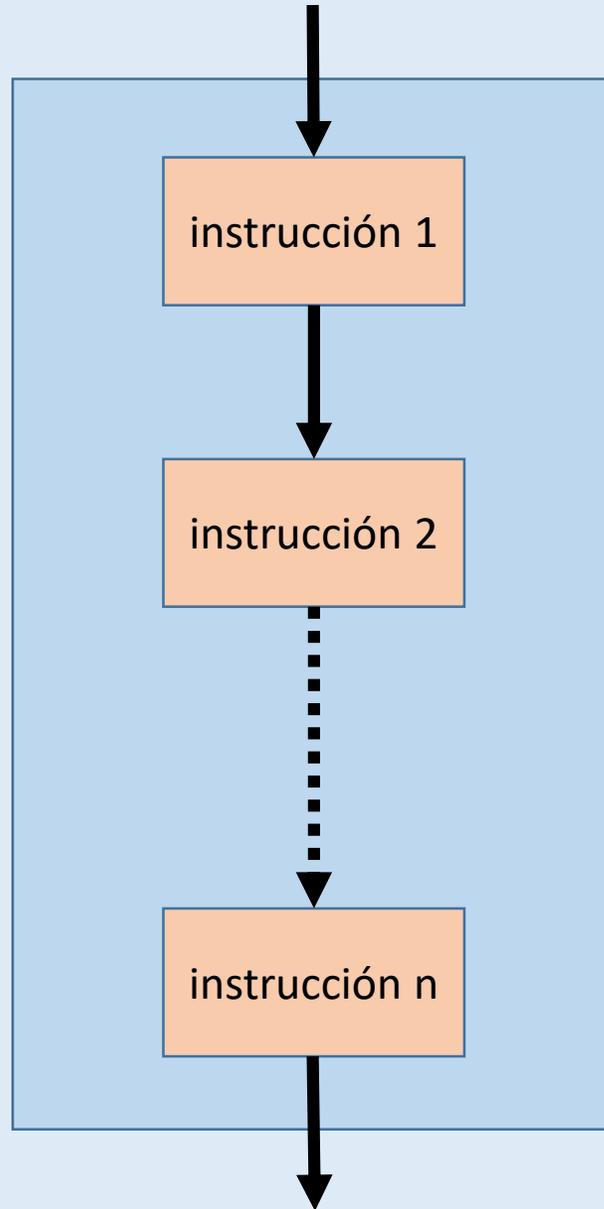


# Introducción a la programación

con Python

José A. Corrales

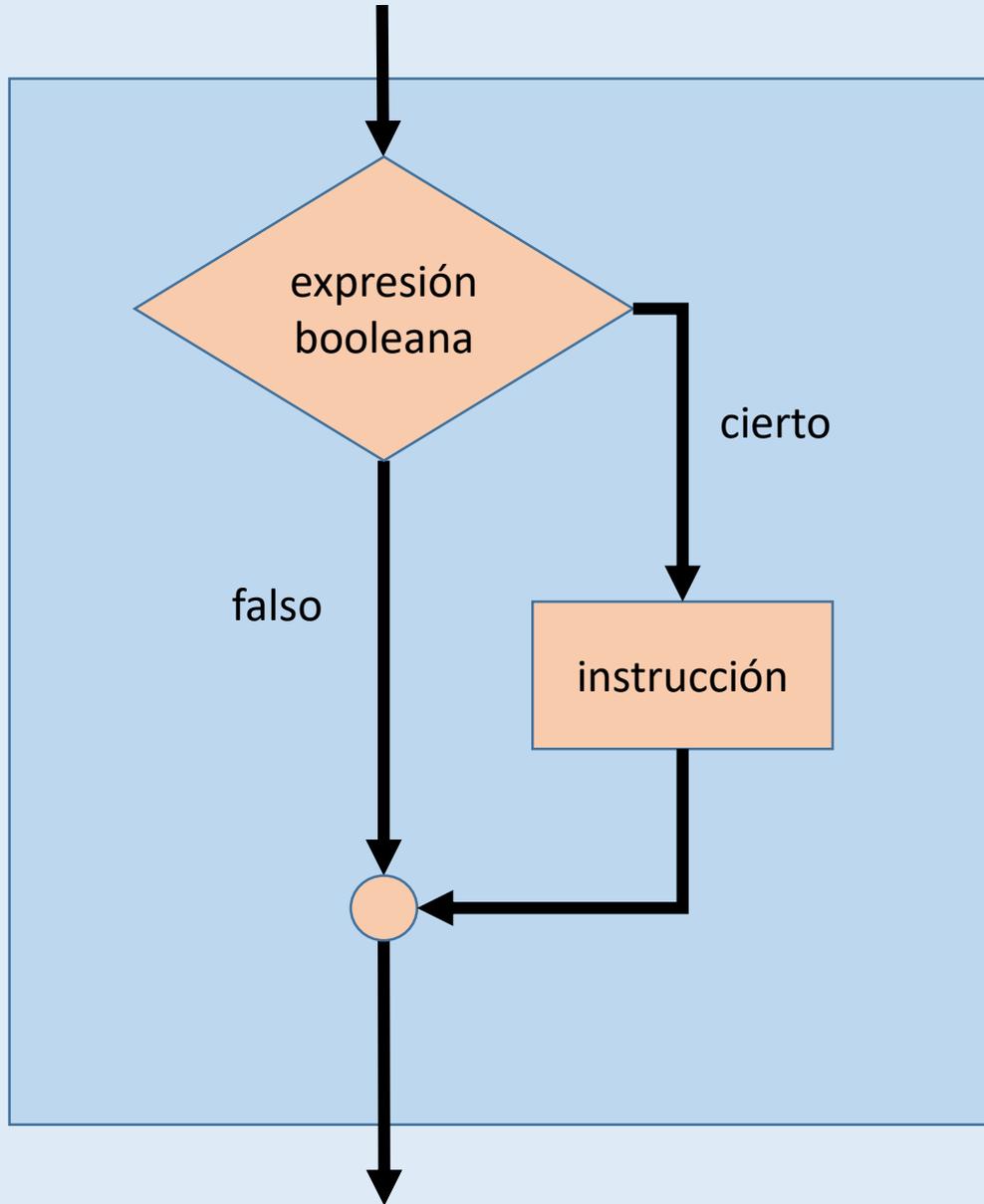
ja@uniovi.es



programa "algodón"

- prelavado 10 m
- lavado 30 m
- aclarado 5 m
- centrifugado 15 m
- secado 30 m

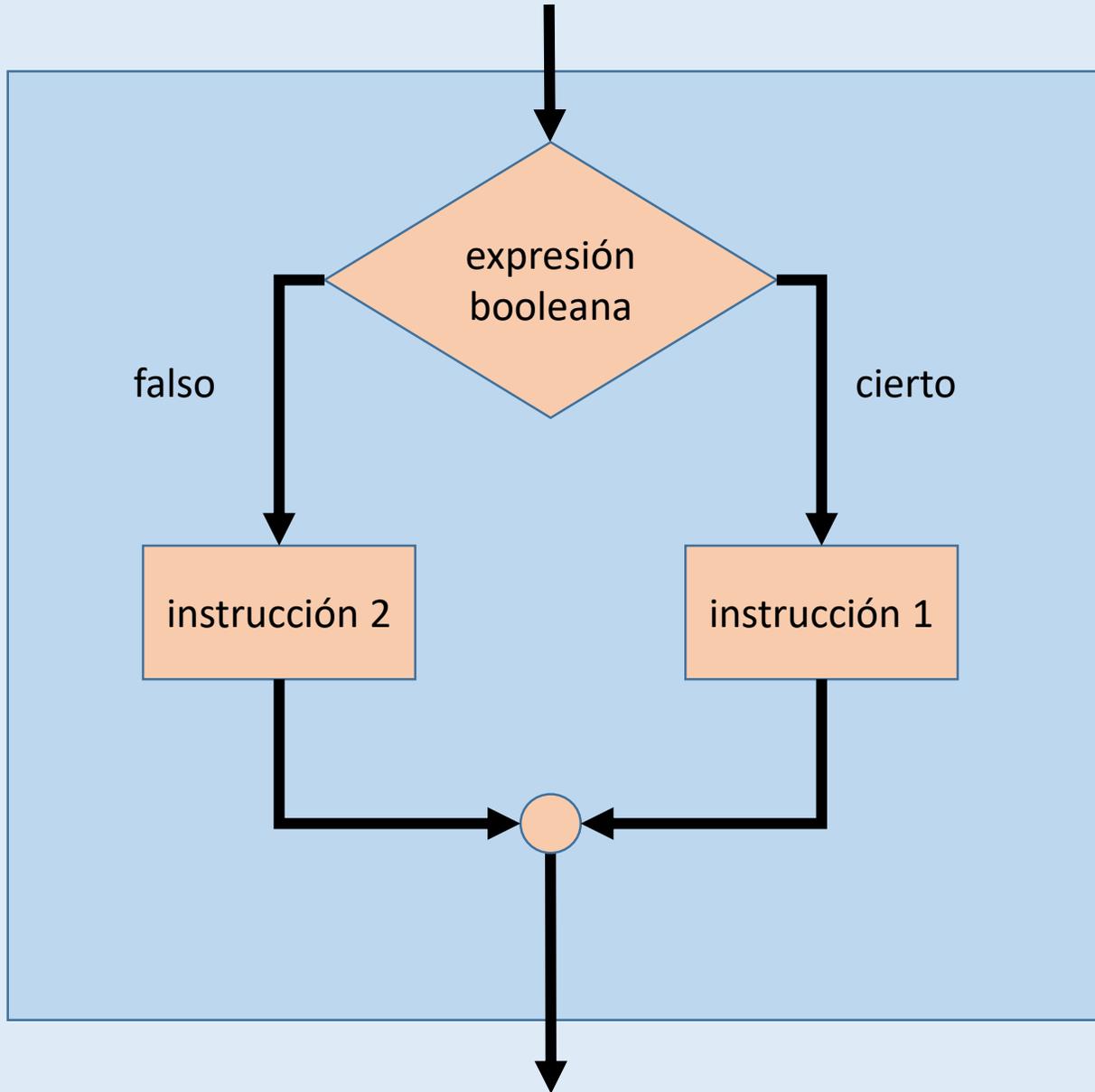
secuencia o  
instrucción secuencial



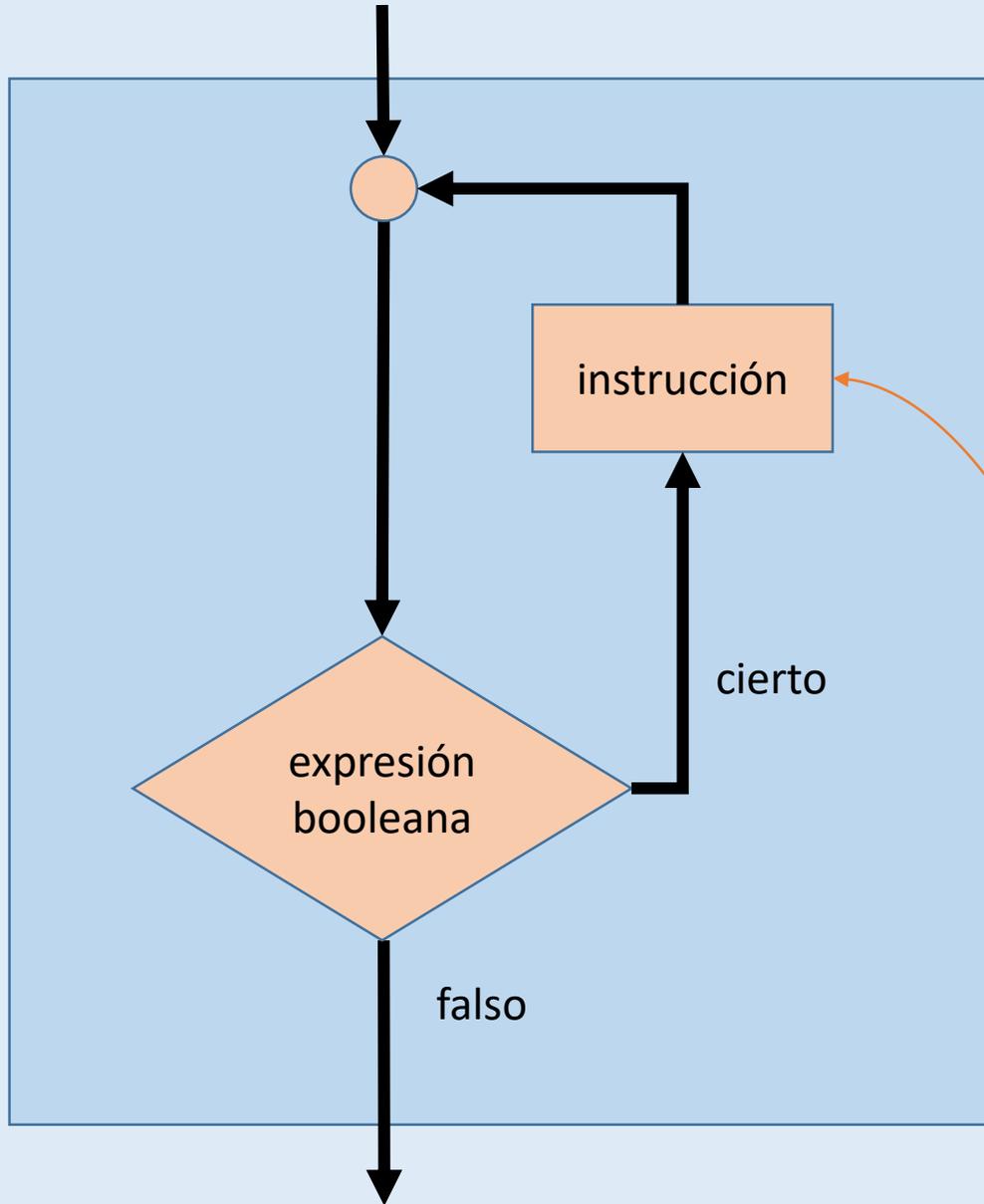
programa "salir de paseo"

- asearse
- vestirse
- si dan lluvia
- coger un paraguas
- salir de casa
- si ya está lloviendo
- abrir el paraguas
- caminar

condición o instrucción  
condicional simple



condición o instrucción  
condicional completa

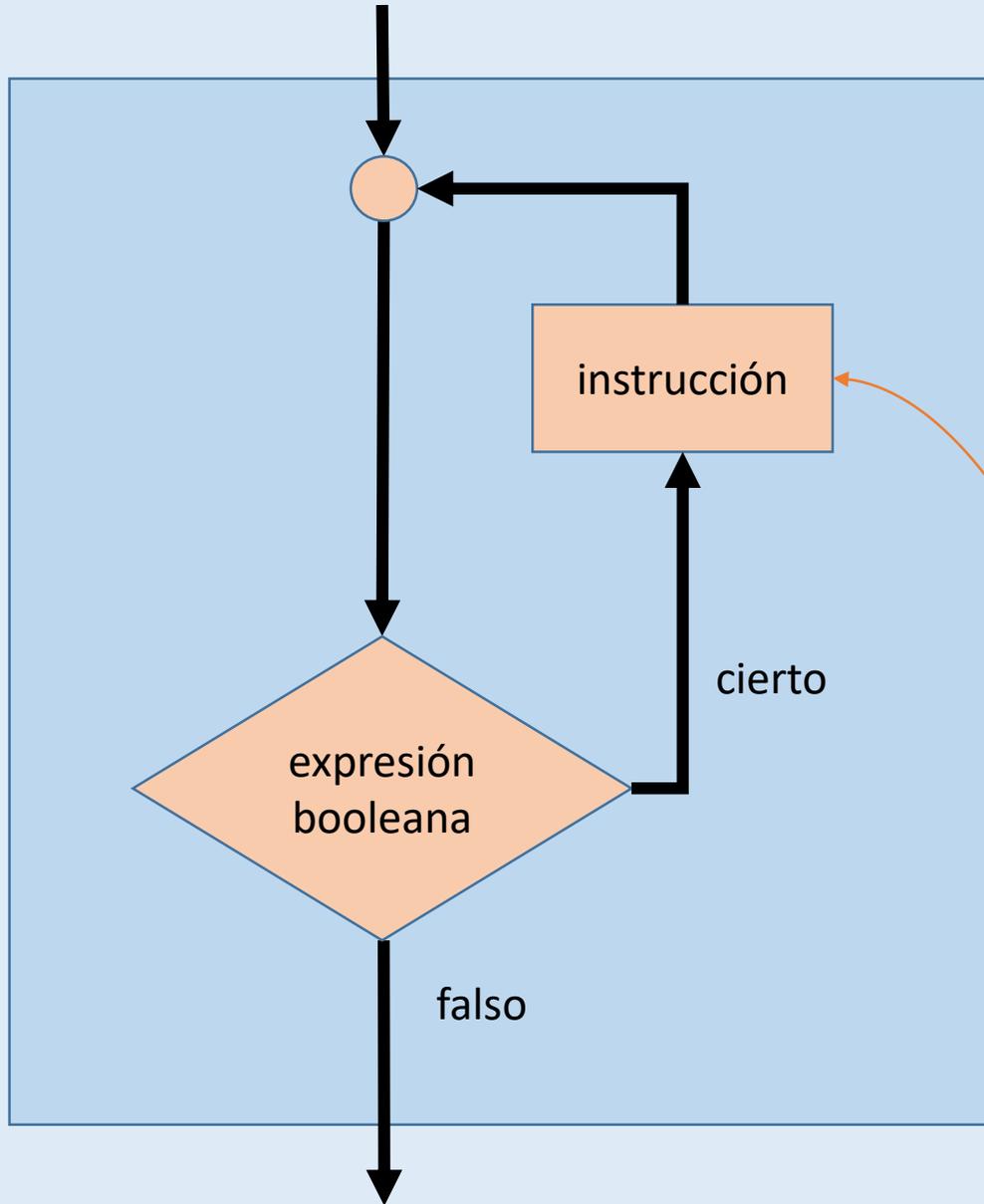


programa "comer bolsa de patatas"

- abrir la bolsa de patatas
- **mientras queden patatas**
- **sacar la siguiente patata**
- **comer patata**
- tirar y reciclar la bolsa

bucle o iteración o  
instrucción repetitiva  
de tipo "mientras"

cuerpo del  
bucle



programa "freír tres huevos"

- coger la sartén
- ponerle aceite
- ponerla al fuego y esperar
- repetir tres veces
- romper y echar un huevo
- esperar a que dore
- retirar el huevo
- esperar a que enfríe el aceite
- reciclar aceite
- lavar y recoger la sartén

bucle o iteración o  
instrucción repetitiva  
de tipo "para"

cuerpo del  
bucle

instrucción de asignación

forma genérica: `<variable>` `<operador de asignación>` `<expresión>`

ejemplos:

```
n = 97
```

```
alfa = 2*3.1416
```

```
apellidos = "Fernández Álvarez"
```

```
fuerza = masa * aceleración
```

```
Contador_1 = 0
```

```
positivo = alfa > 0
```

```
Z = (1 + 2i) / 5.8
```

**nombre de variable**: debe comenzar por letra (en cualquier idioma) y puede seguir con letras, dígitos o el guion bajo \_

**operador de asignación**: el signo igual =

**expresión**: cualquier expresión válida para cada tipo de dato (numérico, caracteres, booleano, etc...)

memoria del intérprete de Python

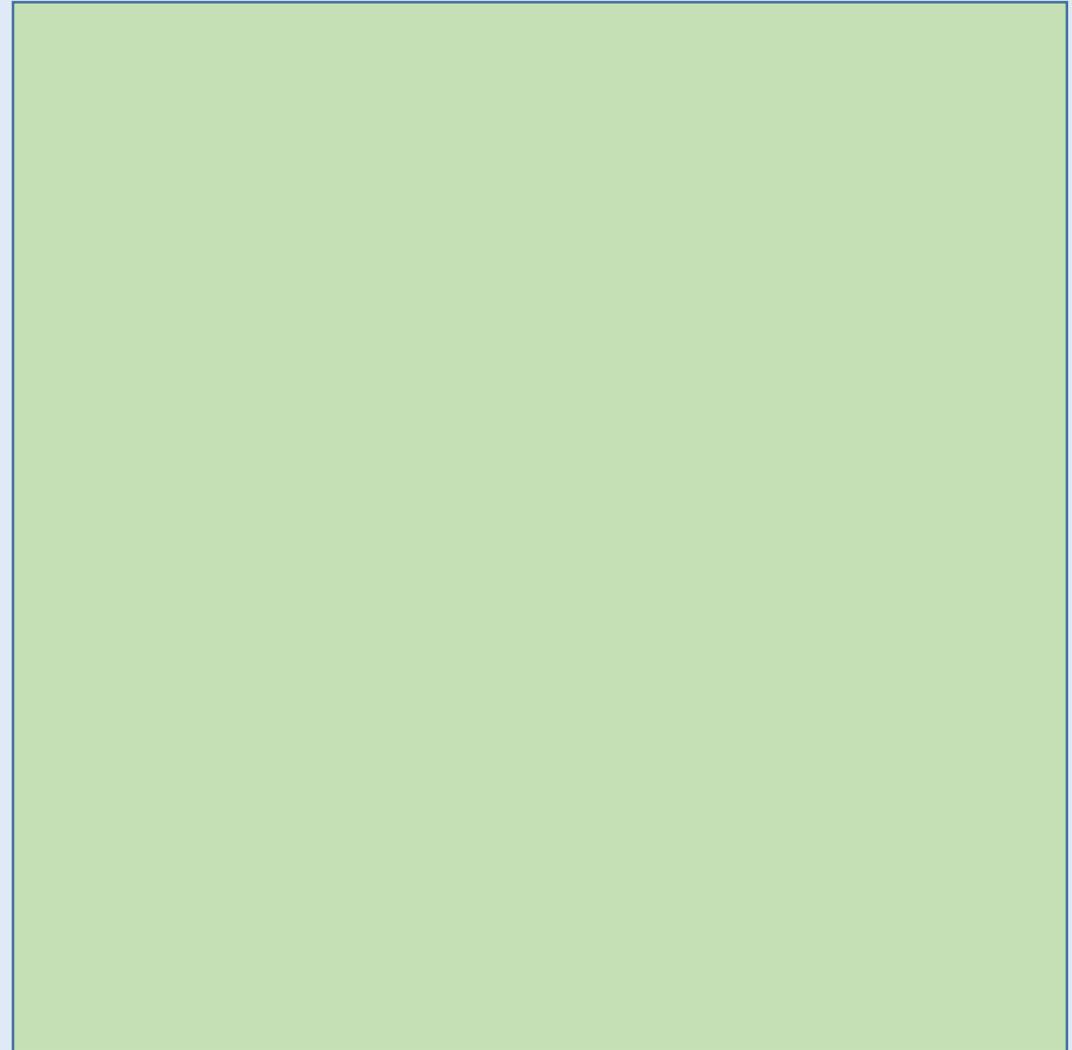


```
a=2
```

```
b=3
```

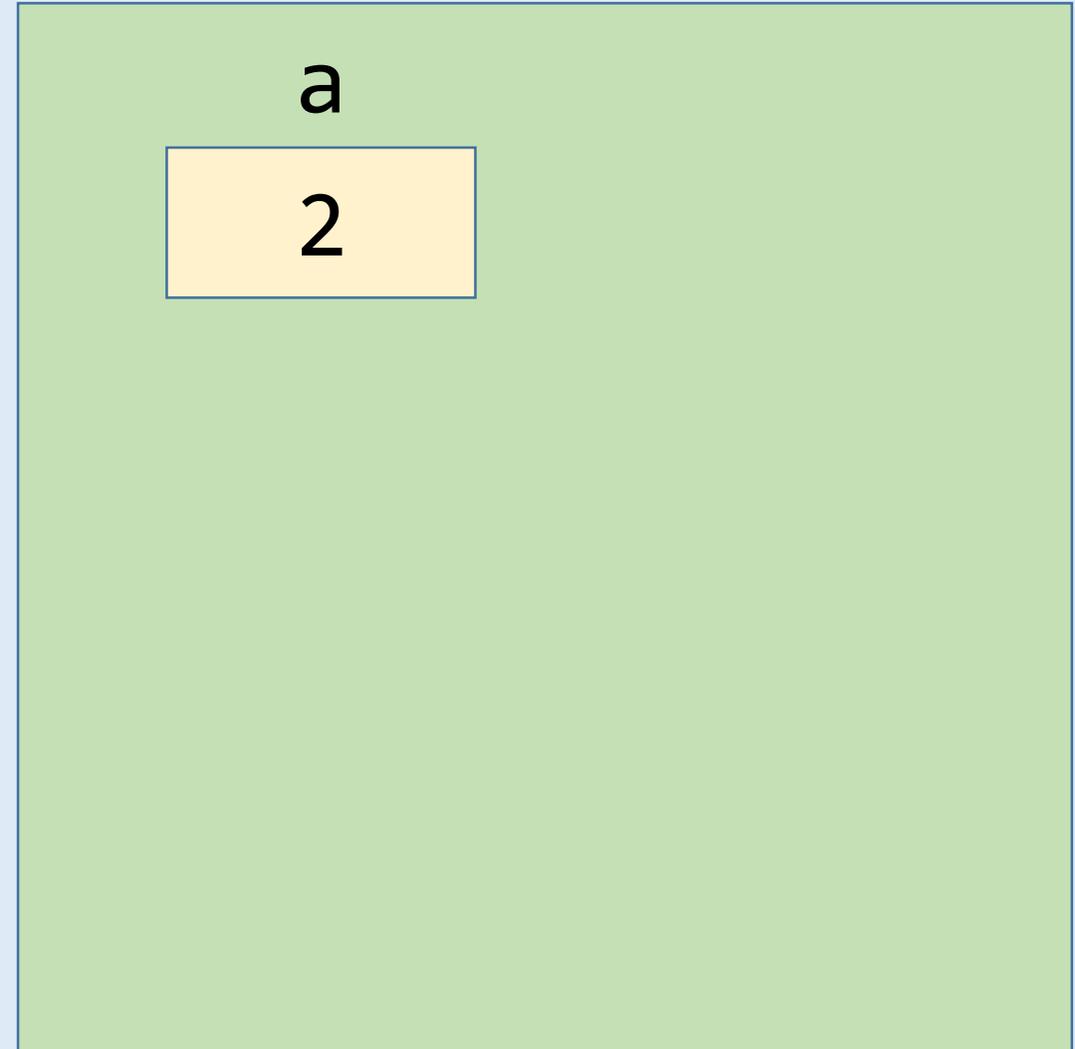
```
c=a*b+1
```

```
print(c)
```



memoria del intérprete de Python

```
a=2  
→ b=3  
c=a*b+1  
print(c)
```



memoria del intérprete de Python

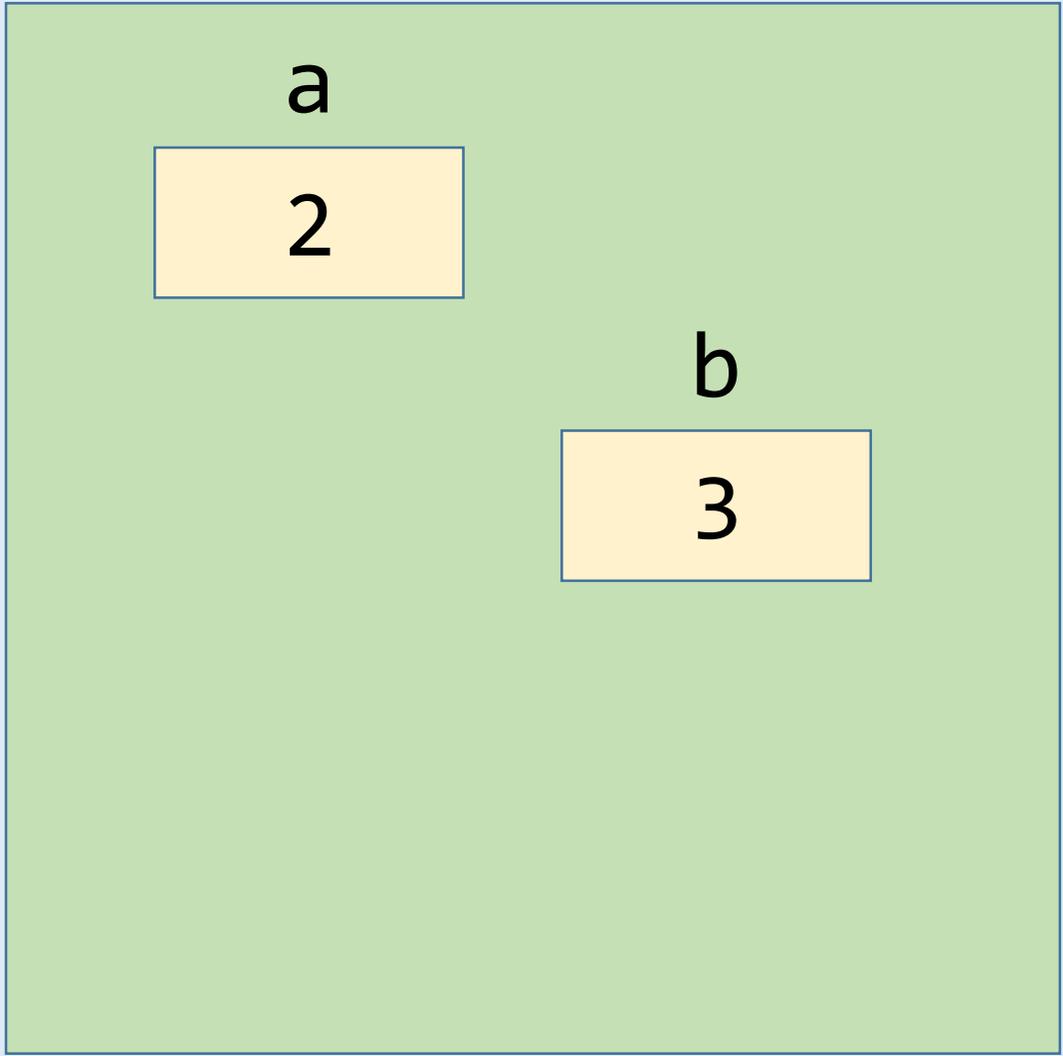
a=2

b=3



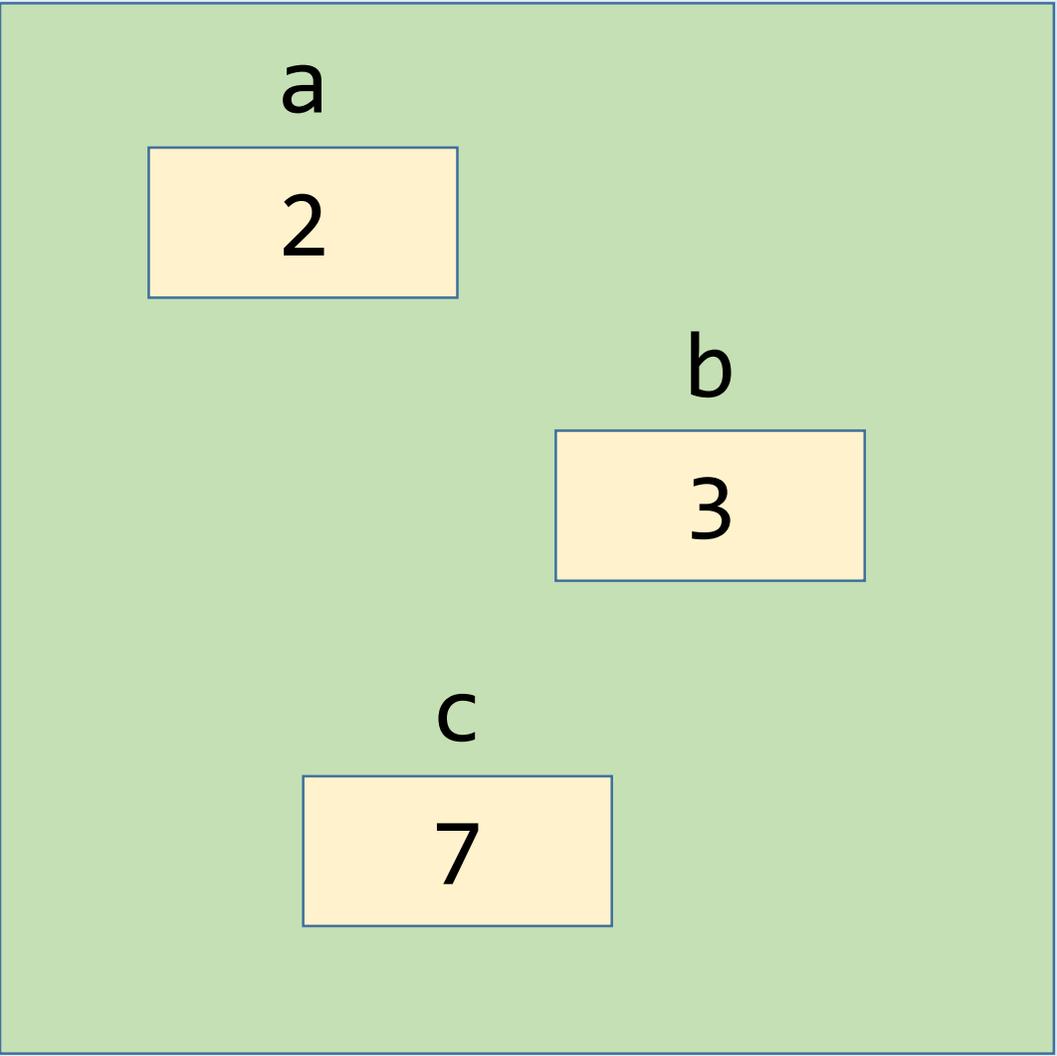
c=a\*b+1

print(c)



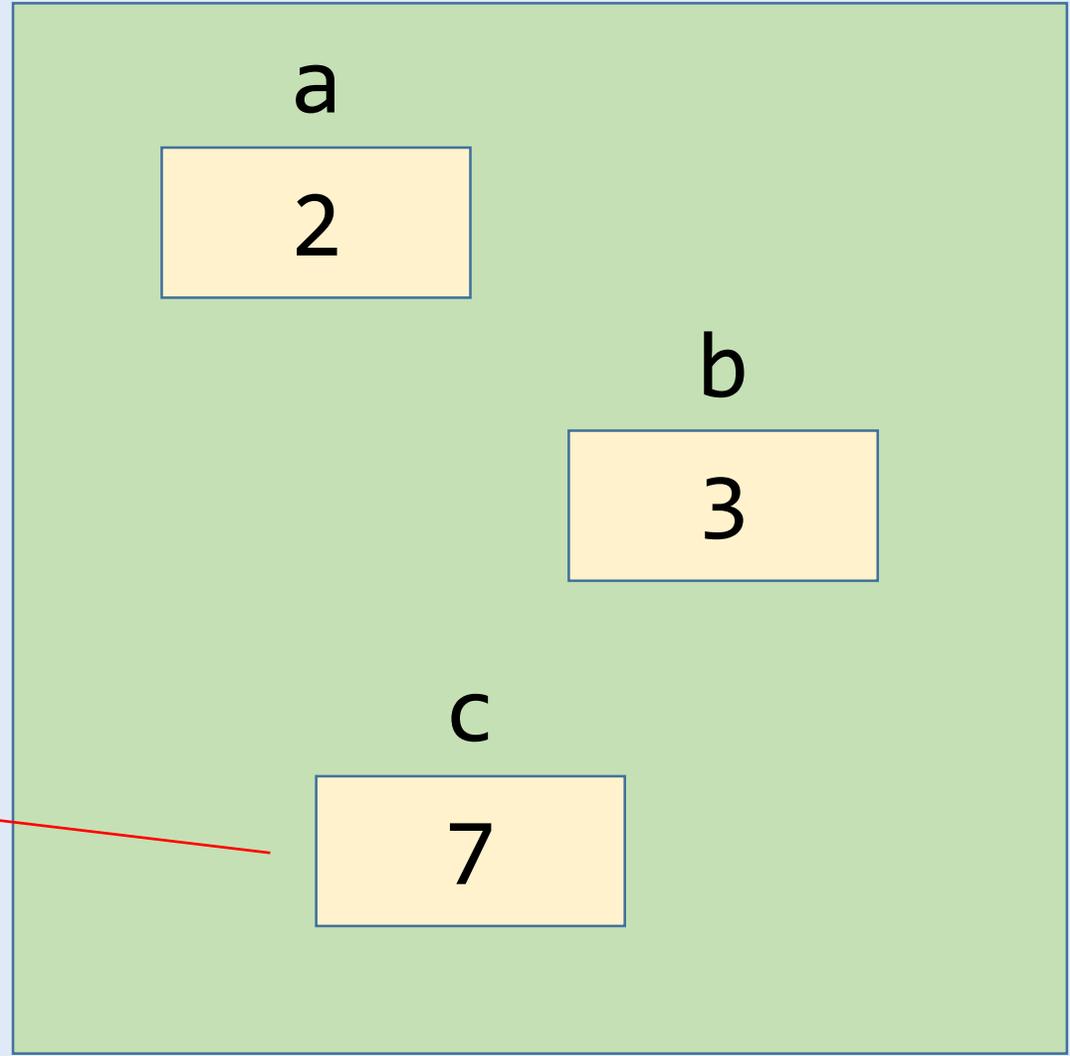
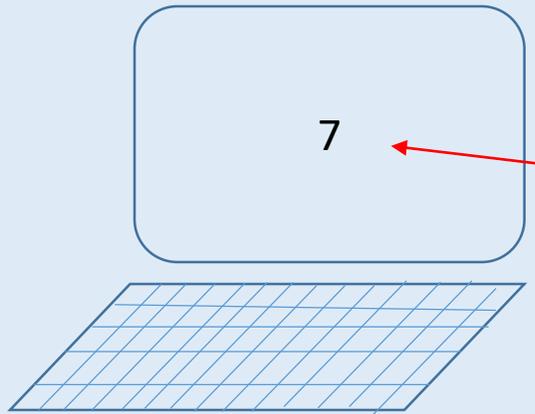
memoria del intérprete de Python

```
a=2  
b=3  
c=a*b+1  
print(c)
```



memoria del intérprete de Python

```
a=2  
b=3  
c=a*b+1  
print(c)
```



# expresiones numéricas

se evalúan de acuerdo a los siguientes criterios

paréntesis más internos

prioridad de operadores

\*\* (potencia)

\* / % // (producto, cociente, resto y cociente entero)

+ - (suma y resta)

> < >= <= == != (operadores comparativos)

de izquierda a derecha

# expresiones con cadenas de caracteres

se evalúan de acuerdo a los siguientes criterios

paréntesis más internos

prioridad de operadores

\* (repetición)

+ (concatenación)

in (pertenencia)

> < >= <= == != (operadores comparativos)

de izquierda a derecha

# expresiones booleanas (también llamadas lógicas o binarias)

se evalúan de acuerdo a los siguientes criterios

paréntesis más internos

prioridad de operadores

not (negación)

and (y lógico)

or (o lógico)

> < >= <= == != (operadores comparativos)

de izquierda a derecha

# algunas funciones importantes

`print(expr, expr, expr...)`

esta función no devuelve ningún resultado pero muestra por pantalla el valor de las expresiones entre paréntesis

`input(expr)`

muestra el texto de la expresión por pantalla y pide un dato por teclado que devuelve como una cadena de caracteres

`int(expr)`

devuelve un entero del valor de la expresión

`float(expr)`

devuelve un número real del valor de la expresión

`str(expr)`

devuelve una cadena de caracteres del valor de la expresión

## un primer programa

```
a=int(input())  
b=int(input())  
c=a+b  
print(c)
```

## el mismo programa mejorado

```
# Cálculo de la suma de dos números
#
# Escrito por José A. Corrales 13-mar-2023

# introducción de datos
numero_1 = float(input("primer número: "))
numero_2 = float(input("segundo número: "))

# cálculo de la suma
suma = numero_1 + numero_2    # se calcula la suma

# se muestra el resultado
print("la suma es", suma)
```