# PYTHON - ESTRUCTURAS DE DATOS

# Listas

### Construcción de Listas

In [ ]: Primos = [2, 3, 5, 7, 11, 13, 17, 19]

In [ ]: type(Primos)

Out[ ]: list

In [ ]: Temperaturas = [5.5, 6.3, 9.6, 10, 12.4, 14.7, 18, 20.2]

In [ ]: type(Temperaturas)

Out[ ]: list

In [ ]: Nombres = ['Pedro Perez', 'Nelson Vera', 'Carlos Montenegro']

In [ ]: type(Nombres)

Out[ ]: list

In [ ]: Mezcla = [3, 40, 'Hola', 30.5, True, 2+5j]

In [ ]: type(Mezcla)

Out[ ]: list

In [12]: type(Mezcla[2])

Out[12]: str

In [13]: type(Mezcla[5])

Out[13]: complex

### Indexación de Listas

In [ ]: print(Primos[4:])

[11, 13, 17, 19]

In [ ]: print(Primos[:6])

[2, 3, 5, 7, 11, 13]

In [ ]: print(Primos[4:6])

[11, 13]

In [ ]: print(Primos[-7:-2])

[3, 5, 7, 11, 13]

### Operaciones básicas con Listas

In [ ]: print(Primos + Temperaturas)

[2, 3, 5, 7, 11, 13, 17, 19, 5.5, 6.3, 9.6, 10, 12.4, 14.7, 18, 20.2]

In [ ]: print(Primos\*3)

[2, 3, 5, 7, 11, 13, 17, 19, 2, 3, 5, 7, 11, 13, 17, 19, 2, 3, 5, 7, 11, 13, 17, 19]

In [ ]: len(Temperaturas)

Out[ ]: 8

### Métodos de Listas

In [ ]: Nombres.append('Manuel Cifuentes')

In [ ]: print(Nombres)

['Pedro Perez', 'Nelson Vera', 'Carlos Montenegro', 'Manuel Cifuentes']

In [ ]: Mezcla.extend(Primos)

In [ ]: Mezcla.extend(Nombres)

In [ ]: print(Mezcla)

[3, 40, 'Hola', 30.5, True, (2+5j), 2, 3, 5, 7, 11, 13, 17, 19, 'Pedro Perez', 'Nelson Vera', 'Carlos Montenegro', 'Manuel Cifuentes']

In [ ]: Mezcla.extend('Hola Mundo')

In [ ]: print(Mezcla)

[3, 40, 'Hola', 30.5, True, (2+5j), 2, 3, 5, 7, 11, 13, 17, 19, 'Pedro Perez', 'Nelson Vera', 'Carlos Montenegro', 'Manuel Cifuentes', 'H', 'o', 'l', 'a', ' ', 'M', 'u', 'n', 'd', 'o']

In [ ]: Mezcla.insert(3, 'Mundo')

In [ ]: print(Mezcla)

[3, 40, 'Hola', 'Mundo', 30.5, True, (2+5j), 2, 3, 5, 7, 11, 13, 17, 19, 'Pedro Perez', 'Nelson Vera', 'Carlos Montenegro', 'Manuel Cifuentes', 'H', 'o', 'l', 'a', ' ', 'M', 'u', 'n', 'd', 'o']

In [ ]: Mezcla.remove('Mundo')

In [ ]: print(Mezcla)

[3, 40, 'Hola', 30.5, True, (2+5j), 2, 3, 5, 7, 11, 13, 17, 19, 'Pedro Perez', 'Nelson Vera', 'Carlos Montenegro', 'Manuel Cifuentes', 'H', 'o', 'l', 'a', ' ', 'M', 'u', 'n', 'd', 'o']

In [ ]: Mezcla.pop()

Out[56]: 'o'

In [57]: print(Mezcla)

[3, 40, 'Hola', 30.5, True, (2+5j), 2, 3, 5, 7, 11, 13, 17, 19, 'Pedro Perez', 'Nelson Vera', 'Carlos Montenegro', 'Manuel Cifuentes', 'H', 'o', 'l', 'a', ' ', 'M', 'u', 'n', 'd']

In [ ]: Mezcla.pop(2)

Out[ ]: 'Hola'

In [ ]: print(Mezcla)

[3, 40, 30.5, True, (2+5j), 2, 3, 5, 7, 11, 13, 17, 19, 'Pedro Perez', 'Nelson Vera', 'Carlos Montenegro', 'Manuel Cifuentes', 'H', 'o', 'l', 'a', ' ', 'M', 'u', 'n', 'd']

In [ ]: Mezcla.count(3)

Out[ ]: 2

In [ ]: Temperaturas.reverse()

In [ ]: print(Temperaturas)

[20.2, 18, 14.7, 12.4, 10, 9.6, 6.3, 5.5]

In [ ]: Nombres.sort()

In [ ]: print(Nombres)

['Carlos Montenegro', 'Manuel Cifuentes', 'Nelson Vera', 'Pedro Perez']

In [ ]: Temperaturas.sort()

In [ ]: print(Temperaturas)

[5.5, 6.3, 9.6, 10, 12.4, 14.7, 18, 20.2]

In [ ]: Temperaturas.index(12.4)

Out[ ]: 4

In [ ]: Temperaturas.index(12.4,2,7)

Out[ ]: 4

In [ ]: Temperaturas.index(12.4,1,3)

ValueError: 12.4 is not in list

# Tuplas

### Construcción de Tuplas

In [ ]: T = 40, 32, 10.5, 'Hola', 'Mundo'

In [ ]: L = [40, 32, 10.5, 'Hola', 'Mundo']

In [ ]: type(T)

Out[ ]: tuple

In [ ]: type(L)

Out[ ]: list

### Las Tuplas son inmutables

In [ ]: L[4]='Todos'

In [ ]: print(L)

[40, 32, 10.5, 'Hola', 'Todos']

In [85]: T[4]='Todos'

TypeError: 'tuple' object does not support item assignment

# Sets

### Construcción de Sets - Elimina elementos repetidos - No importa el orden

In [ ]: Frutas = {'Naranja', 'Manzana', 'Piña', 'Durazno', 'Manzana'}

In [ ]: print(Frutas)

{'Piña', 'Naranja', 'Manzana', 'Durazno'}

### Conversión a Sets

In [ ]: S = set('manzanas')

In [ ]: print(S)

{'a', 's', 'n', 'm', 'z'}

In [ ]: E = set('duraznos')

In [ ]: print(E)

{'a', 's', 'n', 'u', 'd', 'o', 'z', 'r'}

### Operaciones con Sets

Letras en S pero no en E

In [ ]: S-E

Out[ ]: {'m'}

Letras en S o E o en ambos

In [ ]: S|E

Out[ ]: {'a', 'd', 'm', 'n', 'o', 'r', 's', 'u', 'z'}

Letras en ambos conjuntos

In [ ]: S&E

Out[ ]: {'a', 'n', 's', 'z'}

Letras en S o E pero no en ambos

In [96]: S^E

Out[96]: {'d', 'm', 'o', 'r', 'u'}