

# Listas

## Capítulo 8

Python para Todos  
[es.py4e.com](http://es.py4e.com)



# Programación

## Algoritmos

- Un conjunto de reglas o pasos usados para resolver un problema

## Estructuras de Datos

- Una forma particular de organizar datos en una computadora

<https://es.wikipedia.org/wiki/Algoritmo>

[https://es.wikipedia.org/wiki/Estructura\\_de\\_datos](https://es.wikipedia.org/wiki/Estructura_de_datos)

# ¿Qué No es Una “Colección”?

La mayoría de nuestras **variables** solo tienen un valor en ellas – cuando ponemos un nuevo valor en una **variable**, el valor anterior es sobrescrito

```
$ python
>>> x = 2
>>> x = 4
>>> print(x)
4
```

# Una Lista Es un Tipo de Colección



- Una **colección** nos permite poner muchos valores en una sola “**variable**”
- Una **colección** es buena porque podemos mover **muchos valores** alrededor en un paquete conveniente

```
amigos = [ 'Joseph', 'Glenn', 'Sally' ]
```

```
equipaje = ['calcetin', 'camisa', 'perfume' ]
```

# Constantes en Listas

- Las constantes de una **Lista** están encerradas por corchetes y los elementos en la lista están separados por comas
- Un elemento de una **lista** puede ser cualquier objeto de Python – incluso **otra lista**
- Una **lista** puede estar vacía

```
>>> print([1, 24, 76])
[1, 24, 76]
>>> print(['rojo', 'amarillo',
'azul'])
['rojo', 'amarillo', 'azul']
>>> print(['rojo', 24, 98.6])
['rojo', 24, 98.6]
>>> print([ 1, [5, 6], 7])
[1, [5, 6], 7]
>>> print([])
[]
```

# ¡Ya estamos usando Listas!

```
for i in [5, 4, 3, 2, 1] :  
    print(i)  
print(' ¡Despegue! ')
```

5

4

3

2

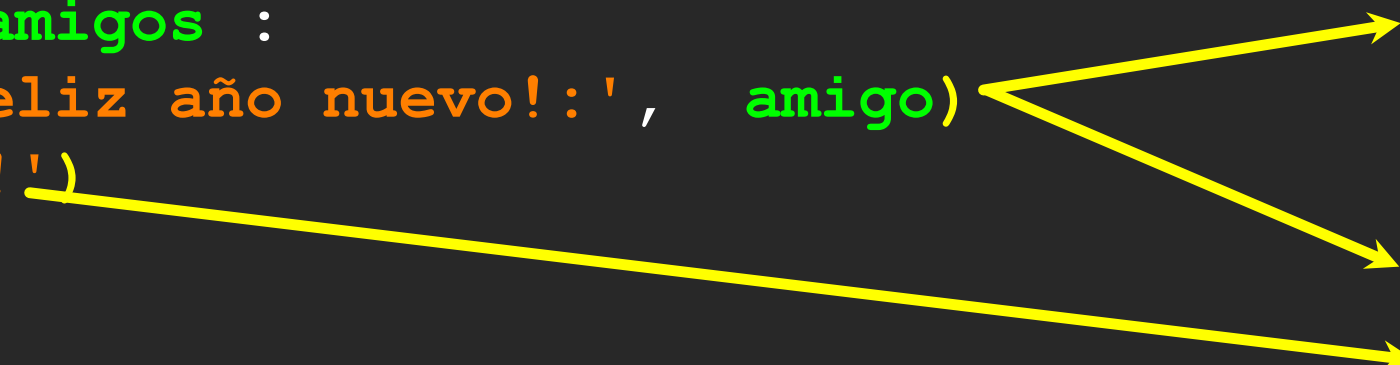
1

¡Despegue!

# Listas y Bucles Definidos – Mejores

## Amigos

```
amigos = ['Joseph', 'Glenn', 'Sally']  
for amigo in amigos :  
    print('¡Feliz año nuevo!', amigo)  
print('¡Hecho!')
```



¡Feliz Año Nuevo!: Joseph  
¡Feliz Año Nuevo!: Glenn  
¡Feliz Año Nuevo!: Sally  
¡Hecho!

```
z = ['Joseph', 'Glenn', 'Sally']  
for x in z:  
    print('¡Feliz año nuevo!', x)  
print('¡Hecho!')
```

# Búsqueda dentro de Listas

Así como en las cadenas, podemos obtener cualquier elemento individual de una lista utilizando un índice especificado en **corchetes**



Joseph	Glenn	Sally
0	1	2

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



# Las Listas son Mutables

- Las Cadenas son “**inmutables**” – no podemos cambiar el contenido de una cadena – tenemos que crear una **nueva cadena** para haer cualquier cambio
- Las Listas son “**mutables**” – podemos **cambiar** un elemento de una lista utilizando el operador índice

```
>>> fruta = 'Banana'
>>> fruta[0] = 'b'
Traceback
TypeError: 'str' object does not
support item assignment
>>> x = fruta.lower()
>>> print(x)
banana
>>> loto = [2, 14, 26, 41, 63]
>>> print(loto)
[2, 14, 26, 41, 63]
>>> loto[2] = 28
>>> print(loto)
[2, 14, 28, 41, 63]
```

# ¿Qué tan Larga es una Lista?

- La función `len()` toma una `lista` como parámetro y retorna el número de `elementos` en la `lista`
- De hecho, `len()` nos da el número de elementos de cualquier conjunto o secuencia (tal como una cadena...)

```
>>> saludo = 'Hola Bob'
>>> print(len(saludo))
9
>>> x = [ 1, 2, 'joe', 99]
>>> print(len(x))
4
>>>
```

# Usando la Función Rango

- La función `range` retorna una lista de números que van desde cero hasta el número anterior al parámetro
- Podemos construir un bucle por índices usando un `for` y un entero `iterador`

```
>>> print(range(4))
[0, 1, 2, 3]
>>> amigos = ['Joseph', 'Glenn', 'Sally']
>>> print(len(amigos))
3
>>> print(range(len(amigos)))
[0, 1, 2]
>>>
```

# Un Cuento Sobre Dos Bucles...

```
amigos = ['Joseph', 'Glenn', 'Sally']

for amigo in amigos :
    print('Feliz año nuevo:', amigo)

for i in range(len(amigos)) :
    amigo = amigos[i]
    print('Feliz año nuevo:', amigo)
```

```
>>> amigos = ['Joseph', 'Glenn', 'Sally']
>>> print(len(amigos))
3
>>> print(range(len(amigos)))
[0, 1, 2]
>>>
```

Feliz año nuevo: Joseph  
Feliz año nuevo: Glenn  
Feliz año nuevo: Sally