

## Lenguaje Python Versión 3 (3.x.y)

- Python oficial de python.org
- Anaconda Python
- WinPython
- Python(x,y)
- ActivePython
- Pydroid 3 (Android)

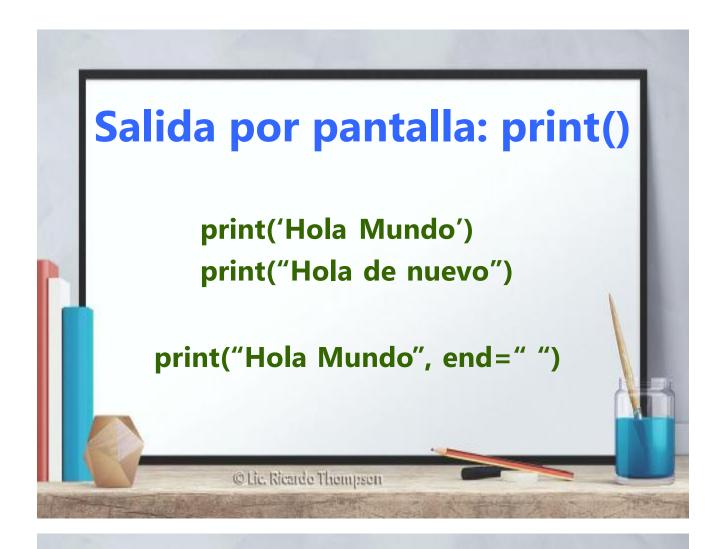
## ¿Qué es Python?

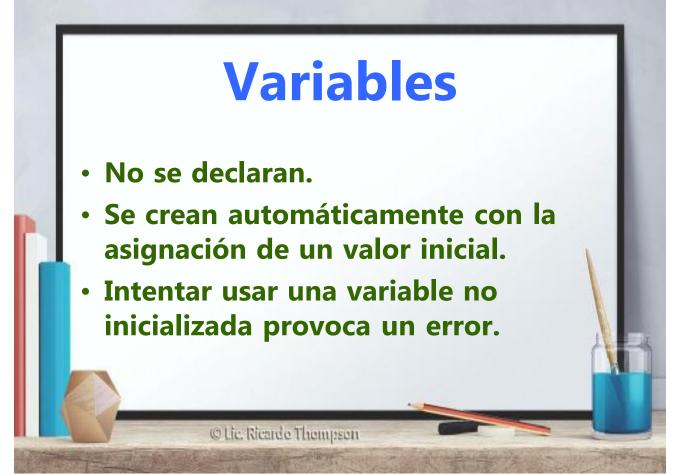
- Es un lenguaje de muy alto nivel.
- Es multiparadigma.
- Es interpretado: No se compila.
- El espíritu de Python ("zen") privilegia la legibilidad del código.
- Es portable.
- Es sensible a mayúsculas y minúsculas.

@ Lie. Ricarde Thempsen

# Estructura de los programas en Python

- No tienen un formato rígido.
- No hay llaves ni puntos y coma.
- Las instrucciones terminan con Enter.
- Los comentarios comienzan con el signo numeral: #





## **Variables**

Reglas para crear nombres de variables:

- Sólo letras, números y el guión bajo.
- No pueden comenzar con un número.
- No pueden coincidir con las palabras reservadas del lenguaje.

@ Lic. Ricarde Thempsen

### **Variables**

- Los nombres de variables <u>deben</u> tener sentido.
- Deben evitarse variables llamadas "l" (ele minúscula) u "o" (letra o) porque pueden confundirse fácilmente con números.



- Los caracteres se consideran cadenas de longitud 1.
- Hay valores booleanos: True y False.
- Se permite la asignación múltiple: a,b,c = 3, "Lunes", 5.18

@ Lic. Ricarde Thempsen

## **Variables**

 Para imprimir variables se las separa de las constantes con una coma:

dia = 5
print("Hoy es", dia)



 Puede lograrse mayor control de la salida impresa con el operador %:

print("Precio: %5.2f" %precio)

No va coma luego de cerrar comillas

© Lie. Ricarde Thempsen

### **Variables**

 Los especificadores de conversión son los mismos que en Lenguaje C:

> %d para números enteros %f para números reales %c para caracteres



 Si hay más de una variable, éstas deben encerrarse entre paréntesis:

$$print("X = %4d - Y = %4d" %(x,y))$$

@ Lie. Ricarde Thempsen

## **Variables**

 Para ingresar valores por teclado se utiliza la función input():

a = input("Mensaje")

input() <u>siempre</u> devuelve un string.



Existen funciones para convertir este string a otros tipos de dato:

```
n = int(input("Mensaje"))
```

r = float(input("Mensaje"))

@ Lie. Ricarde Thempsen

## **Operadores aritméticos**

+ Suma

- Resta
- \* Multiplicación / División real
- // División entera % Módulo o resto
- \*\* Potenciación

#### Orden de evaluación

- 1. Potenciación
- 2. Menos unario
- 3. Multiplicación, división y módulo
- 4. Suma y resta

© Lie. Ricarde Thempsen

## Asignación extendida

$$a += 1 \Leftrightarrow a = a + 1$$

$$a = 2 \Leftrightarrow a = a - 2$$

$$a *= 3 \Leftrightarrow a = a * 3$$

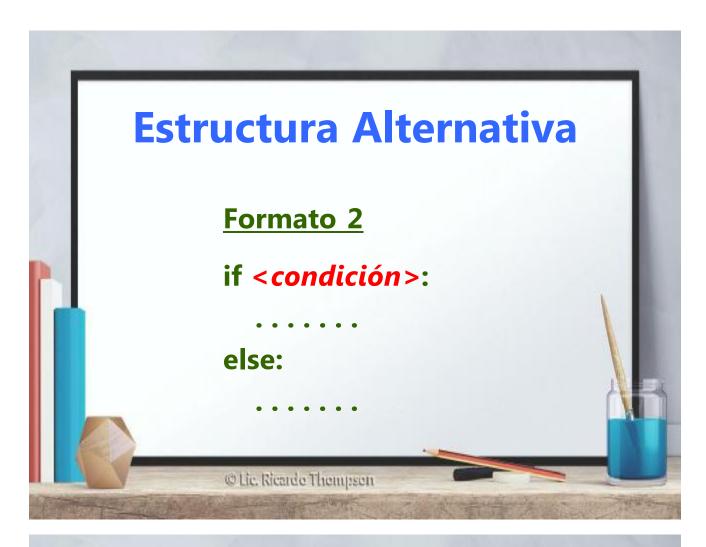
$$a/=4 \Leftrightarrow a=a/4$$

$$a^{**} = 5 \iff a = a^{**} = 5$$

No hay operadores incrementales









#### **Estructura Alternativa**

- Las condiciones y el else van seguidas del carácter "dos puntos".
- La sangría o indentación es lo que establece el alcance del if.
- Python recomienda una sangría standard de 4 espacios, sin tabs.
  - La sangría debe ser uniforme.

@ Lic. Ricarde Thempsen

## **Operadores relacionales**

== igual

> mayor

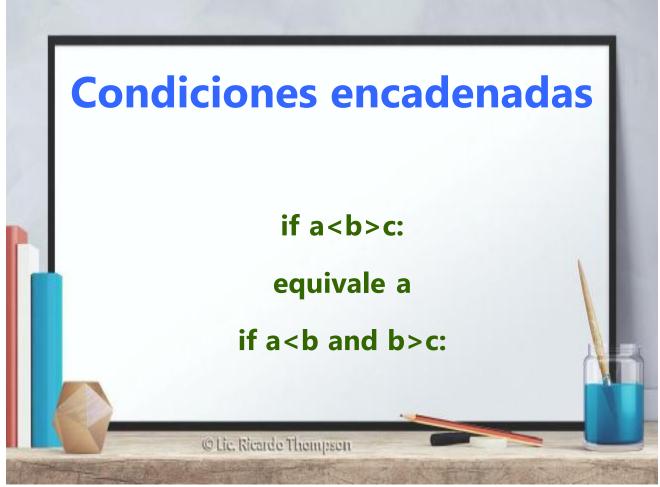
< menor

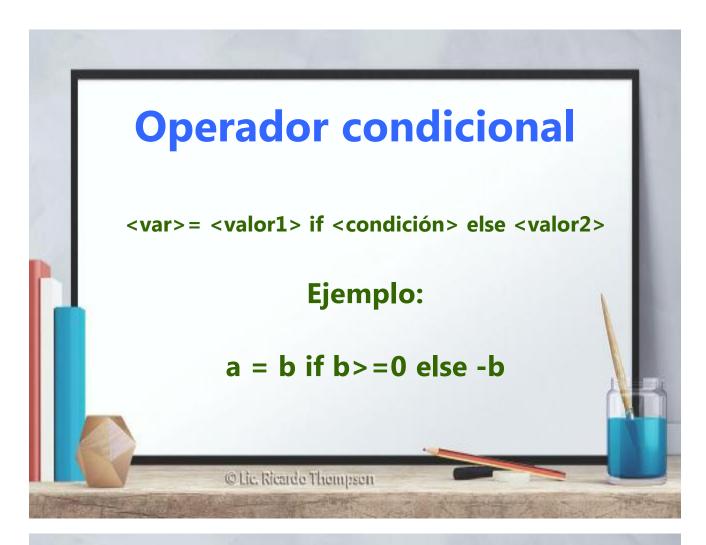
>= mayor o igual

<= menor o igual

!= distinto









## Ejemplo 1

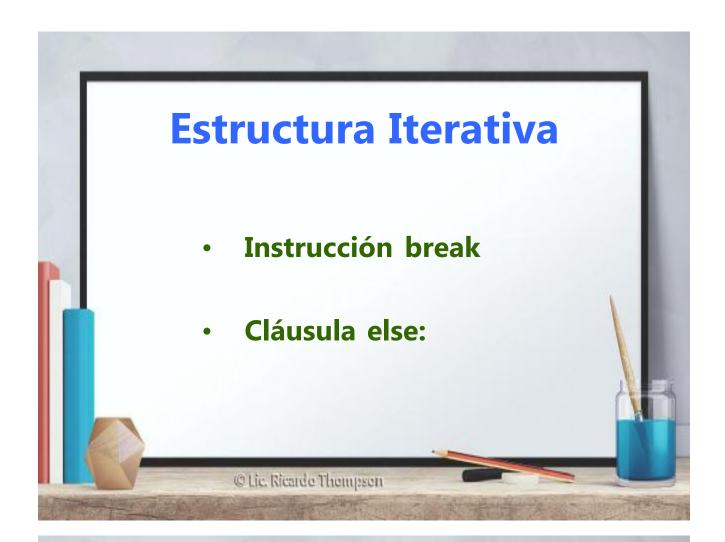
Ingresar un número entero e imprimir el nombre del mes correspondiente (1: Enero, 2: Febrero, 3: Marzo...)

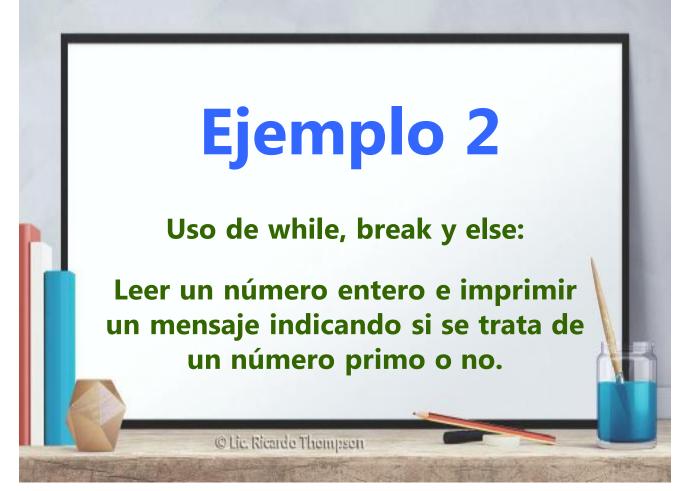
```
mes = int(input("Mes?"))
if mes == 1:
    print("Enero")
elif mes == 2:
    print("Febrero")
elif mes == 3:
    print("Marzo")
[.....]
elif mes == 12:
    print("Diciembre")
else:
    print("Mes inválido")
```



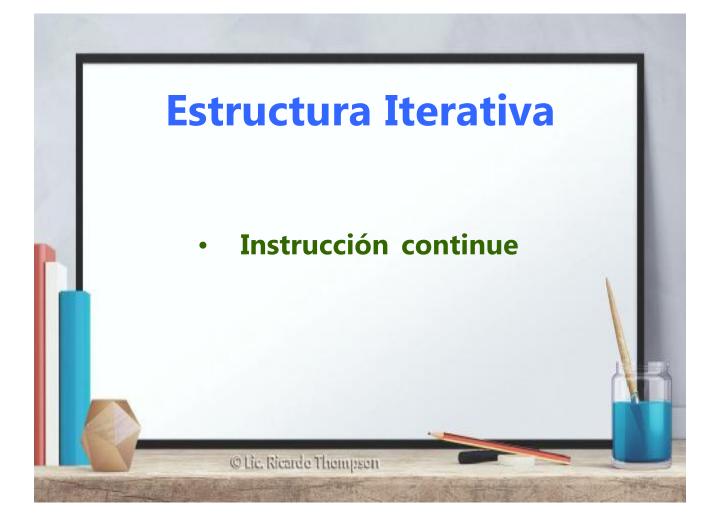
#### **Estructura Iterativa**

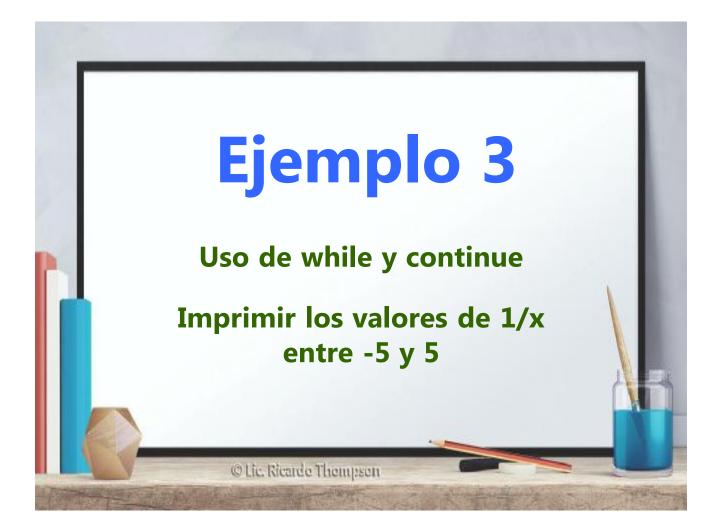
- La condición va seguida del carácter "dos puntos".
- La sangría o indentación es lo que establece el alcance del ciclo.
- Python recomienda una sangría standard de 4 espacios, sin tabs.
  - La sangría debe ser uniforme.

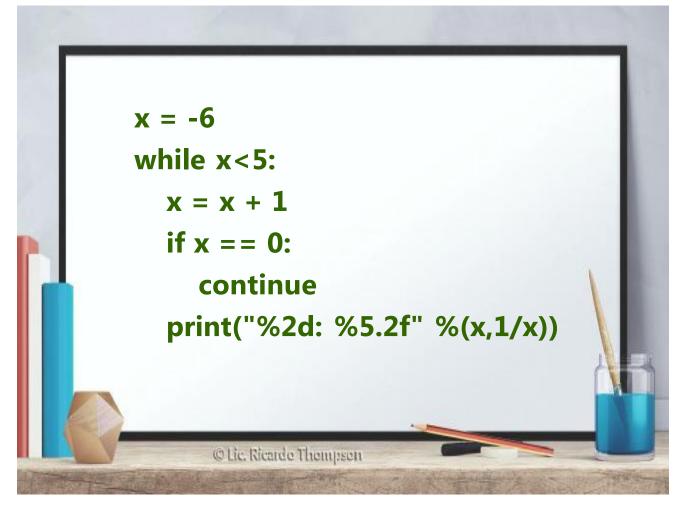




```
n = int(input("Ingrese un número "))
i = 2
while i<n:
if n % i == 0:
    print(n,"no es un número primo")
    break
i = i + 1
else:
    print(n,"es un número primo")
```









#### **Estructura Iterativa**

- for se utiliza para recorrer una secuencia.
- Esta secuencia puede ser cualquier iterable: Un rango, una lista, una cadena, un archivo, una tupla o un diccionario.



- Genera una secuencia de números enteros.
- Admite tres formas de utilización.

@ Lic. Ricarde Thempsen

## Función range()

#### Formato 1:

range(<vfinal>)

- Genera una secuencia de números enteros entre 0 y <vfinal>.
- <vfinal> no está incluido.

## Función range()

#### Formato 2:

range(<vinicial>, <vfinal>)

- Genera una secuencia de enteros entre <vinicial> y <vfinal>.
- <vfinal> no está incluido.

@ Lie. Ricarde Thempsen

### Función range()

#### Formato 3:

range(<vinicial>, <vfinal>, <inc>)

- Genera una secuencia de enteros entre <vinicial> y <vfinal> con incremento <inc>.
- <vfinal> no está incluido.

## Función range()

#### Formato 3:

range(<vinicial>, <vfinal>, <inc>)

- El incremento puede ser negativo.
- Las instrucciones break y continue hacen lo mismo que en while.

@ Lic. Ricarde Thempsen

## **Ejemplo 4**

Uso de for y range()

Imprimir los números impares entre 1 y N

```
n = int(input("Ingrese un número: "))
for i in range(1, n+1, 2):
   print(i, end=" ")
print()
```

## **Ejercitación**

- Buscar en Internet el *Zen de Python* y analizarlo.
- Escribir un programa para resolver el antiguo acertijo chino que se describe a continuación:
   "He contado 35 cabezas y 94 patas entre las gallinas y los conejos de mi granja. Cuántos conejos y cuántas gallinas tengo?"
   (Sugerencia: Verificar todas las combinaciones posibles hasta hallar la correcta).