



Bucles e Iteración

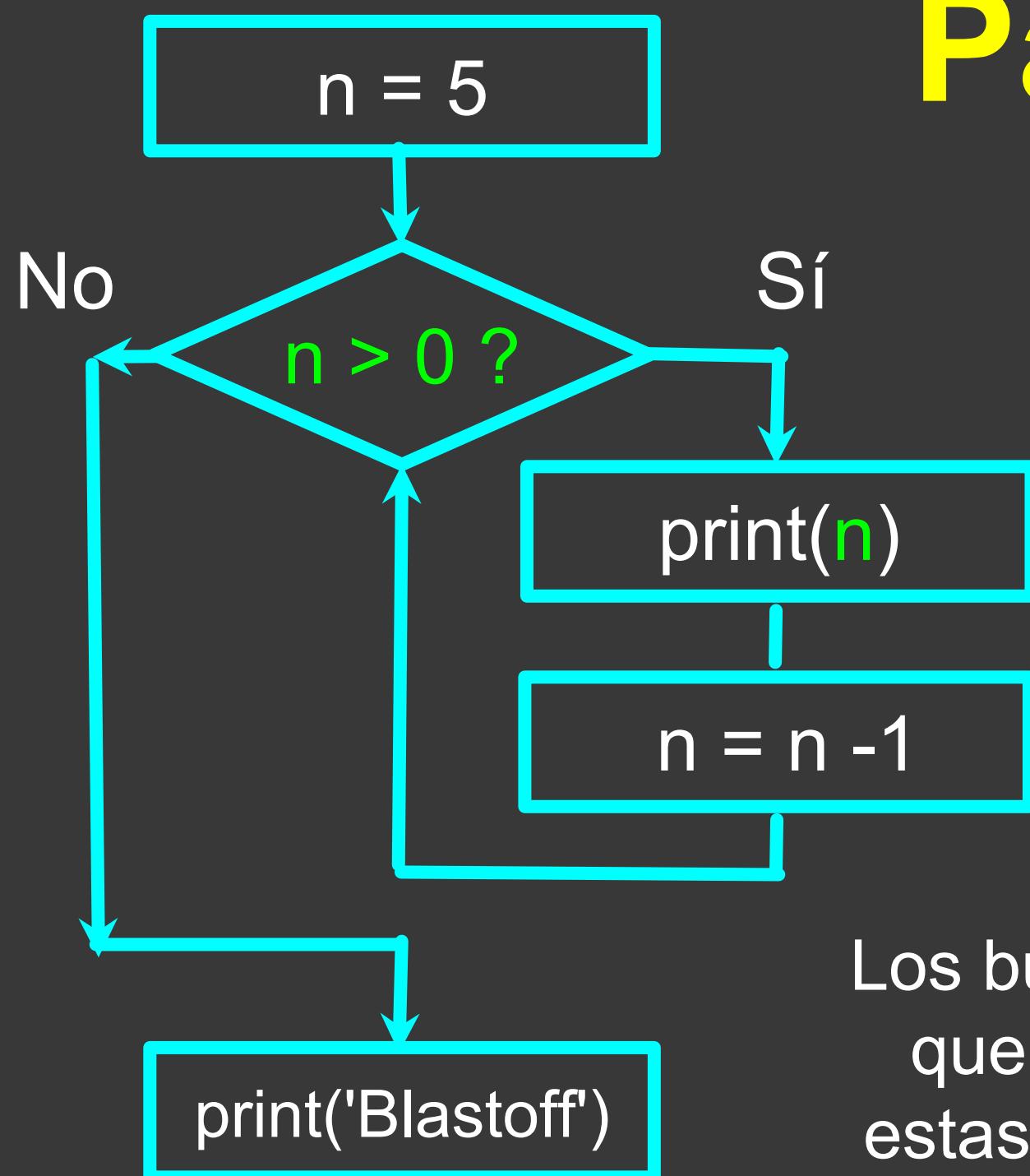
Capítulo 5



Python para Todos
www.py4e.com



Pasos Repetidos



Programa:

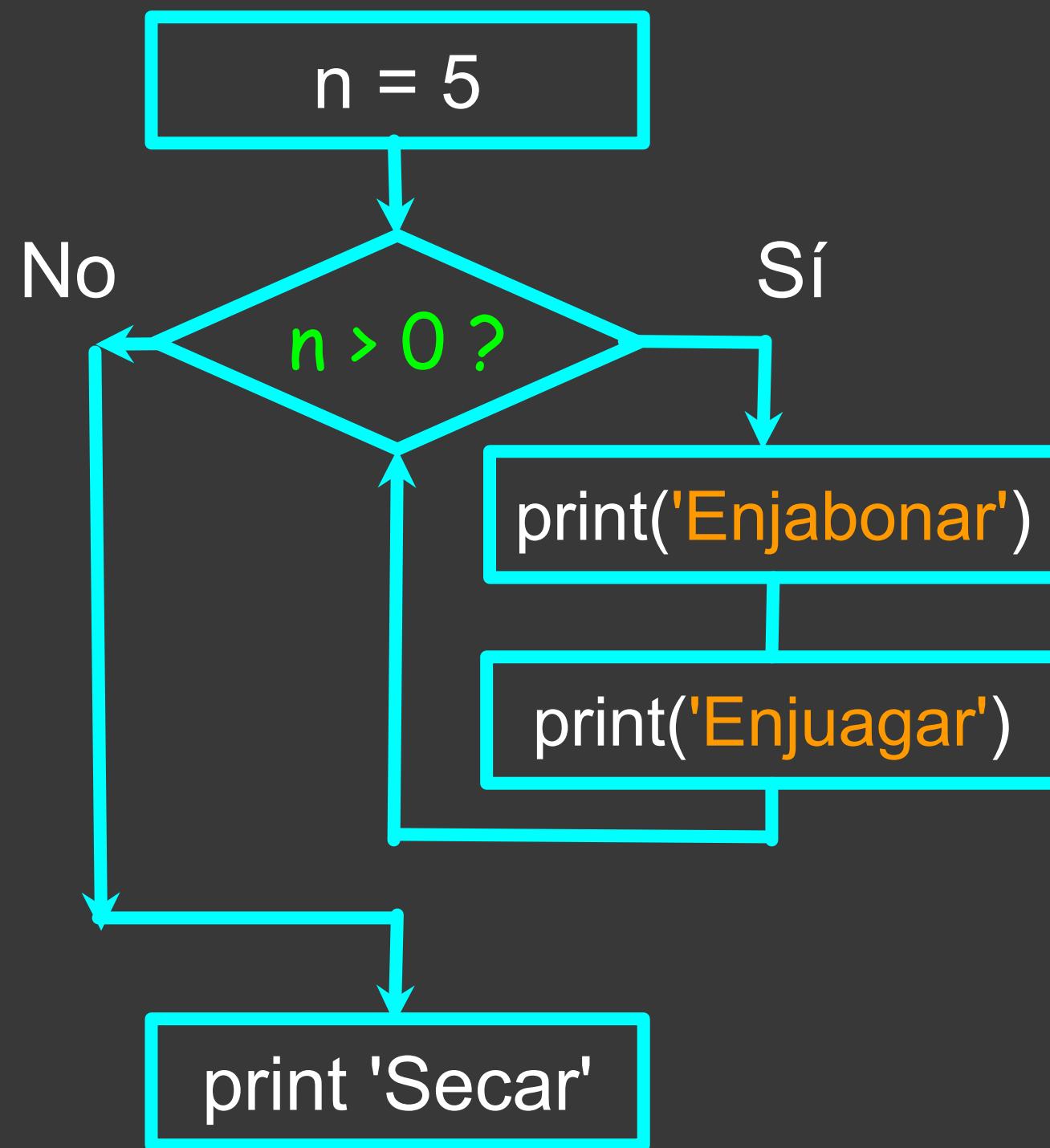
```
n = 5
while n > 0 :
    print(n)
    n = n - 1
print('Blastoff')
print(n)
```

Resultado:

```
5
4
3
2
1
¡Blastoff!
```

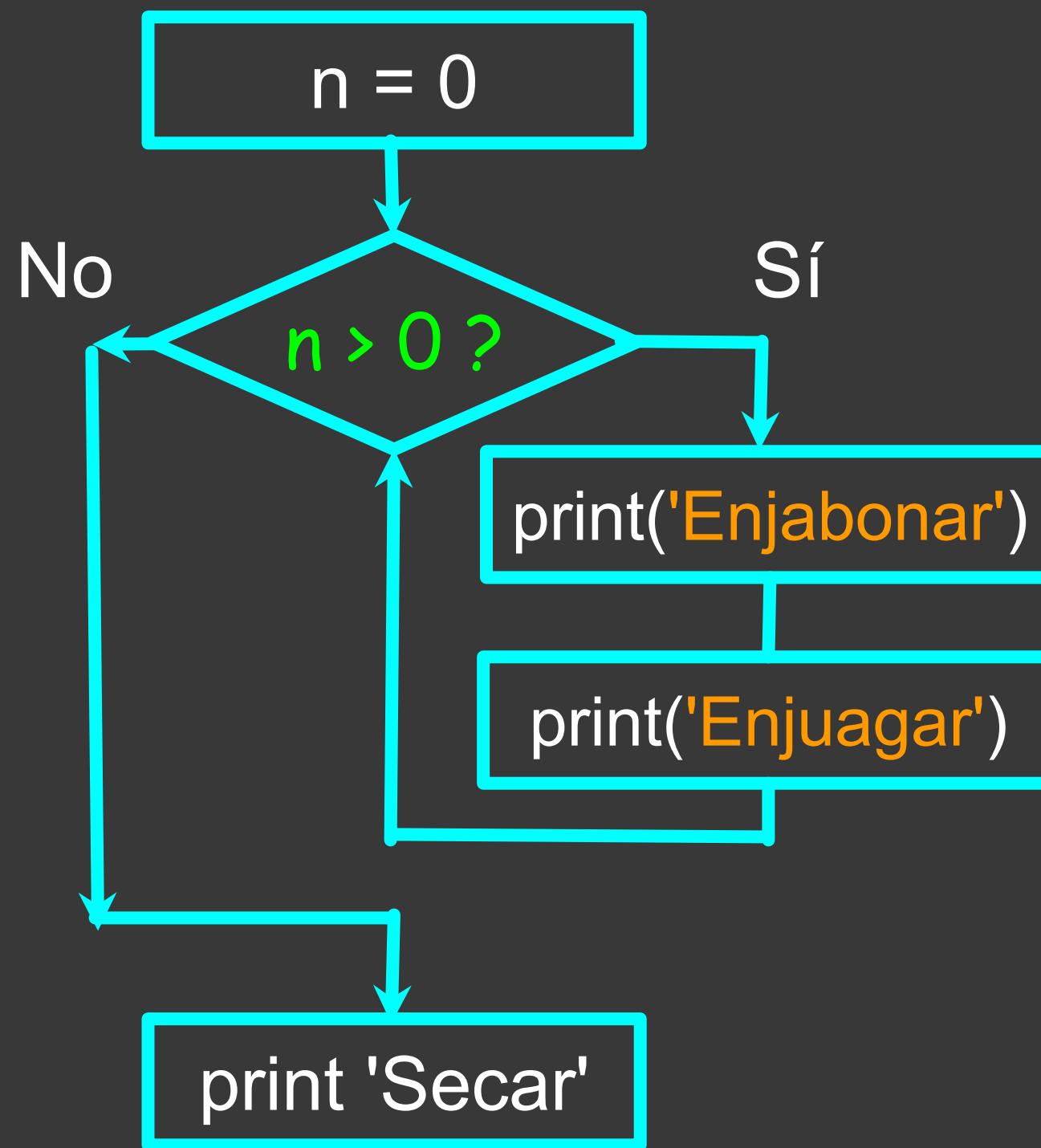
Los bucles (pasos repetidos) tienen **variables de iteración** que cambian cada vez a través del bucle. A menudo, estas **variables de iteración** atraviesan una secuencia de números.

Un Bucle Infinito



```
n = 5
while n > 0 :
    print('Enjabonar')
    print('Enjuagar')
    print('Secar')
```

¿Qué es lo que está mal en este bucle?



Otro Bucle

```
n = 0
while n > 0 :
    print('Enjabonar')
    print('Enjuagar')
    print('Secar!')
```

¿Qué es lo que está haciendo este bucle?

Romper un Bucle

- El enunciado **break** (romper) termina el bucle actual y salta al enunciado que le sigue inmediatamente al bucle
- Es como una prueba de bucle que puede suceder en cualquier lado en el cuerpo del bucle

```
while True:  
    linea = input('> ')  
    if linea == 'terminado':  
        break  
    print(linea)  
print('terminado')
```

> hola
hola
> finished
finalizado
> done
terminado

Romper un Bucle

- El enunciado **break** (romper) termina el bucle actual y salta al enunciado que le sigue inmediatamente al bucle
- Es como una prueba de bucle que puede suceder en cualquier lado en el cuerpo del bucle

```
while True:  
    linea = input('> ')  
    if linea == 'terminado'  
        :  
            break  
        print(linea)  
    print('Terminado')
```

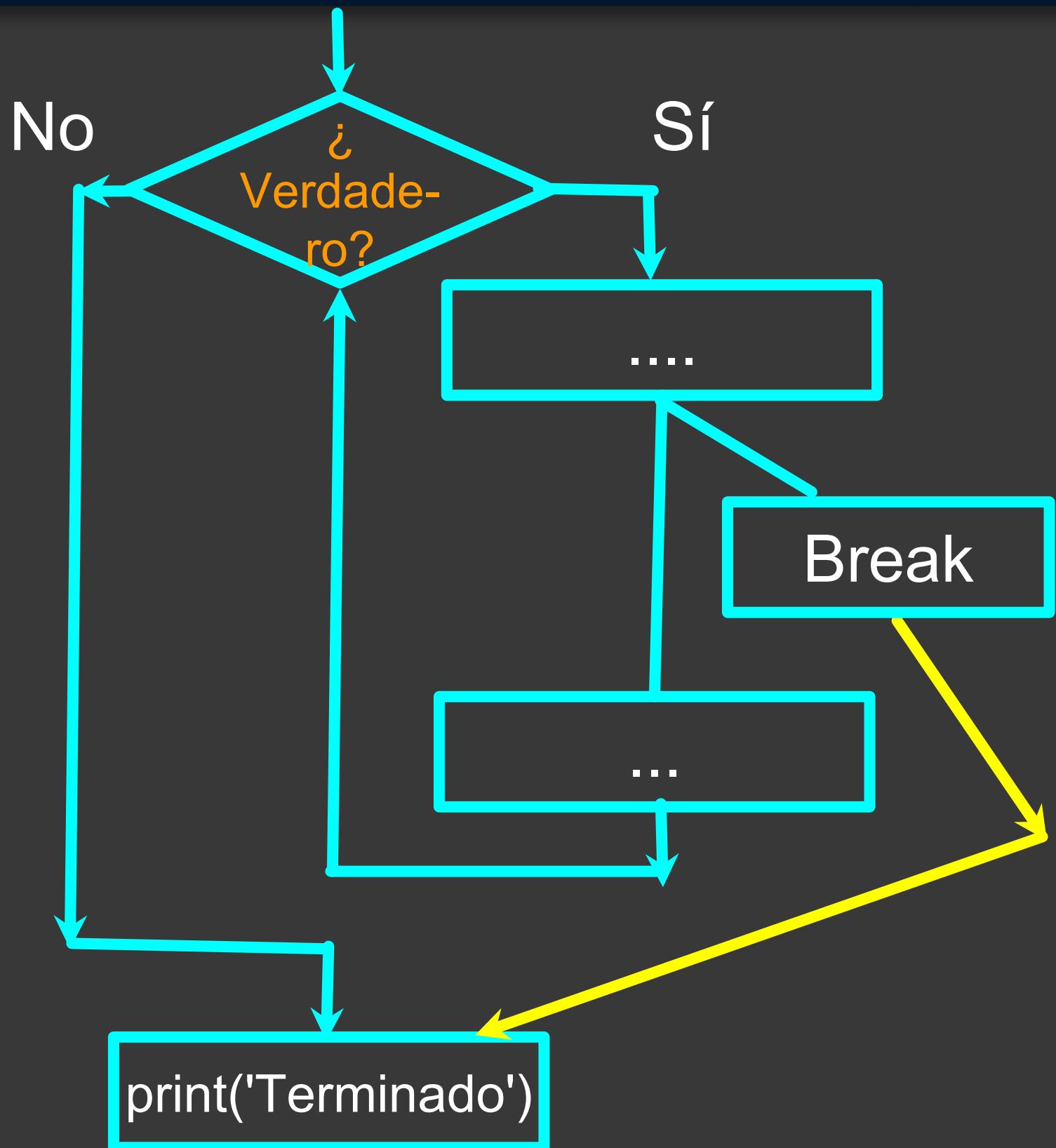
> hola
hola
> finished
finalizado
> done
terminado



```
while True:  
    linea = input('> ')  
    if linea == 'terminado'  
        :  
        break  
    print(linea)  
print('terminado')
```



[http://en.wikipedia.org/wiki/Transporter_\(Star_Trek\)](http://en.wikipedia.org/wiki/Transporter_(Star_Trek))



Finalizar una Iteración con Continue

El enunciado **continue** (continuar) termina la iteración actual y salta a la parte superior del bucle y comienza la siguiente iteración

while True:

```
    line = input('> ')
    if line[0] == '#':
        continue
    if line == 'terminado':
        :
        break
    print(line)
print('Terminado')
```

```
> hola
Hola
> # no imprimir esto
> Imprimir esto
imprimir esto
> terminado
Terminado
```

Finalizar una Iteración con Continue

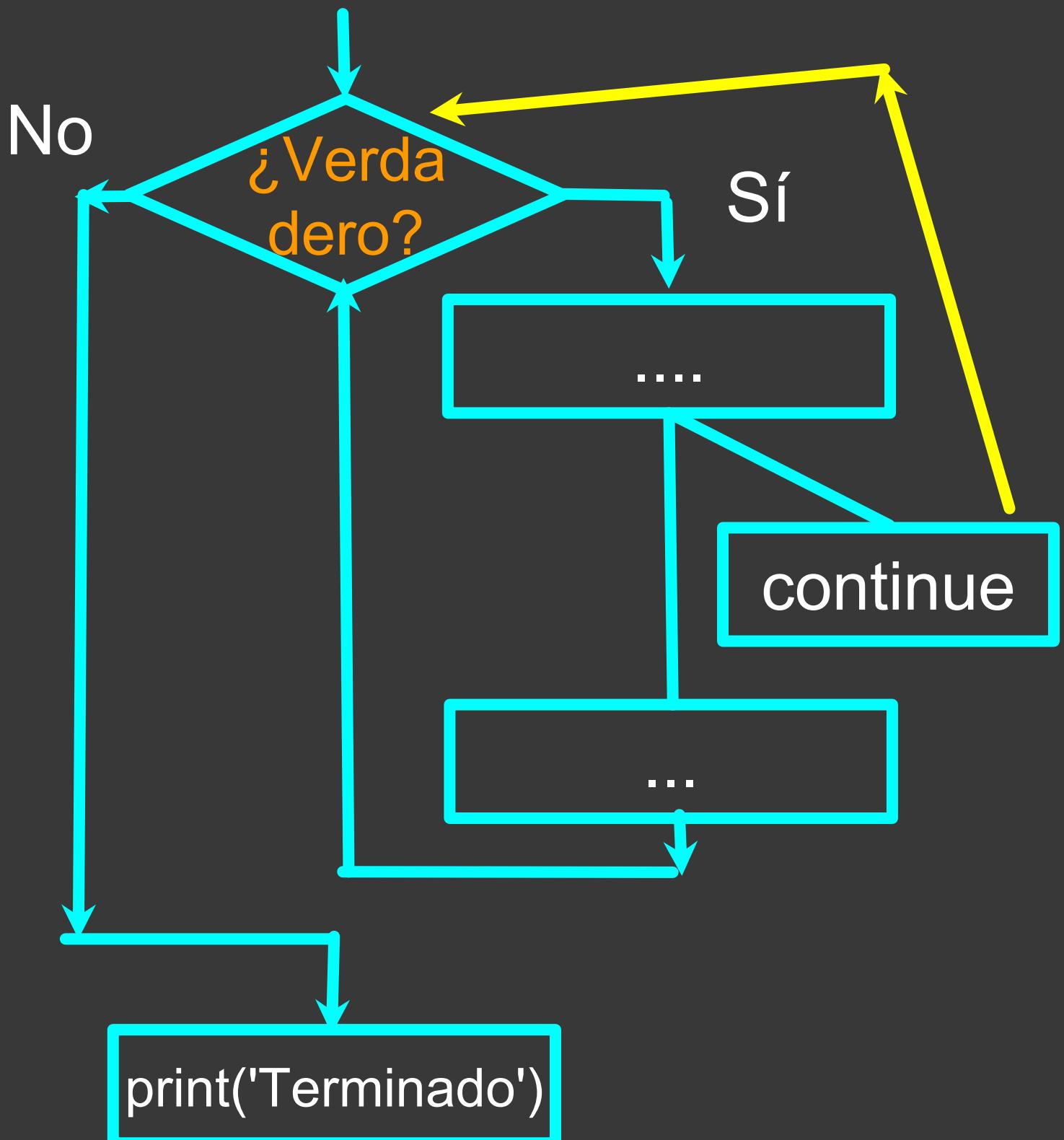
El enunciado **continue** (continuar) termina la **iteración actual** y salta a la parte superior del bucle y comienza la siguiente iteración

```
while True:  
    linea = input('> ')  
    if linea[0] == '#':  
        continue  
    if linea == 'terminado':  
        break  
    print(linea)  
print('Terminado')
```



> hola
hola
> # no imprimir esto
> Imprimir esto
imprimir esto
> terminado
Terminado

```
while True:  
    linea = raw_input('> ')  
    if linea[0] == '#':  
        continue  
    if linea == 'terminado' :  
        break  
    print(linea)  
print('Terminado')
```



Bucles Indefinidos

- Los bucles while se llaman “**bucles indefinidos**” porque continúan hasta que una condición lógica se vuelve **False (Falsa)**
- Los bucles que hemos visto hasta ahora son bastante fáciles de examinar para determinar si terminarán o si serán “bucles infinitos”
- A veces, es más difícil saber con seguridad si un bucle terminará



Bucles Definidos



Agradecimientos / Colaboraciones



Estas diapositivas están protegidas por derechos de autor 2010-Charles R. Severance (www.dr-chuck.com) de la Facultad de Información de la Universidad de Michigan y open.umich.edu, y se ponen a disposición bajo licencia de Creative Commons Attribution 4.0. Por favor, conserve esta última diapositiva en todas las copias del documento para cumplir con los requisitos de atribución de la licencia. Si realiza algún cambio, siéntase libre de agregar su nombre y el de su organización a la lista de colaboradores en esta página cuando republique los materiales.

...

Desarrollo inicial: Charles Severance, Facultad de Información de la Universidad de Michigan

... Ingrese nuevos colaboradores y traductores aquí