

# Concatenando Listas Usando +

Podemos crear una nueva lista al juntar dos listas creadas previamente

```
>>> a = [1, 2, 3]
>>> b = [4, 5, 6]
>>> c = a + b
>>> print(c)
[1, 2, 3, 4, 5, 6]
>>> print(a)
[1, 2, 3]
```

# Las Listas Pueden Ser Rebanadas Usando :

```
>>> t = [9, 41, 12, 3, 74, 15]
>>> t[1:3]
[41, 12]
>>> t[:4]
[9, 41, 12, 3]
>>> t[3:]
[3, 74, 15]
>>> t[:]
[9, 41, 12, 3, 74, 15]
```

Recuerda: Tal como en las cadenas, el segundo número es “hasta pero no incluyendo”

# Métodos de Listas

```
>>> x = list()
>>> type(x)
<type 'list'>
>>> dir(x)
['append', 'count', 'extend', 'index', 'insert',
'pop', 'remove', 'reverse', 'sort']
>>>
```

<http://docs.python.org/tutorial/datastructures.html>

# Construyendo una Lista desde Cero

- Podemos crear una **lista** vacía y después agregar elementos usando el método **append** (agregar)
- La **lista** mantiene su orden y los nuevos elementos son **agregados** al final de la **lista**

```
>>> cosas = list()
>>> cosas.append('libro')
>>> cosas.append(99)
>>> print(cosas)
['libro', 99]
>>> cosas.append('galleta')
>>> print(cosas)
['libro', 99, 'galleta']
```

# ¿Se Encuentra Algo en una Lista?

- Python provee dos **operadores** que te permiten revisar si un elemento se encuentra en una lista
- Son operadores lógicos retornan **True** o **False**
- Esos operadores no modifican la lista

```
>>> algunos = [1, 9, 21, 10, 16]
>>> 9 in algunos
True
>>> 15 in algunos
False
>>> 20 not in algunos
True
>>>
```

# Las Listas están en Orden

- Una **lista** puede almacenar muchos elementos y los mantiene en orden hasta que hagamos algo para cambiarlo
- Una **lista** puede ser **ordenada** (i.e., cambiar su orden)
- El método **sort (ordenar)** (al contrario que en las cadenas) significa “**ordénalo tu mismo**”

```
>>> amigos = [ 'Joseph', 'Glenn', 'Sally' ]
>>> amigos.sort()
>>> print(amigos)
['Glenn', 'Joseph', 'Sally']
>>> print(amigos[1])
Joseph
>>>
```

# Funciones Nativas y Listas

- Hay un cierto número de **funciones** nativas en **Python** que toman **listas** como parámetros
- ¿Recuerdas los bucles que hicimos? Son mucho más sencillos

```
>>> nums = [3, 41, 12, 9, 74, 15]
>>> print(len(nums))
6
>>> print(max(nums))
74
>>> print(min(nums))
3
>>> print(sum(nums))
154
>>> print(sum(nums)/len(nums))
25.6
```

```
total = 0
contador = 0
while True :
    inp = input('Ingresa un número: ')
    if inp == 'hecho' : break
    valor = float(inp)
    total = total + valor
    contador = contador + 1
```

```
promedio = total / contador
print('Promedio:', promedio)
```

Ingresa un número: 3

Ingresa un número: 9

Ingresa un número: 5

Ingresa un número: hecho

Promedio: 5.6666666666667

```
numlista = list()
while True :
    inp = input('Ingresa un número: ')
    if inp == 'hecho' : break
    valor = float(inp)
    numlista.append(valor)

promedio = sum(numlista) / len(numlista)
print('Promedio:', promedio)
```