



ESCUELA DE INGENIERÍA

Listas y operaciones



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

Cristian Ruz Ruz

Profesor Asistente Adjunto, Departamento de Ciencia de la Computación
Facultad de Ingeniería – Pontificia Universidad Católica de Chile

Índice



Concepto de lista



Estructura **list**



Operaciones sobre listas

Introducción

- Conocer la estructura de datos lista.
- Crear y manipular listas en Python.
- Qué operaciones se pueden aplicar sobre listas.



Concepto de lista

¿Qué es una lista?

Colección de datos

Estructura de datos

Estructura **ordenada**

Acceso **eficiente** a cualquier
posición





Estructura list

Listas en Python:

list

1

Creación de una lista vacía

```
lista = list()
```

2

Mostrar el contenido

```
print(lista)
```

```
[]
```

3

Tipo de dato

```
print(type(lista))
```

```
<class 'list'>
```


Listas en Python:

list()

4

Creación de una lista con elementos

```
profesores = ['Jaime', 'Cristian', 'Denis']  
lista_vacia = []  
mezcla = ['Chile', 8, 4.15, 'Abril']  
sublista = [15, ['Arica', 9], 'Febrero']
```

```
print(profesores)  
print(lista_vacia)  
print(mezcla)  
print(sublista)
```

```
['Jaime', 'Cristian', 'Denis']  
[]  
['Chile', 8, 4.15, 'Abril']  
[15, ['Arica', 9], 'Febrero']
```

Creación por concatenación

```
otoño = ['Abril', 'Mayo', 'Junio']  
invierno = ['Julio', 'Agosto']  
frío = otoño + invierno  
más_frío = invierno * 2
```

```
print(frío)  
print(más_frío)
```

```
['Abril', 'Mayo', 'Junio', 'Julio', 'Agosto']  
['Julio', 'Agosto', 'Julio', 'Agosto']
```

Listas en Python:
`list()`



Operaciones sobre listas

Consultas sobre una lista

<i>índices</i> →	0	1	2	3	4	5	6	7
mezcla = ['Chile', 8, 4.15, 'Abril', 3, 5, '2020', 'MOOC']								

Longitud de una lista

```
largo = len(mezcla)
print(largo)
```

8

Elemento en posición *i*

```
print(mezcla[0])
print(mezcla[4])
print(mezcla[7])
print(mezcla[-2])
```

Chile
3
MOOC
2020

Extrayendo porciones (*slicing*)

```
print(mezcla[2:6])
print(mezcla[:3])
print(mezcla[5:])
print(mezcla[1:6:2])
```

[4.15, 'Abril', 3, 5]
['Chile', 8, 4.15]
[5, '2020', 'MOOC']
[8, 'Abril', 5]

Modificar elementos de una lista

`lista[i] = ...`

```
mezcla = ['Chile', 8, 4.15, 'Abril', 3, 5, '2020', 'MOOC']
```

1

Modificar elementos en la posición *i*

```
mezcla[3] = 'Mayo'  
mezcla[4] += 5  
print(mezcla)
```

```
['Chile', 8, 4.15, 'Mayo', 8, 5, '2020', 'MOOC']
```

2

Modificar rangos de elementos

```
mezcla[2:5] = [-1, 0, -2]  
print(mezcla)
```

```
['Chile', 8, -1, 0, -2, 5, '2020', 'MOOC']
```

```
mezcla[1:6] = ['X']  
print(mezcla)
```

```
['Chile', 'X', '2020', 'MOOC']
```

Recorriendo listas

`for _ in lista:`

```
mezcla = ['Chile', 8, 4.15, 'Abril', 3, 5, '2020', 'MOOC']
```

1

Recorriendo con `for _ in range:`

```
for i in range(len(mezcla)):
    print(mezcla[i])
```

- Recorrido usando un rango de índices.
- Útil para buscar elementos y recordar su posición.

2

Recorriendo con `for _ in lista:`

```
for elemento in mezcla:
    print(elemento)
```

- Recorrido por elemento.
- Fácil para observar cada elemento si no importa la posición.

Operaciones sobre listas: agregar y sacar desde el final

1

Agregar al final: append

```
opciones = ['Si', 'No']  
opciones.append('No sé')  
print(opciones)
```

```
['Si', 'No', 'No sé']
```

2

Eliminar desde el final: pop

```
cuando = ['Mañana', 'Pasado', 'Hoy']  
eliminado = cuando.pop()  
print(cuando)  
print(eliminado)
```

```
['Mañana', 'Pasado']  
Hoy
```

Operaciones sobre listas: insertar y eliminar desde cualquier posición

1

Insertar en posición *i*: insert

```
opciones = ['Si', 'No', 'No sé']  
opciones.insert(2, 'Quizás')  
print(opciones)
```

```
['Si', 'No', 'Quizás', 'No sé']
```

2

Eliminar elemento: pop

```
cuando = ['Mañana', 'Pasado', 'Hoy']  
eliminado = cuando.pop(1)  
print(cuando)  
print(eliminado)
```

```
['Mañana', 'Hoy']  
Pasado
```


Operaciones sobre listas: encontrar y eliminar un elemento

1

Encontrar posición de elemento: `index`

```
opciones = ['Si', 'No', 'Quizás', 'No sé']  
print(opciones.index('Quizás'))  
print(opciones)
```

2

2

Eliminar elemento: `remove`

```
cuando = ['Mañana', 'Pasado', 'Hoy']  
cuando.remove('Pasado')  
print(cuando)
```

['Mañana', 'Hoy']

Síntesis

- Concepto de lista.
- Implementación en Python: `list`.
- Consultas: longitud, índices, *slicing*.
- Recorrido: `for ... in`.
- Operaciones: `append`, `insert`, `pop`, `index`, `remove`.

Referencias bibliográficas

- Data Structures. The Python Tutorial, v3.8.2rc1. Recuperado de: <https://docs.python.org/3/tutorial/datastructures.html>
- Imagen diapositiva 6: "La lista" por My Buffo licencia bajo CC BY-SA 2.0
- Imagen diapositiva 6: Recuperado de: inspireexpressmiami en Pixabay
- Imagen diapositiva 6: Recuperado de: <https://www.flickr.com/photos/duncan/6618989131/> by duncan c is licensed under CC BY-NC 2.0



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE