

Introducción al Lenguaje de Programación PYTHON

Leopoldo González-Santos

Instituto de Neurobiología
UNAM - México

August 8, 2017

Contenido

1 Números

- Números

2 Cadenas de caracteres

- Cadenas de caracteres

3 Listas

- Listas

4 Operadores en Python

- Operadores lógicos
- Operadores relacionales
- Operadores aritméticos
- Operadores a nivel de Bit

5 Estructuras de Control

- La sentencia if
- La sentencia for

Números

```
>>> ancho = 20
>>> largo = 5*9
>>> ancho * largo
900
>>> 3 * 3.75 / 1.5
7.5
>>> 7.0 / 2
3.5
>>> 1j * complex(0,1)
(-1+0j)
>>> 3+1j*3
(3+3j)
>>> (3+1j)*3
(9+3j)
>>> (1+2j)/(1+1j)
(1.5+0.5j)
```

Números

```
>>> a=1.5+0.5j
```

```
>>> a.real
```

```
1.5
```

```
>>> a.imag
```

```
0.5
```

```
>>> a=3.0+4.0j
```

```
>>> abs(a)
```

```
5.0
```

Números Enteros

```
>>> impuesto = 12.5 / 100
```

```
>>> precio = 100.50
```

```
>>> precio * impuesto
```

```
12.5625
```

```
>>> precio + _
```

```
113.0625
```

```
>>> round(_, 2)
```

```
113.06
```

Contenido

- 1 **Números**
 - Números
- 2 **Cadenas de caracteres**
 - Cadenas de caracteres
- 3 **Listas**
 - Listas
- 4 **Operadores en Python**
 - Operadores lógicos
 - Operadores relacionales
 - Operadores aritméticos
 - Operadores a nivel de Bit
- 5 **Estructuras de Control**
 - La sentencia if
 - La sentencia for

Cadenas de caracteres

```
>>> 'huevos y pan'
```

```
'huevos y pan'
```

```
>>> "doesn't"
```

```
"doesn't"
```

```
>>> '''Si," le dijo.'
```

```
'''Si," le dijo.'
```

Cadenas de caracteres

```
>>> hola = "Esta es una larga cadena que contiene \n \  
varias líneas de texto, tal y como se hace en C. \n \  
Notar que los espacios en blanco al principio de la línea \  
son significantes."
```

```
>>> print hola
```

```
>>> palabra = 'Ayuda' + 'A'
```

```
>>> palabra
```

```
'AyudaA'
```

```
>>> '<' + palabra*5 + '>'
```

```
'<AyudaAAyudaAAyudaAAyudaAAyudaA>'
```


Cadenas de caracteres

```
>>> palabra[4]
```

```
'a'
```

```
>>> palabra[0:2]
```

```
'Ay'
```

```
>>> palabra[2:4]
```

```
'ud'
```

```
>>> palabra[:2]
```

```
'Ay'
```

```
>>> palabra[2:]
```

```
'udaA'
```

Cadenas de caracteres

```
>>> 'x' + palabra[1:]  
'xyudaA'  
>>> 'Mas' + palabra[5]  
'MasA'  
>>> palabra[-1]  
'A'  
>>> palabra[-2]  
'a'  
>>> palabra[-2:]  
'aA'  
>>> palabra[:-2]  
'Ayud'
```

Contenido

- 1 Números
 - Números
- 2 Cadenas de caracteres
 - Cadenas de caracteres
- 3 Listas
 - Listas
- 4 Operadores en Python
 - Operadores lógicos
 - Operadores relacionales
 - Operadores aritméticos
 - Operadores a nivel de Bit
- 5 Estructuras de Control
 - La sentencia if
 - La sentencia for

Listas

```
>>> a = ['pan', 'huevos', 100, 1234]
>>> a
['pan', 'huevos', 100, 1234]
>>> a[0]
'pan'
>>> a[3]
1234
>>> a[-2]
100
>>> a[1:-1]
['huevos', 100]
>>> a[:2] + ['carne', 2*2]
['pan', 'huevos', 'carne', 4]
>>> 3*a[:3] + ['Boo!']
['pan', 'huevos', 100, 'pan', 'huevos', 100, 'pan', 'huevos', 100, 'Boo!']
```

Listas

```
>>> a
['pan', 'huevos', 100, 1234]
>>> a[2] = a[2] + 23
>>> a
['pan', 'huevos', 123, 1234]
>>> a[0:2] = [1, 12]
>>> a
[1, 12, 123, 1234]
>>> a[0:2] = []
>>> a
[123, 1234]
>>> a[1:1] = ['bruja', 'xyzzzy']
>>> a
[123, 'bruja', 'xyzzzy', 1234]
>>> a[:0] = a
>>> a
[123, 'bruja', 'xyzzzy', 1234, 123, 'bruja', 'xyzzzy', 1234]
>>> a[1:] = []
```

Contenido

- 1 Números
 - Números
- 2 Cadenas de caracteres
 - Cadenas de caracteres
- 3 Listas
 - Listas
- 4 Operadores en Python
 - Operadores lógicos
 - Operadores relacionales
 - Operadores aritméticos
 - Operadores a nivel de Bit
- 5 Estructuras de Control
 - La sentencia if
 - La sentencia for

Operadores lógicos

Operador	Descripción	Ejemplo
and	se cumple a y b?	<code>r=True and False = r es False</code>
or	se cumple a o b?	<code>r=True or False = r es True</code>
not	No a	<code>r=not True = r es False</code>

Operadores relacionales

Operador	Descripción	Ejemplo
<code>==</code>	Son iguales a y b?	<code>r=5 == 3 = r</code> es False
<code>!=</code>	Son distintos a y b?	<code>r=5 != 3 = r</code> es True
<code><</code>	Es a menor que b?	<code>r=5 < 3 = r</code> es False
<code>></code>	Es a mayor que b?	<code>r=5 > 3 = r</code> es True
<code><=</code>	Es a menor o igual que b?	<code>r=5 <= 5 = r</code> es True
<code>>=</code>	Es a mayor o igual que b?	<code>r=5 >= 3 = r</code> es True

Operadores aritméticos

Operador	Descripción	Ejemplo
+	Suma	$r=3+2 = r$ es 5
-	Resta	$r=4-7 = r$ es -3
-	Negación	$r=-7 = r$ es -7
*	Multiplicación	$r=2*6 = r$ es 12
**	Exponente	$r=2**6 = r$ es 64
/	División	$r=3.5/2 = r$ es 1.75
//	División Entera	$r=3.5 // 2 = r$ es 1.0
%	Módulo	$r=7 \% 2 = r$ es 1

Operadores a nivel de Bit

Operador	Descripción	Ejemplo
&	and	$r=3 \& 2 = r \text{ es } 2$
	or	$r=3 2 = r \text{ es } 3$
.	xor	$r=3 . 2 = r \text{ es } 1$
.	not	$r=.3 = r \text{ es } -4$
<<	Desplazamiento a la izquierda	$r=3 << 1 = r \text{ es } 6$
>>	Desplazamiento a la derecha	$r=3 >> 1 = r \text{ es } 1$

Contenido

- 1 Números
 - Números
- 2 Cadenas de caracteres
 - Cadenas de caracteres
- 3 Listas
 - Listas
- 4 Operadores en Python
 - Operadores lógicos
 - Operadores relacionales
 - Operadores aritméticos
 - Operadores a nivel de Bit
- 5 Estructuras de Control
 - La sentencia if
 - La sentencia for

La sentencia **if**

```
>>> x = int(raw_input("Ingresa un entero, por favor: "))
Ingresa un entero, por favor: 42
>>> if x < 0:
...     x = 0
...     print 'Negativo cambiado a cero'
... elif x == 0:
...     print 'Cero'
... elif x == 1:
...     print 'Simple'
... else:
...     print 'Mas'
...
'Mas'
```

La sentencia **for**

```
>>> a = ['gato', 'ventana', 'defenestrado']
>>> for x in a:
...     print x, len(x)
...
gato 4
ventana 7
defenestrado 12
```