

Cadenas

Capítulo 6



Python para Todos
es.py4e.com



El Tipo de dato Cadena

- Una **cadena (string)** es una secuencia de caracteres.
- Una cadena utiliza comillas 'Hola' o "Hola"
- Para las cadenas, + significa “concatenar”.
- Cuando una cadena contiene números, aún sigue siendo una cadena.
- Podemos convertir números dentro de una cadena, a enteros, utilizando **int()**

```
>>> str1 = "Hola"
>>> str2 = 'ahí'
>>> bob = str1 + str2
>>> print(bob)
Holaahí
>>> str3 = '123'
>>> str3 = str3 + 1
Traceback (most recent call
last):  File "<stdin>", line 1,
in <module>
TypeError: cannot concatenate
'str' and 'int' objects
>>> x = int(str3) + 1
>>> print(x)
124
>>>
```

Leyendo y convirtiendo datos

- Preferimos leer datos de entrada utilizando **cadenas** y después analizar y convertir los datos conforme sea necesario
- Esto nos da más control sobre situaciones de error y/o datos de entrada del usuario incorrectos
- Los números como datos de entrada deben ser **convertidos** de cadenas a enteros

```
>>> nombre = input('Ingresa:')
Ingresa:Chuck
>>> print(nombre)
Chuck
>>> manzana = input('Ingresa:')
Ingresa:100
>>> x = manzana - 10
Traceback (most recent call
last):  File "<stdin>", line 1,
in <module>
TypeError: unsupported operand
type(s) for -: 'str' and 'int'
>>> x = int(manzana) - 10
>>> print(x)
90
```



Buscando dentro de una Cadena

- Podemos obtener cualquier carácter en una cadena usando un índice especificado en **corchetes**
- El valor del índice debe ser un entero y comienza desde el cero
- El valor del índice puede ser una expresión que se ha calculado

b	a	n	a	n	a
0	1	2	3	4	5

```
>>> fruta = 'banana'
>>> letra = fruta[1]
>>> print(letra)
a
>>> x = 3
>>> w = fruta[x - 1]
>>> print(w)
n
```

Un carácter muy lejano

- Vas a obtener un **error de Python** si tratas de acceder un índice más allá del final de la cadena.
- Así que sé cuidadoso cuando construyas valores de índices y rebanadas

```
>>> zot = 'abc'
>>> print(zot[5])
Traceback (most recent call
last):  File "<stdin>", line
1, in <module>IndexError:
string index out of range
>>>
```

Las Cadenas tienen Tamaño

La función nativa `len` nos retorna el tamaño de una cadena



```
>>> fruta = 'banana'
>>> print(len(fruta))
6
```

Función len

```
>>> fruta = 'banana'
>>> x = len(fruta)
>>> print(x)
6
```

Una función es un código almacenado que utilizamos. Una función toma **datos de entrada** y produce **datos de salida**.

'banana'
(una cadena)



Función
len()



6
(un número)

Función len

```
>>> fruta = 'banana'
>>> x = len(fruta)
>>> print(x)
6
```

Una función es un código almacenado que utilizamos. Una función toma **datos de entrada** y produce **datos de salida**.

'banana'
(una cadena)



```
def len(inp):
    bla
    bla
    for x in y:
        bla
        bla
```



6
(un número)

Recorriendo una Cadena

Utilizando una sentencia **while**, una **variable de iteración**, y la función **len**, podemos construir un bucle para mirar cada una de las letras de una cadena de forma individual

```
fruta = 'banana'
indice = 0
while indice < len(fruta):
    letra = fruta[indice]
    print(indice, letra)
    indice = indice + 1
```

```
0 b
1 a
2 n
3 a
4 n
5 a
```

Recorriendo una Cadena

- Un bucle finito utilizando una sentencia **for** es mucho más elegante
- La **variable de iteración** es completamente manejada por el bucle **for**

```
fruta = 'banana'
for letra in fruta:
    print(letra)
```

b
a
n
a
n
a

Recorriendo una Cadena

- Un bucle finito utilizando una sentencia **for** es mucho más **elegante**
- La **variable de iteración** es completamente manejada por el bucle **for**

```
fruta = 'banana'
for letra in fruta :
    print(letra)
```

b
a
n
a
n
a

```
indice = 0
while indice < len(fruta) :
    letra = fruta[indice]
    print(letra)
    indice = indice + 1
```

Recorriendo una Cadena

Este es un bucle sencillo que itera a través de cada letra en una cadena y cuenta el número de veces que el bucle encuentra el carácter 'a'


```
palabra = 'banana'
contador = 0
for letra in palabra :
    if letra == 'a' :
        contador = contador + 1
print(contador)
```

Analizando **in** más a fondo

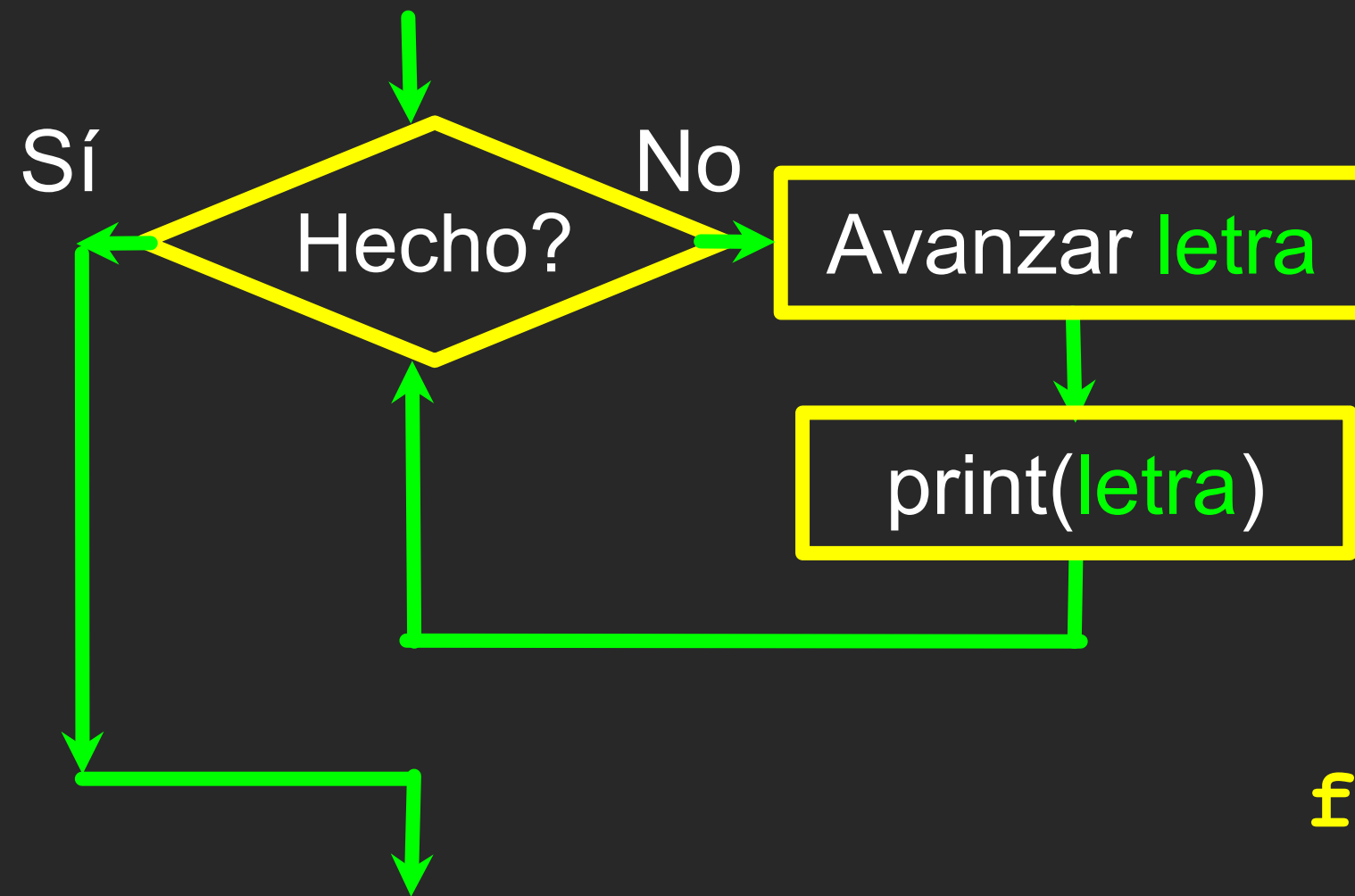
- La **variable de iteración** “itera” a través de una **secuencia** (un conjunto ordenado)
- El **bloque (cuerpo)** de código es ejecutado una vez para cada valor **en (in)** la **secuencia**
- La **variable de iteración** se mueve a través de todos los valores **en (in)** la **secuencia**

Variable de
iteración

Cadena de seis
caracteres



```
for letra in 'banana' :  
    print(letra)
```



b	a	n	a	n	a
---	---	---	---	---	---

```
for letra in 'banana' :  
    print(letra)
```

La **variable de iteración** “itera” a través de la **cadena** y el **bloque (cuerpo)** de código es ejecutado para cada valor **en (in)** la **secuencia**

Rebanado de Cadenas

- También podemos mirar a cualquier sección continua de una cadena utilizando el **operador dos puntos**
- El segundo número es un número más allá del final de la rebanada - “hasta pero no incluyendo”
- Si el segundo número está más allá del final de la cadena, entonces termina al final

M	o	n	t	y		P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11

```
>>> c = 'Monty Python'
>>> print(c[0:4])
Mont
>>> print(c[6:7])
P
>>> print(c[6:20])
Python
```

Rebanado de Cadenas

Si dejamos en blanco el primer o el último número de la rebanada, se asume que es el inicio o el final de la cadena, respectivamente

M	o	n	t	y		P	y	t	h	o	n
0	1	2	3	4	5	6	7	8	9	10	11

```
>>> c = 'Monty Python'
>>> print(c[:2])
Mo
>>> print(c[8:])
thon
>>> print(c[:])
Monty Python
```


Manipulando Cadenas..



Agradecimientos / Contribuciones



Las diapositivas están bajo el Copyright 2010- Charles R. Severance (www.dr-chuck.com) de la Escuela de Informática de la Universidad de Michigan y open.umich.edu, y están disponibles públicamente bajo una Licencia Creative Commons Attribution 4.0. Favor de mantener esta última diapositiva en todas las copias del documento para cumplir con los requerimientos de atribución de la licencia. Si haces un cambio, siéntete libre de agregar tu nombre y organización a la lista de contribuidores en esta página conforme sean republicados los materiales.

Desarrollo inicial: Charles Severance, Escuela de Informática de la Universidad de Michigan.

Traducción al Español por Juan Carlos Pérez Castellanos -
2020-04-10