

# Instalación de Python en Windows

- Instalación en Windows:

Python-2.7.1.msi

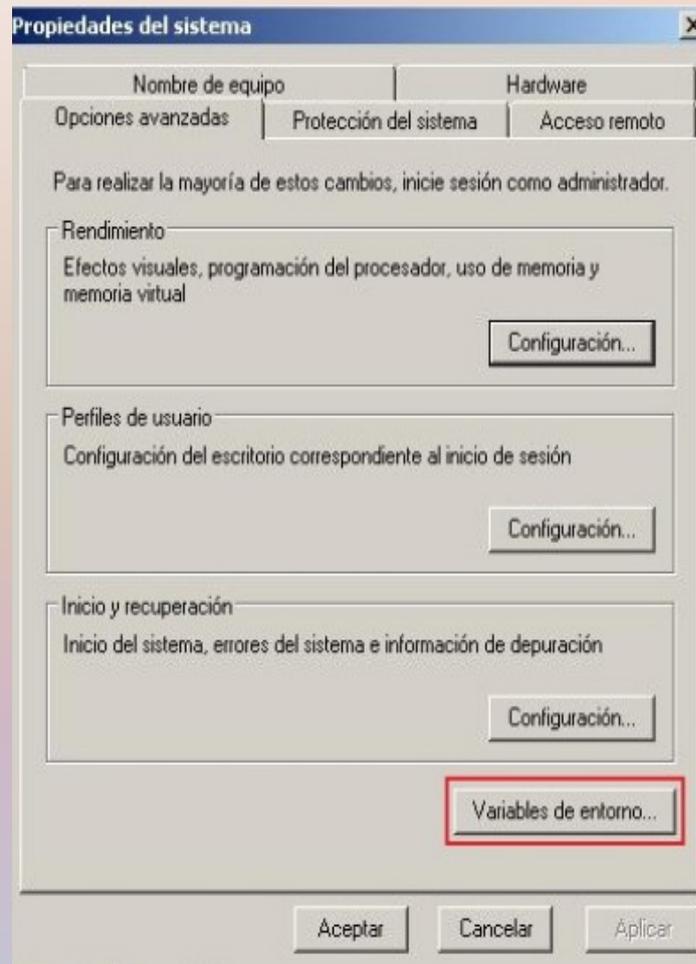


Recordar Directorio

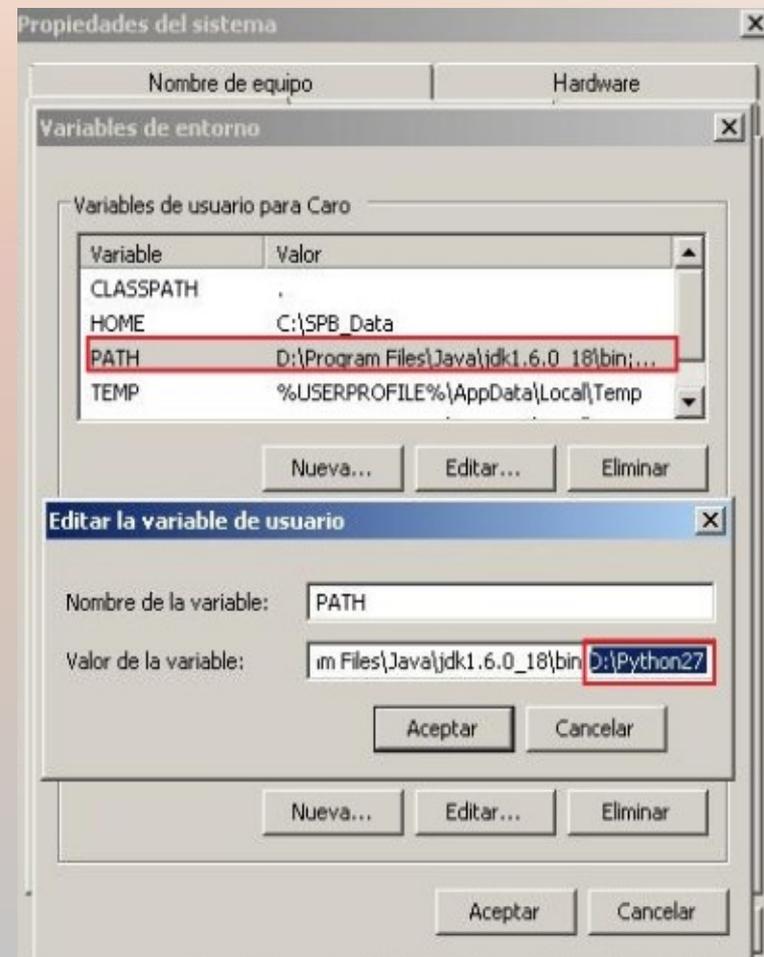


NO modificar nada

# Instalación de Python en Windows



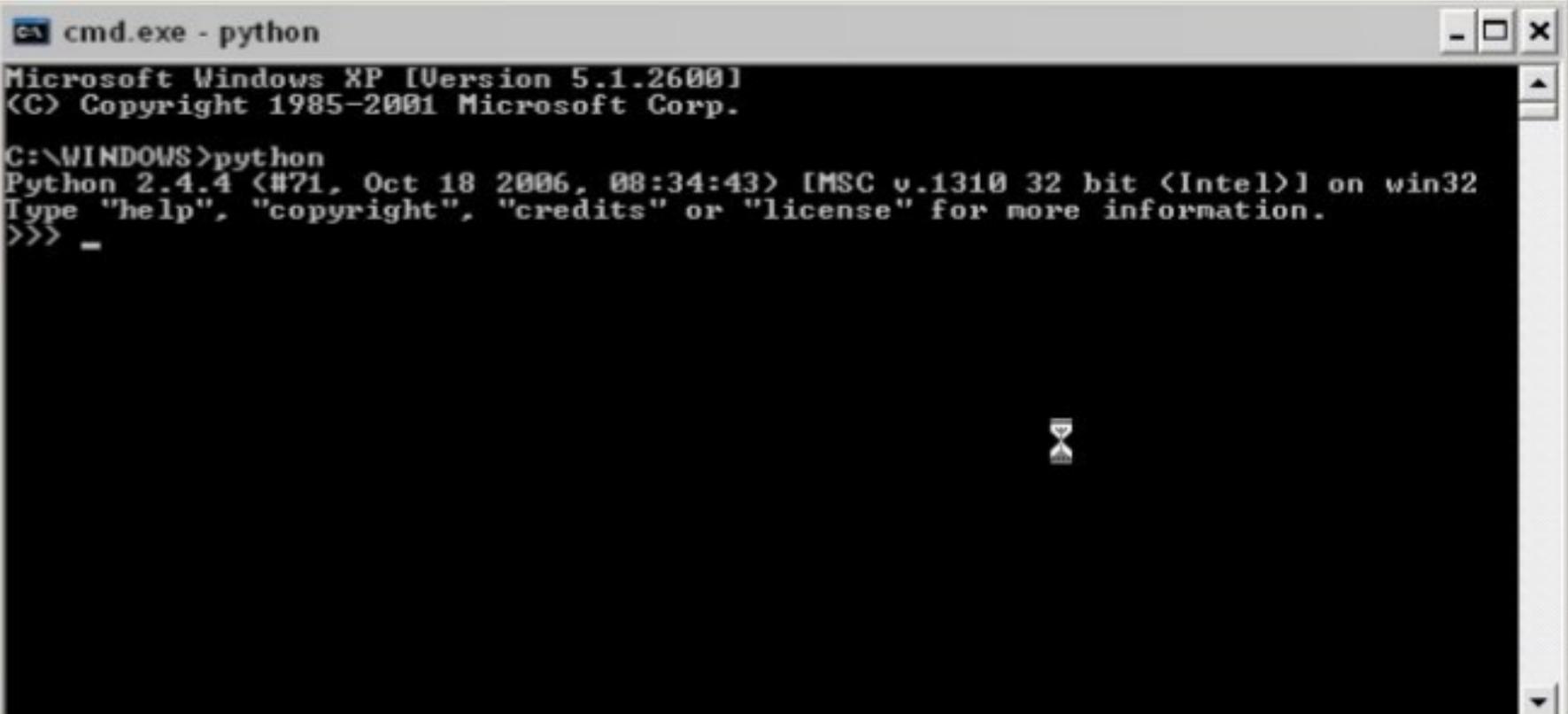
Abrir propiedades del sistema (win+pause) y modificar variables de entorno



Agregar en la variable 'Path' la el directorio donde instaló python

# Ejecución de Python en Windows

Inicio → ejecutar → 'cmd' → >python



```
cmd.exe - python
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS>python
Python 2.4.4 (#71, Oct 18 2006, 08:34:43) [MSC v.1310 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```