

# Auxiliar 1

## CC1000-5 Computación I

Profesor Auxiliar:  
**Carlos González Cortés**  
[carlgonz@ug.uchile.cl](mailto:carlgonz@ug.uchile.cl)



# Introducción a Python



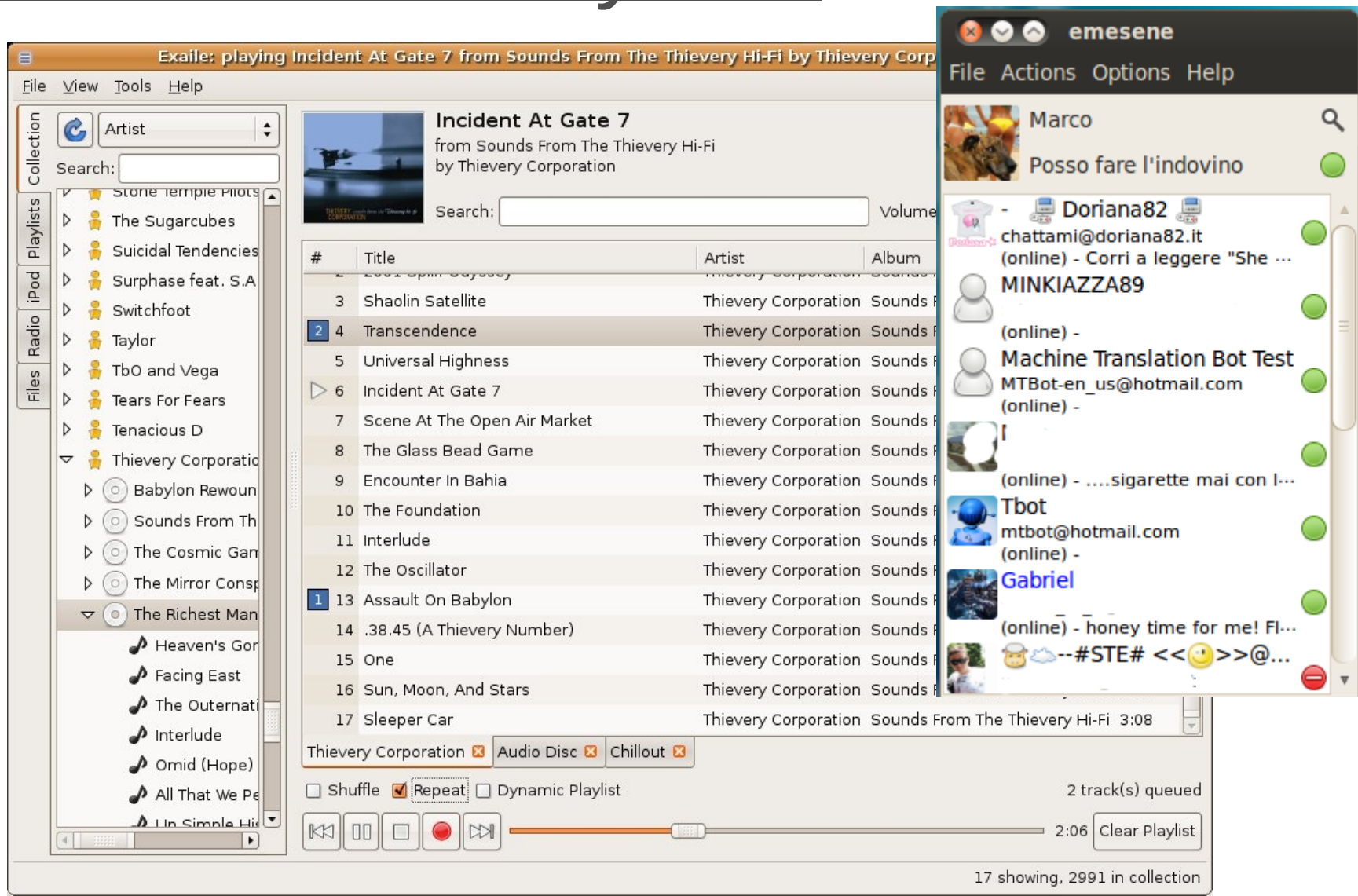
- Características
  - Leguaje interpretado
  - Multiplataforma
  - Tipado dinámico – Fuertemente Tipado
  - Orientado a objetos
- Ventajas:
  - Sintaxis limpia y sencilla. Alto nivel
- Desventajas:
  - No adecuado a bajo nivel

# Introducción a Python

Position Mar 2011	Position Mar 2010	Delta in Position	Programming Language	Ratings Mar 2011	Delta Mar 2010	Status
1	1	=	Java	19.711%	+2.20%	A
2	2	=	C	15.262%	-2.02%	A
3	4	↑	C++	8.754%	-0.86%	A
4	6	↑↑	C#	7.210%	+2.95%	A
5	3	↓↓	PHP	6.566%	-3.34%	A
6	7	↑	Python	5.737%	+1.51%	A
7	5	↓↓	(Visual) Basic	4.710%	-1.86%	A
8	12	↑↑↑↑	Objective-C	3.518%	+1.55%	A
9	8	↓	Perl	1.969%	-1.85%	A
10	10	=	JavaScript	1.866%	-0.78%	A
11	11	=	Ruby	1.498%	-0.83%	A
12	-	=	Assembly*	1.345%	-	A
13	9	↓↓↓↓	Delphi	0.997%	-1.69%	A
14	13	↓	Go	0.958%	+0.04%	A-
15	21	↑↑↑↑↑	Lisp	0.934%	+0.38%	A
16	24	↑↑↑↑↑↑	Lua	0.812%	+0.30%	A-
17	32	↑↑↑↑↑↑↑	Ada	0.726%	+0.35%	A-
18	18	=	Pascal	0.706%	+0.10%	A
19	38	↑↑↑↑↑↑↑↑	NXT-G	0.640%	+0.34%	B
20	-	=	Scheme*	0.634%	-	B

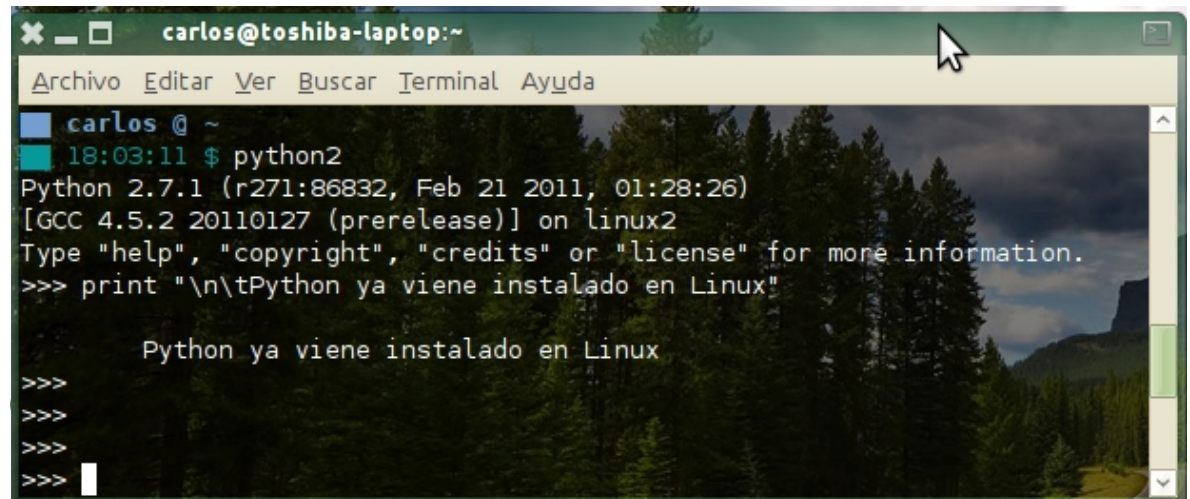
Fuente: TIOBE Software. [www.tiobe.com](http://www.tiobe.com)

# Introducción a Python



# Instalacion de Python

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

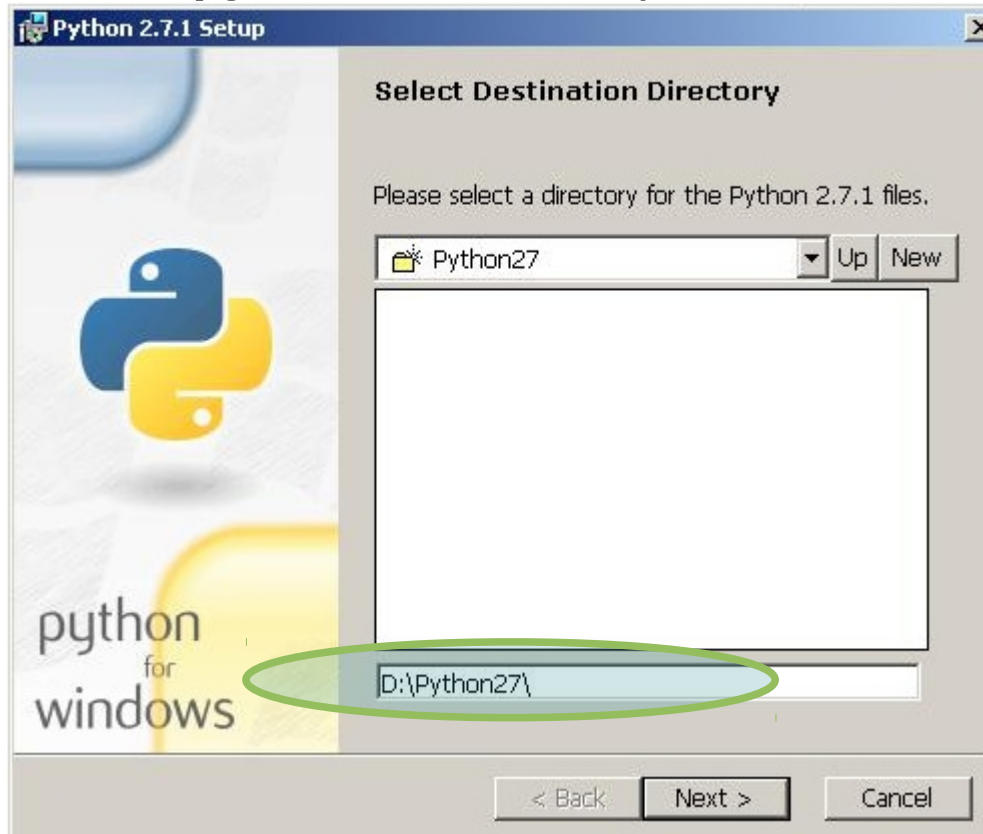


```
carlos@toshiba-laptop:~  
Archivo Editar Ver Buscar Terminal Ayuda  
carlos @ ~  
18:03:11 $ python2  
Python 2.7.1 (r271:86832, Feb 21 2011, 01:28:26)  
[GCC 4.5.2 20110127 (prerelease)] on linux2  
Type "help", "copyright", "credits" or "license" for more information.  
>>> print "\n\tPython ya viene instalado en Linux"  
  
        Python ya viene instalado en Linux  
>>>  
>>>  
>>>  
>>>
```



# Instalacion de Python

- Instalación en Windows:
  - python-2.7.1.msi (Material docente)



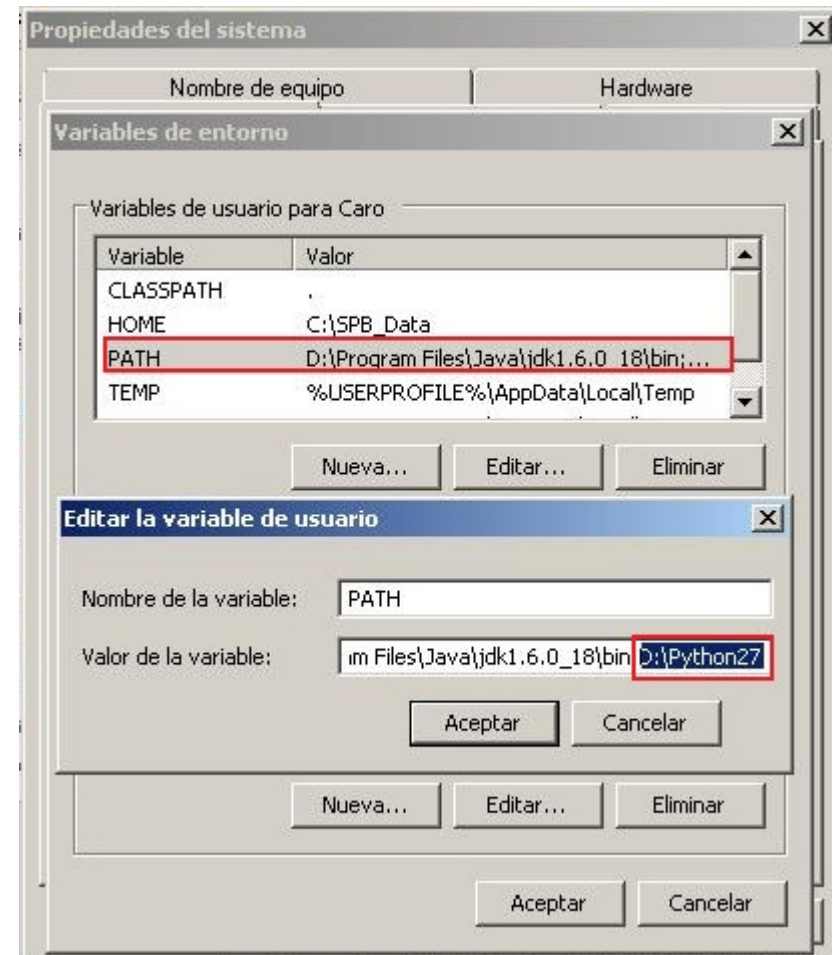
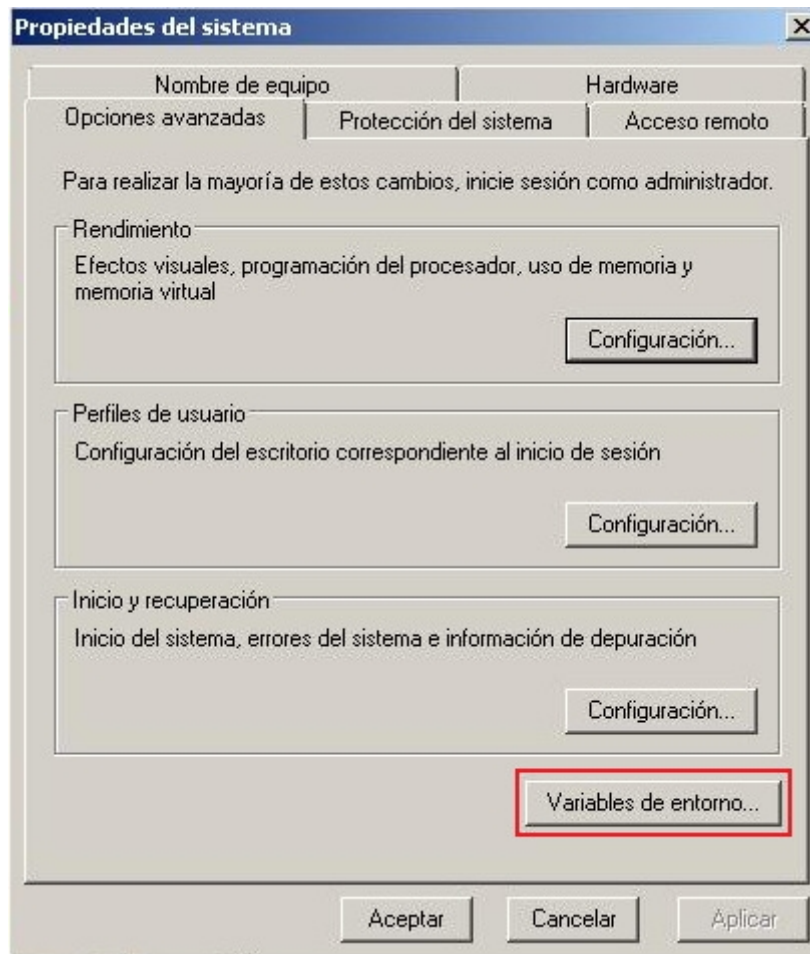
**Recordar directorio de instalación**



**No modificar nada**

# Instalacion de Python

- Instalación en Windows:

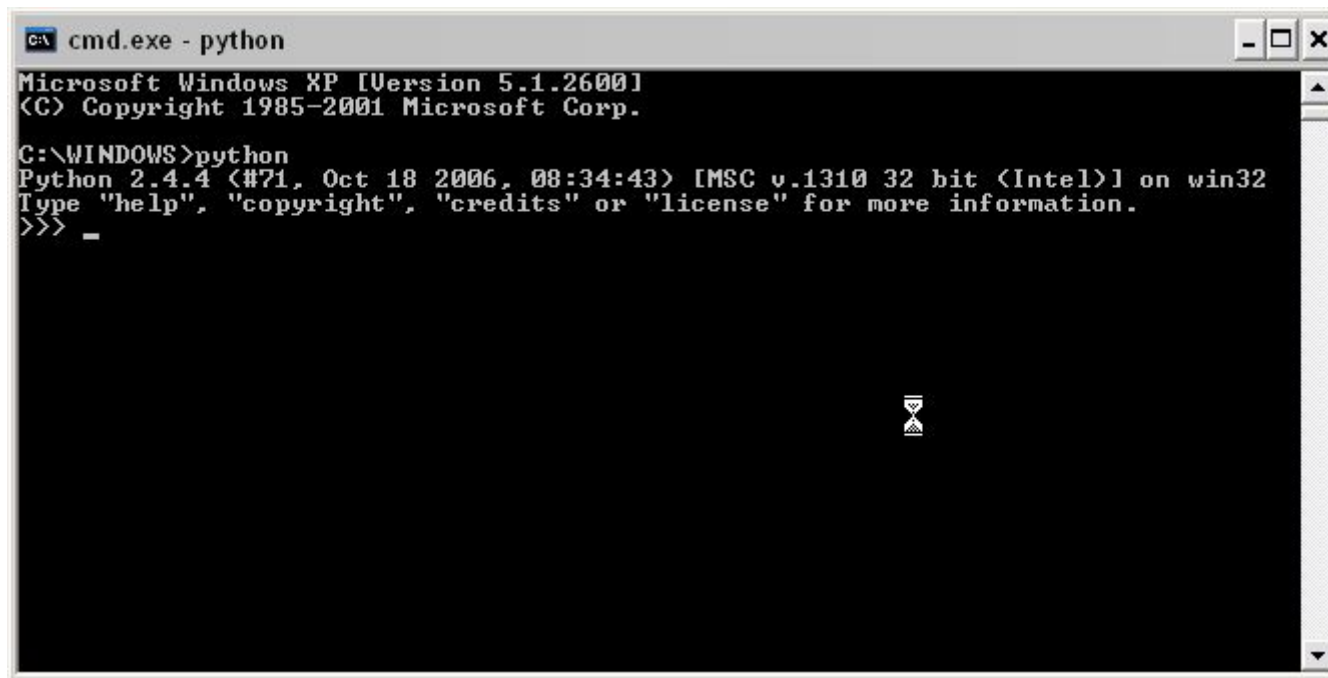


**Abrir propiedades del sistema (win+pause) y modificar variables de entorno**

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

# Instalacion de Python

- Ejecución en Windows:
  - Inicio → ejectuar → '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.
>>> _
```



# Programación en Python

- Programación
  - Consola de Python
  - IDE (<http://wiki.python.org/moin/PythonEditors>)
- Sintaxis
  - Palabras claves
    - Resaltadas por el editor
    - No se pueden usar como nombres de funciones o variables
  - Identación
    - No hacen falta llaves
    - Separa bloques de código (ejemplo)

# Programación en Python

- Variables

- Un espacio de memoria. Almacena datos. Posee un nombre, un valor y un tipo.

- Int: enteros [-2.147.483.648 , 2.147.483.647]
- Double: reales [ $\pm 2,2250738585072020 \times 10^{-308}$   $\pm 1,7976931348623157 \times 10^{308}$ ]
- String: texto (cadena de caracteres)

- Sintaxis:

```
>>> x=10
```

```
>>> y=10.0
```

```
>>> x=y
```

```
>>> z="Hola mundo"
```

```
>>> y=z
```

# Programación en Python

- Operaciones matemáticas

Operación	Operador	Aridad	Asociatividad	Precedencia
Exponenciación	**	Binario	Por la derecha	1
Identidad	+	Unario	—	2
Cambio de signo	-	Unario	—	2
Multiplicación	*	Binario	Por la izquierda	3
División	/	Binario	Por la izquierda	3
Módulo (o resto)	%	Binario	Por la izquierda	3
Suma	+	Binario	Por la izquierda	4
Resta	-	Binario	Por la izquierda	4

- Ejemplo:

>>3.6\*100/3

>>3.6\*(100/3)

# Programación en Python

- Operaciones matemáticas

Operación	Operador	Aridad	Asociatividad	Precedencia
Exponenciación	**	Binario	Por la derecha	1
Identidad	+	Unario	—	2
Cambio de signo	-	Unario	—	2
Multiplicación	*	Binario	Por la izquierda	3
División	/	Binario	Por la izquierda	3
Módulo (o resto)	%	Binario	Por la izquierda	3
Suma	+	Binario	Por la izquierda	4
Resta	-	Binario	Por la izquierda	4

- Ejemplo:

```
>>3.6*100/3  
120.0
```

```
>>3.6*(100/3)  
118.8
```

# Programación en Python

- Funciones:
  - Alteran el flujo de ejecución de un programa
  - Permiten reutilizar código redundante
- Sintaxis:

```
def funcion (param1, param2, ....):
```

```
    instrucción 1
```

```
    ....
```

```
    instrucción n-1
```

```
return variable
```



# Programación en Python

- Condicionales:
  - Alteran el flujo de ejecución de un programa
  - Permiten ejecutar códigos dependiendo del caso
- Sintaxis:

```
if condicion :  
    instrucción 1  
    ...  
    instrucción n  
elif condicion:  
    instrucción n+1  
    ...  
    instrucción m  
else:  
    instrucción m+1  
    ...
```

# Programación en Python

- Operadores lógicos

Operación	Operador	Aridad	Asociatividad	Precedencia
Negación	<b>not</b>	Unario	—	alta
Conjunción	<b>and</b>	Binario	Por la izquierda	media
Disyunción	<b>or</b>	Binario	Por la izquierda	baja

operador	comparación
<b>==</b>	es igual que
<b>!=</b>	es distinto de
<b>&lt;</b>	es menor que
<b>&lt;=</b>	es menor o igual que
<b>&gt;</b>	es mayor que
<b>&gt;=</b>	es mayor o igual que