Variables y Tipos de datos

Conversión entre tipos de datos
Funciones de entrada y de salida
Funciones de cadenas

Objetivos

- Utilizar sentencias de entrada y salida de datos con formato.
- Usar funciones del lenguaje para convertir tipos de datos explícitamente
- Usar funciones del lenguaje para manipular cadenas de caracteres
- Aplicar funciones para generar números aleatorios y utilizarlos en un programa

Conversión de tipos de datos

Conversión de tipos

- Cuando coloca un entero y un flotante en una expresión, el entero implícitamente es convertido a un flotante

- Usted puede controlar esto con las funciones predeterminadas de Python `int()` y `float()`

```
>>> float(99) + 100
199.0
>>> i = 42
>>> type(i)
<type 'int'>
>>> f = float(i)
>>> f
42.0
>>> type(f)
<type 'float'>
>>> 1 + 2 * float(3) + 4 - 5
6.0
>>>
```

Conversión de cadenas

- Usted también puede usar `int()` y `float()` para convertir entre cadenas y enteros
- Obtendrá un error si la cadena no contiene caracteres numéricos

```
>>> snum = '123'
>>> type(snum)
<type 'str'>
>>> snum + 1
TypeError: cannot concatenate 'str' and  'int'
>>> ival = int(snum)
>>> type(ival)
<type 'int'>
>>> ival + 1
124
>>> msj = 'hello bob'
>>> niv = int(msj)
ValueError: invalid literal for int()
```

Entrada y salida de datos

Entrada y Salida de Usuario

- Podemos instruir a Python que haga una pausa y lea datos directamente del usuario a través de la función `input()`
- La función `input()` retorna una cadena
- Véase en el uso de la función `print()` que para concatenar cadenas se utiliza el operador `+`

prog03.py

```
name = input('Hola, ¿cómo te llamas?: ')
print('Bienvenido ' + name + ' !')
```

Salida de prog03.py

```
Hola, ¿cómo te llamas?: Jorge
Bienvenido Jorge !
```

Convertir una Entrada de Usuario

- Si deseamos pedir un número al usuario, debemos convertirlo de cadena a entero o flotante, usando una función de conversión de tipo
- Más adelante veremos qué hacer con datos mal ingresados por el usuario

prog04.py

```
num1 = int(input('Enter #1: '))
num2 = int(input('Enter #2: '))
suma = num1 + num2
print('La suma es', num1 + num2)
```

Salida de prog04.py

```
Enter #1: 32
Enter #2: 10
La suma es 42
```

Cadenas de caracteres

Secuencia de caracteres

Indexación

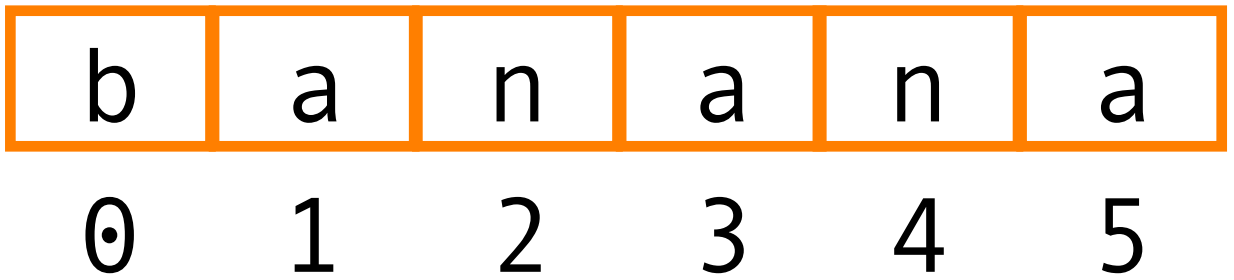
Cadenas

- Una cadena es una secuencia de caracteres. Una secuencia utiliza comillas `'Hello'` o `"Hello"`
- En las secuencias, + significa «concatenar»
- Cuando una cadena contiene dígitos, sigue siendo una cadena
- Los números de una cadena se pueden convertir en un número con la función `int()`

```
>>> str1 = "Hola"
>>> str2 = 'Mundo'
>>> bob = str1 + str2
>>> print(bob)
HolaMundo
>>> str3 = '123'
>>> str3 = str3 + 1
TypeError: cannot concatenate 'str' and 'int'
>>> x = int(str3) + 1
>>> print x
124
>>>
```

Manejo de cadenas

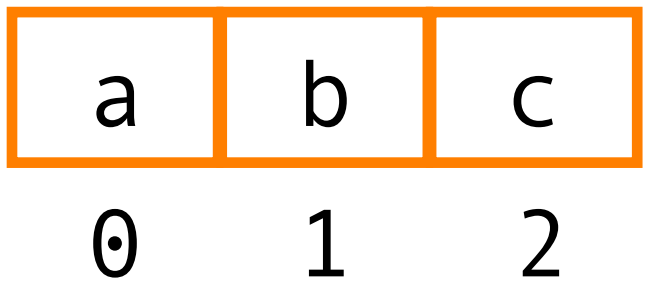
- Se puede acceder a cualquier carácter de una cadena usando un índice especificado entre corchetes
- El valor del índice debe ser un número entero y empezar por cero.
- El valor del índice puede ser una expresión.




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

Un caracter de más

- Si se intenta indexar más allá del último carácter de una secuencia, se obtendrá un error
- Por tanto, hay que tener cuidado al construir los valores del índice y sus partes




```
>>> frase = 'abc'
>>> print(frase[5])
IndexError: string index out of range
>>>
```

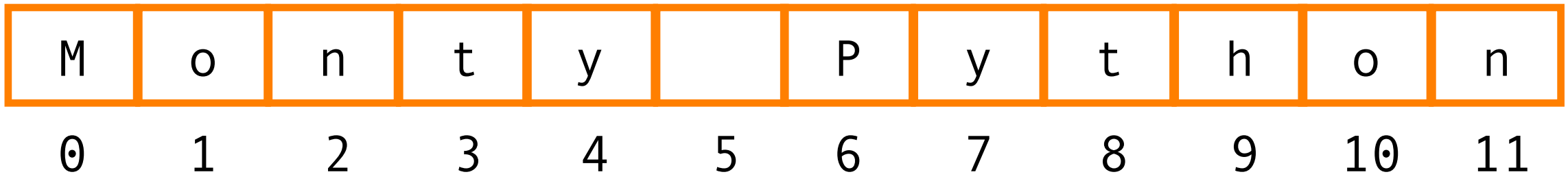
Operaciones con cadenas de caracteres

Longitud

- Hay una función `len()` interna de Python que muestra la longitud de una cadena


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

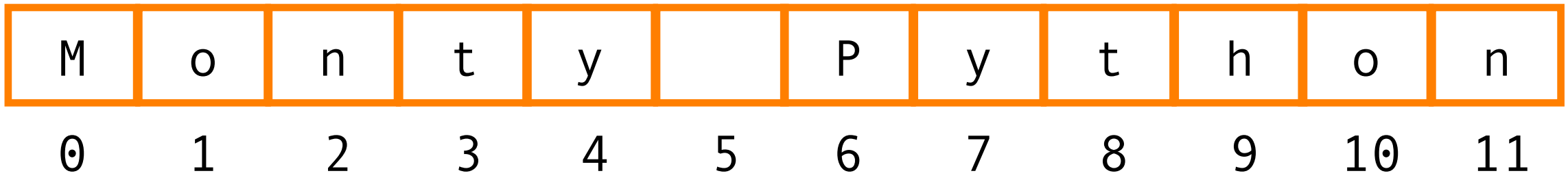
Recorte de cadenas



- También se puede observar cualquier sección continúa de una cadena usando dos puntos
- El segundo número es el último de la sección, pero sin incluirlo
- Si el segundo es mayor al último, se detiene al final

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

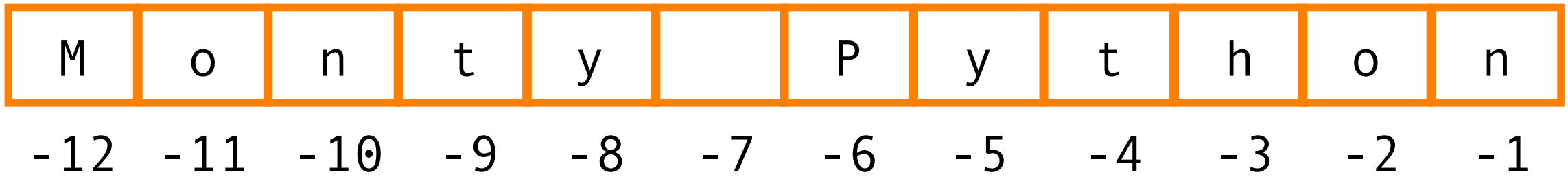
Recorte de cadenas



- Si se omite el primer o el segundo número de la sección, se asume que es el principio o el final de la cadena respectivamente
- El recorte siempre es de izquierda a derecha y no de derecha a izquierda

```
>>> s = 'Monty Python'
>>> print( s[:2] )
Mo
>>> print( s[8:] )
Thon
>>> print( s[:] )
Monty Python
```

Recorte de cadenas



- Es posible especificar índices negativos que indican posiciones a partir del final de la cadena
- Observe que el primer índice negativo es -1
- Para dar la vuelta a la cadena es posible especificar un tercer elemento en el rango -1

```
>>> s = 'Monty Python'
>>> print( s[-5:] )
ython
>>> print( s[-6:-1] )
Pytho
>>> print( s[::-1] )
nohtyP ytnoM
```

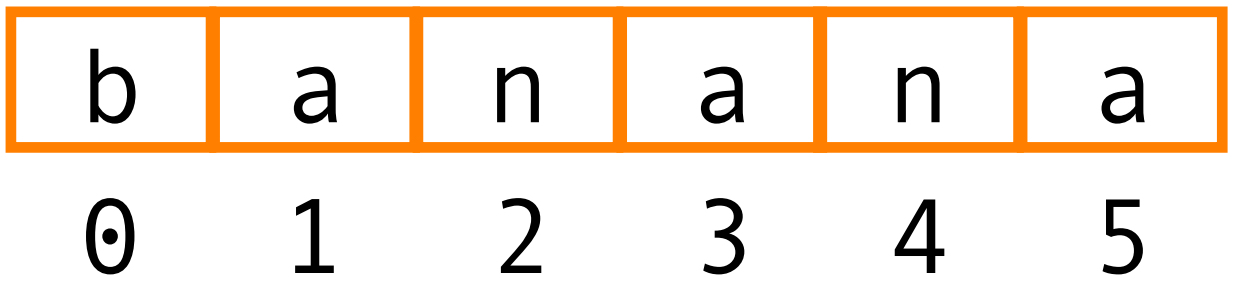
Concatenar cadena

- Aplicado a las secuencias en general, el operador + significa concatenación

```
>>> a = 'Hola'
>>> b = a + 'Mundo'
>>> print b
HolaMundo
>>> c = a + '  ' + 'a todos'
>>> print c
Hola a todos
>>>
```

Buscar una subcadena

- La función `find()` se usa para buscar una subcadena dentro de una cadena
- `find()` encuentra la primera coincidencia de una subcadena
- Si no se encuentra, `find()` devuelve -1
- Hay que recordar que la primera posición de la cadena es cero




```
>>> fruta = 'banana'
>>> pos = fruta.find('na')
>>> print(pos)
2
>>> aa = fruta.find('z')
>>> print(aa)
-1
```

Formatos de salida

Dando formato a los números

Python tiene muchos métodos para dar formato a números. El formato se refiere a la forma de presentarlos.

Es útil, cuando se requiere recortar decimales o que estos tengan el mismo ancho de dígitos

(f-strings) le permiten incluir el valor de las expresiones dentro de una cadena prefijando la cadena con f y escribiendo expresiones como {expresion}.:

```
>>> a = 10
>>> b = 3
>>> a / b
3.3333333333333335
>>> f'{a/b:.2f}'
'3.33'
>>> f'{a/b:6.2f}'
'  3.33'
>>> f'{a/b:06.2f}'
'003.33'
```

Uso de .format()

prog04.py

```
pi = 3.1415
print('Pi es igual a ' + str(pi))
#Ejemplo con format:
print('Pi es igual a {}'.format(pi))
a, b = 10, 3
#Ejemplo con format:
print('Pi={:.2f}, a/b={:6.2f}'.format(pi, a/b))
```

Output

```
Pi es igual a 3.1415
Pi es igual a 3.1415
Pi=3.14, a/b=  3.33
```