




Hablemos con Python

```
csev$ python3
```

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dic 5 2015, 21:12:44)  
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] en darwin. Escriba  
"ayuda ", "derechos de autor ", "créditos " o "licencia " si  
desea más información.
```

```
>>>
```



¿Qué
sigue?

```
csev$ python3
```

```
Python 3.5.1 (v3.5.1:37a07cee5969, Dic 5 2015, 21:12:44)  
[GCC 4.2.1 (Apple Inc. build 5666) (dot 3)] en darwin. Escriba  
"ayuda ", "derechos de autor ", "créditos " o "licencia " si  
desea más información.
```

```
>>> x = 1
```

```
>>> print (x)
```

```
1
```

```
>>> x = x + 1
```

```
>>> print (x)
```

```
2
```

```
>>> exit()
```

Esta es una buena prueba para asegurarse de
que ha instalado Python correctamente.
Observe que `quit()` también sirve para terminar
una sesión interactiva.

¿Qué decimos?

Elementos de Python

- **Vocabulario / Palabras** – variables y palabras reservadas (capítulo 2)
- **Estructura de la sentencia** – patrones de sintaxis válidos (capítulos 3-5)
- **Estructura de la historia** – construir un programa para un determinado propósito

```
name = input('Ingresar archivo:')  
handle = open(nombre)
```

```
conteos = dict()  
for línea in handle:  
    palabras = line.split()  
    for palabra in palabras:  
        conteos [palabra] =  
counts.get(palabra,0) + 1
```

```
bigcount = Ninguno  
bigword = Ninguna  
for palabra, conteo in counts.items():  
    if bigcount is Ninguno or conteo >  
bigcount:  
    bigword = palabra  
    bigcount = conteo  
  
print(bigword, bigcount)
```

Una “historia” breve
sobre cómo contar
palabras en un
archivo realizado en
Python

```
python words.py  
Ingresar archivo: words.txt  
hasta 16
```

Palabras Reservadas

- No puede utilizar las **palabras reservadas** como nombres o identificadores de variables

| | | | | |
|--------|--------|--------|--------|----------|
| False | class | return | is | finally |
| None | if | for | lambda | continue |
| True | def | from | while | nonlocal |
| and | del | global | not | with |
| as | elif | try | or | yield |
| assert | else | import | pass | |
| break | except | in | raise | |

Sentencias o Líneas

| | | | | | | |
|--------------------|----------------|----------------|----------------|-------------------------|--------------------------|--------------------------|
| <code>x</code> | <code>=</code> | <code>2</code> | ← | Enunciado de asignación | | |
| <code>x</code> | <code>=</code> | <code>x</code> | <code>+</code> | <code>2</code> | ← | Asignación con expresión |
| <code>print</code> | <code>(</code> | <code>x</code> | <code>)</code> | ← | Función print (imprimir) | |

Variable

Operador

Constante

Función

Programación de Párrafos

Scripts de Python

- **Interactive Python (Python interactivo) es bueno para los experimentos y programas de 3-4 líneas de largo.**
- **La mayoría de los programas son mucho más largos, entonces los escribimos en un archivo y le decimos a Python que ejecute los comandos en el archivo.**
- **De algún modo, le estamos “dando un script (guión) a Python”.**
- **Como convención, agregamos “.py” como sufijo al final de estos archivos para indicar que contienen Python.**

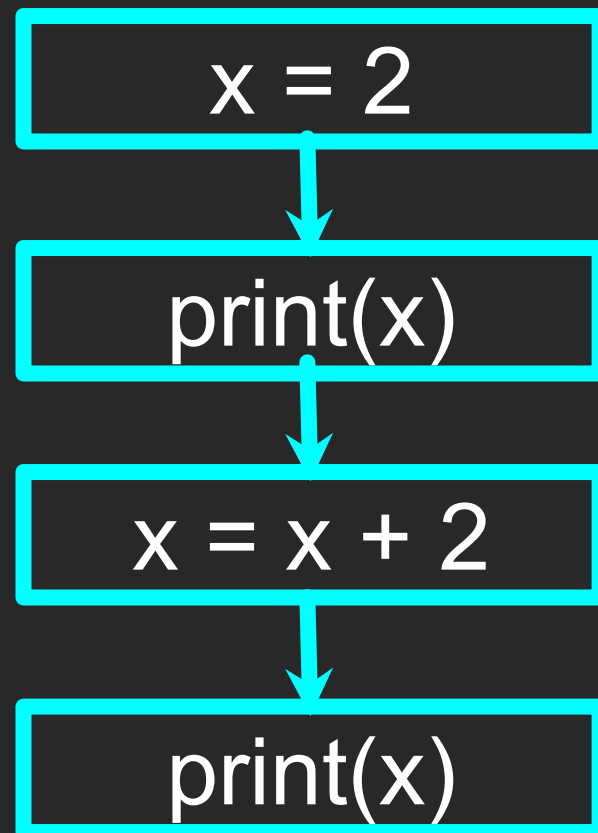
Interactivo versus Script

- Interactivo
 - Usted escribe directamente en Python de a una línea por vez y el programa responde
- Script
 - Usted ingresa una secuencia de enunciados (líneas) en un archivo utilizando un editor de texto y le dice a Python que ejecute los enunciados en el archivo

Pasos del Programa o Flujo del Programa

- Al igual que una receta o las instrucciones de instalación, un programa es una **secuencia** de pasos que se deben dar en orden.
- Algunos pasos son **condicionales**, es decir, pueden saltarse.
- A veces un paso o un grupo de pasos debe **repetirse**.
- A veces, almacenamos un conjunto de pasos para utilizar una y otra vez tal como sea necesario en distintos lugares durante el programa (Capítulo 4).

Pasos Secuenciales



Programa:

```
x = 2
print(x)
x = x + 2
print(x)
```

The diagram shows the execution of the program. The first `print(x)` statement is connected by an arrow to the value 2. The second `print(x)` statement is connected by an arrow to the value 4.

Resultado:

2
4

Cuando se está ejecutando un programa, fluye de un paso al otro. Como programadores, configuramos los “paths” (caminos) que el programa debe seguir.

Pasos Condicionales

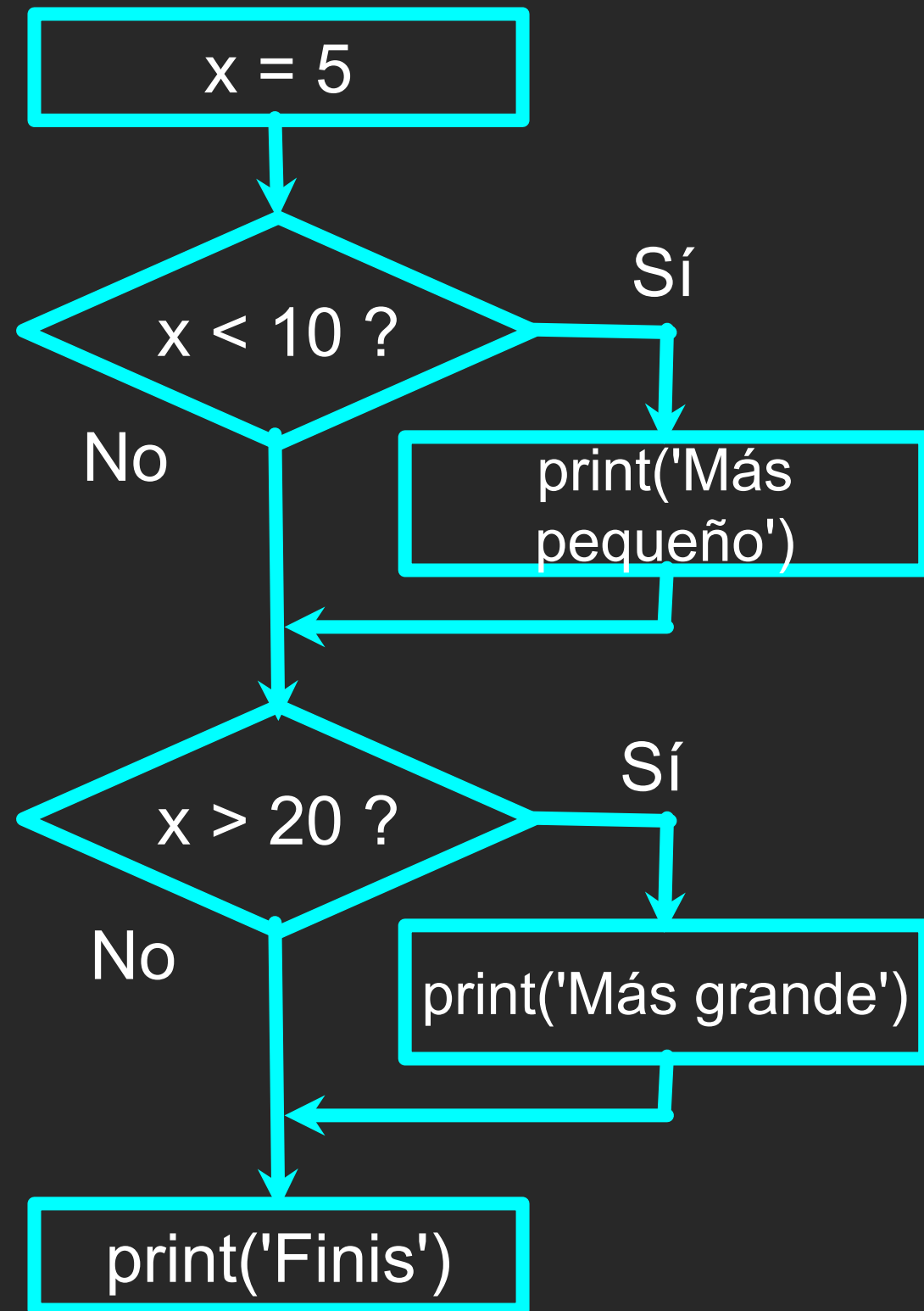
Programa:

```
x = 5
if x < 10:
    print('Más
    pequeño')
if x > 20:
    print('Más
    grande')

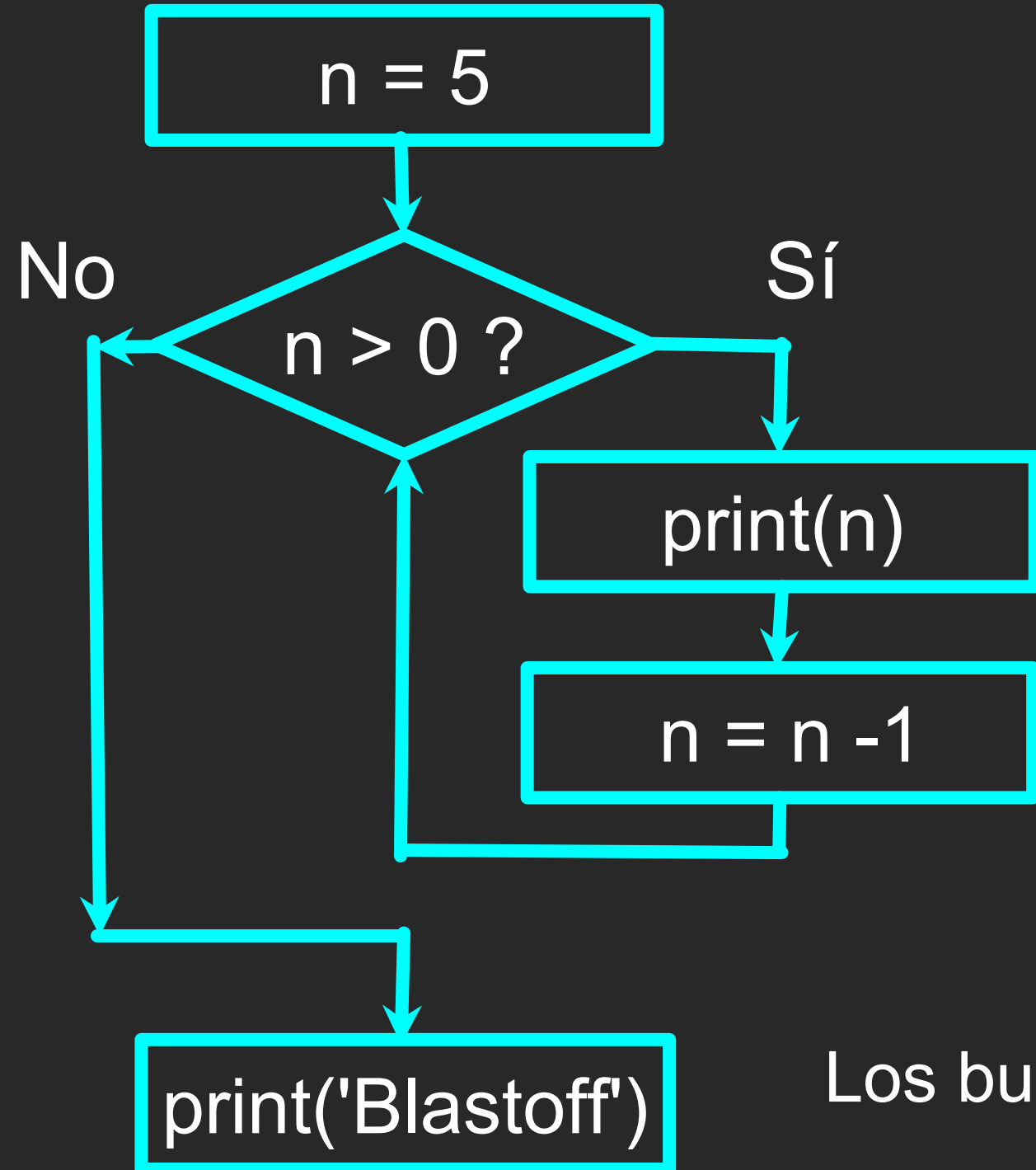
print('Finis')
```

Resultado:

Más pequeño
Finis



Pasos Repetidos



Programa:

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

Resultado:

5
4
3
2
1
¡Blastoff!

Los bucles (pasos repetidos) tienen **variables de iteración** que cambian cada vez a través del bucle.

```
name = input('Ingresar archivo:')
handle = open(nombre)

conteos = dict()
for línea in handle:
    palabras = line.split()
    for palabra in palabras:
        conteos[palabra] =
counts.get(palabra,0) + 1

bigcount = Ninguno
bigword = Ninguna
for palabra, conteo in counts.items():
    if bigcount is Ninguno or conteo >
bigcount:
    bigword = palabra
    bigcount = conteo

print(bigword, bigcount)
```

Secuencial

Repetido

Condicional


```
name = input('Ingresar archivo:')
handle = open(nombre, 'r')

conteos = dict()
for línea in handle:
    palabras = line.split()
    for palabra in palabras:
        conteos[palabra] = counts.get(palabra,0) + 1

bigcount = Ninguno
bigword = Ninguna
for palabra, conteo in counts.items():
    if bigcount is Ninguno or conteo > bigcount:
        bigword = palabra
        bigcount = conteo

print(bigword, bigcount)
```

Una breve “historia” de Python sobre cómo contar palabras en un archivo

Palabra utilizada para leer datos de un usuario

Una sentencia sobre la actualización de uno de los muchos conteos

Un párrafo sobre cómo encontrar el ítem más largo en un listado

Síntesis

- Descripción general rápida del **Capítulo 1**
- Repasaremos estos conceptos durante el curso
- Focalicémonos en el panorama general

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 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

Continúa...