



Cómo involucrarse en GNOME extendiendo las aplicaciones

Germán Poo Caamaño

Proyecto GNOME

gpoo@gnome.org

<http://www.gnome.org/~gpoo/>

<http://www.calcifer.org/>

Involucrate+

Lima, Perú

Marzo 2008



Python

- Generalidades
 - Lenguaje libremente disponible, interpretado y orientado a objetos
 - No provoca conflictos en la comunidad
- Farándula
 - Creado por Guido van Rossum en 1990
 - El nombre proviene de la serie Monty Python
- Disponibilidad
 - En las principales distribuciones, también en otros ambientes distintos de Linux/Unix



Ejemplos básicos

```
$ python  
>>> print "Hola Lima"  
Hola Lima  
>>> 5*60+500  
800  
>>>
```



Programas

```
#!/usr/bin/env python  
print "Hola Lima"
```

```
$ python programa.py  
Hola Lima  
$ chmod 755 programa.py  
$ ./programa.py  
Hola Lima
```



Variables y expresiones

- Similar a otros lenguajes

`5 ** 100`

`5 * (10 + 650)`

`"Hola" + "Lima"`

- Asignación

`nombre = "Rupert"`

`iva = 1.19`

`neto = 1000`

`bruto = neto * iva`

`bruto = "Persona bruta"`



Estructuras de condición

- **if-else**

```
if a == b:
```

```
    print "a y b son iguales"
```

```
else:
```

```
    print "a y b son distintos"
```

- **pass**

```
if a == b:
```

```
    pass
```

```
else:
```

```
    print "a y b son distintos"
```



Tipos básicos

- **Números**

$a = 5$

$b = 4.5$

$c = 12345646L$

$d = 1 + 2j$

- **Cadenas**

$a = \text{"Uso de doble comillas"}$

$a = \text{'Uso de comillas simples'}$

$a = \text{""Asignacion de
multiples lineas""}$



Tipos básicos: Listas

- Listas

```
a = [ 5, 4.5, 'Rupert', "The monkey"]
```

```
b = []
```

```
c = [ 1, [ 4, 5 ] ]
```

```
d = a + c
```

- Manipulación

```
print a[1]
```

```
print a[1:3]
```

```
a[0] = 50
```




Tipos básicos: Listas

- Métodos

```
a = []
```

```
a.append('GUADEC')
```

```
a.append('Soy Rupert')
```

```
a.insert(0, 'Hola')
```

```
len(a)
```

```
del(a[1])
```



Tipos básicos: Tuplas

- Como listas, pero de largo fijo
- Elementos inmutables

```
a = ( 1, 2, 3)
```

```
b = (,)
```

```
print a[1]
```

```
print a[0:2]
```



Tipos básicos: Diccionarios

- Son arreglos asociativos

```
a = { 'rupert': 5, 'gpoo': 20 }
```

```
b = { }
```

```
c = { 'nombre': 'Rupert', 'edad' : 7 }
```

```
a['rupert'] = 10
```

```
d = c['nombre']
```

```
print a.keys()
```



Iteraciones

- **while**

```
a = 1
```

```
while a < 50:
```

```
    a = a + 1
```

- **for**

```
for i in "Hola Lima":
```

```
    print i
```

```
for i in [ 1, 2, 3, 4, 5]:
```

```
    print i
```

```
for i in range[1, 5]:
```

```
    print i
```



Funciones

```
def max(a, b):
```

```
    if a > b:
```

```
        return a
```

```
    else:
```

```
        return b
```

```
def max2(a, b)
```

```
    if a > b:
```

```
        return a, b
```

```
    else:
```

```
        return b, a
```

```
(x, y) = max2(5, 10)
```



Classes

```
class Account:
```

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

```
        self.amount = initial
```

```
    def deposit(self, amount):
```

```
        self.amount += amount
```

```
    def withdraw(self, amount):
```

```
        self.amount -= amount
```

```
    def get_balance(self):
```

```
        return self.amount
```



Classes

```
a = Account(5000)
```

```
a.deposit(2500)
```

```
a.deposit(4000)
```

```
a.withdraw(7000)
```

```
a.get_balance()
```



Excepciones

- try/except

try:

```
b = 5 / 0
```

except:

```
print "Parece que 5 no es divisible por 0"
```




Programas

```
#!/usr/bin/env python  
print "Hola Lima"
```

```
$ python programa.py  
Hola Lima  
$ chmod 755 programa.py  
$ ./programa.py  
Hola Lima
```

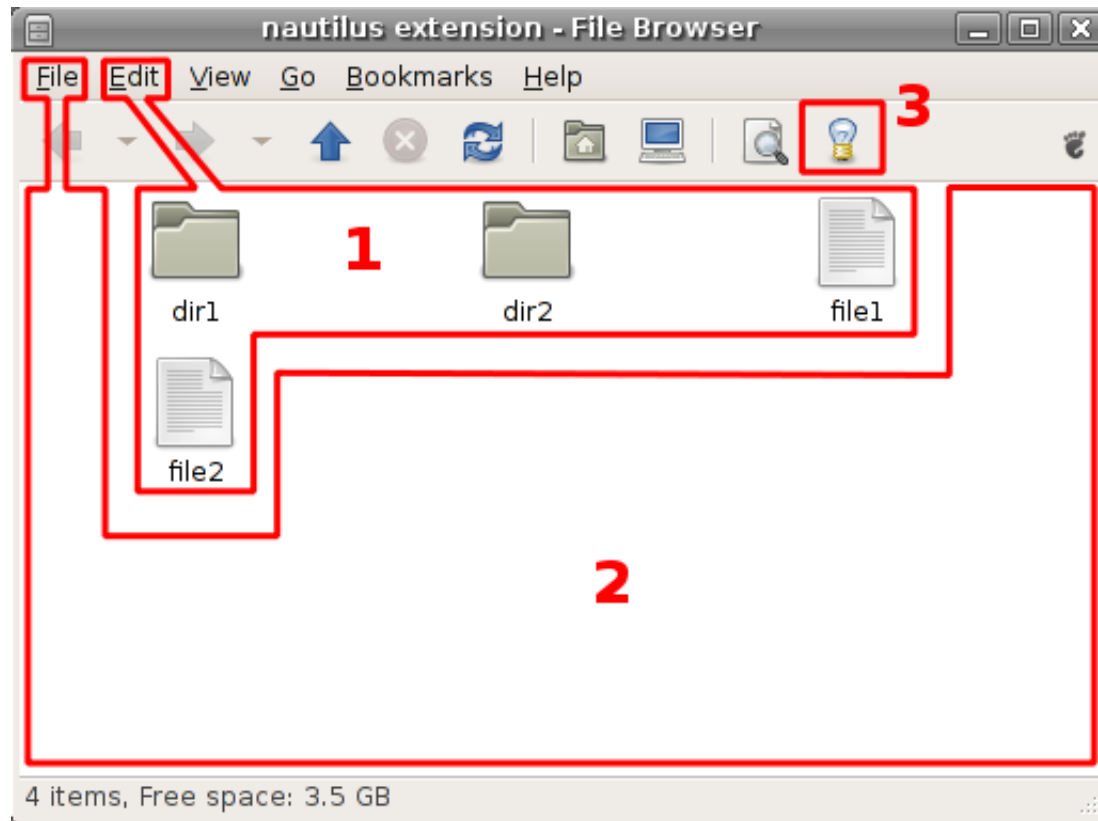


Extensiones





Ejemplo: ¿Qué extender en Nautilus?





Proveedores

- **InfoProvider:** Información y emblemas
- **ColumnProvider:** Añadir columnas a la vista de columnas e íconos
- **MenuProvider:** Menú contextual, carpeta activa y barra de herramientas
- **PagePropertyProvider:** Añadir páginas de información en el diálogo de propiedades



Manos a la obra (Demostración)