IIC1103 - Sección 4

Clase 16: Listas de listas

Prof. Francisca Cattan

Universidad Católica de Chile

11 de octubre, 2022

Repaso listas: métodos para listas

Para las funciones o métodos que veremos a continuación, obtendremos datos o valores como output.

```
1  l1 = [3,4,2,4,9,6]   # lista de numeros
2  l2 = [3,4,2,4,9,6]   # lista de numeros
3  l3 = [3,2,4,4,9,6]   # lista de numeros
4
5  len(l1)              # >>> 6
6  max(l1)              # >>> 9
7  min(l1)              # >>> 2
8  sum(l1)              # >>> 28
9
10 l1.count(4)          # >>> 2
11 l1.count(8)          # >>> 0
12 l1.index(2)          # >>> 2
13 l2.index(9)          # >>> 4
14
15 l1 == l2             # >>> True
16 l1 == l3             # >>> False
```

Repaso listas: métodos para listas

En estas funciones y métodos para utilizar en listas, la lista se modifica permanentemente.

```
1  l1 = [1,2,3]      # lista de numeros
2  l2 = [5,6]        # lista de numeros
3
4  l1.append(4)      # >>> [1,2,3,4]
5  l1.extend(l2)     # >>> [1,2,3,4,5,6]
6  l1.append(l2)     # >>> [1,2,3,4,5,6,[5,6]]
7
8  l2.insert(1,8)    # >>> [5,8,6]
9  l2.insert(7,8)    # >>> [5,8,6,8]
10
11 l2.sort()         # >>> [5,6,8,8]
12 l2.reverse()      # >>> [8,8,6,5]
```

Importante! Duplicado de listas

¿Podemos duplicar listas? ¿Qué ocurre en la memoria?

```
1 lista_original = ['1','A','2','B']
2 lista_duplicada = lista_original
3
4 lista_duplicada.extend(['3','C'])
5
6 print(lista_duplicada)
7 print(lista_original)
```

Importante! Duplicado de listas

¿Podemos duplicar listas? ¿Qué ocurre en la memoria?

```
1 lista_original = ['1','A','2','B']
2 lista_duplicada = lista_original
3
4 lista_duplicada.extend(['3','C'])
5
6 print(lista_duplicada)
7 print(lista_original)
```

```
['1', 'A', '2', 'B', '3', 'C']
['1', 'A', '2', 'B', '3', 'C']
```

Importante! Duplicado de listas

¿Podemos duplicar listas? ¿Qué ocurre en la memoria?

```
1 lista_original = ['1','A','2','B']
2 lista_duplicada = lista_original
3
4 lista_duplicada.extend(['3','C'])
5
6 print(lista_duplicada)
7 print(lista_original)
```

```
['1', 'A', '2', 'B', '3', 'C']
['1', 'A', '2', 'B', '3', 'C']
```

Obs: Está creando un nuevo "alias" para el mismo espacio de memoria. Son la misma variable, y apuntan al mismo ID. Para solucionar esto deben usar copy() o lista[:].

Ejercicios: Listas

Escriba una función que retorne True si la lista está ordenada de manera creciente, y False si no lo está.

```
1 def ordenado(l):
2     .
3     .
4     return ?
```

Ejercicios: Listas

Escriba una función que retorne True si la lista está ordenada de manera creciente, y False si no lo está.

```
1 def ordenado(l):
2     new_l = l[:]
3     new_l.sort()
4
5     if new_l == l:
6         return True
7     else:
8         return False
```

Ejercicios: Listas

Escriba una función que elimine un elemento de la lista según su posición.

```
1 def eliminar_posicion(l,i):
2       .
3       .
4       return ?
```

Ejercicios: Listas

Escriba una función que elimine un elemento de la lista según su posición.

```
1 def eliminar_posicion(l,i):
2     valor = l[i]
3     l.remove(valor)
4     return l
```

Listas de Listas

```
1 lista_1 = [[1,2,3],[3,4,5],[6,7]]     # lista de listas
```

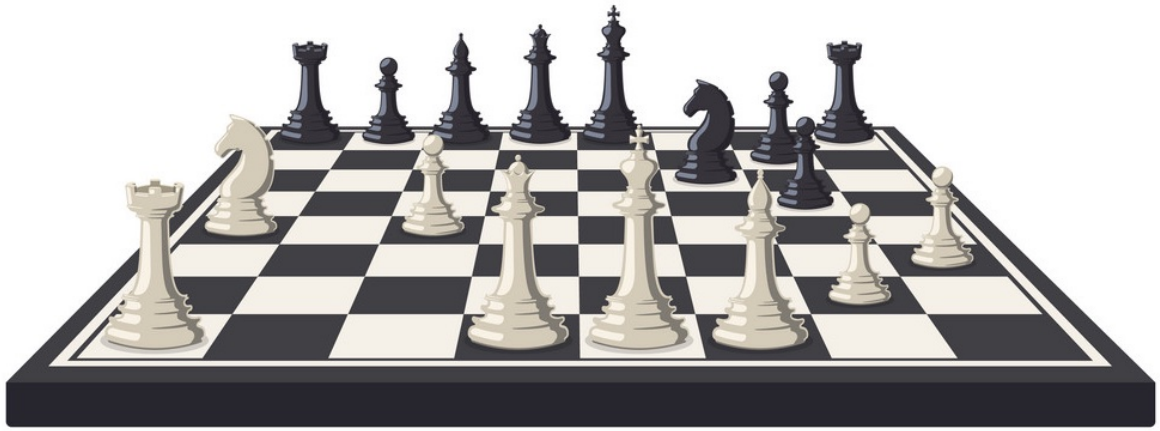
Listas de Listas

```
1 lista_1 = [[1,2,3],[3,4,5],[6,7]]     # lista de listas
```

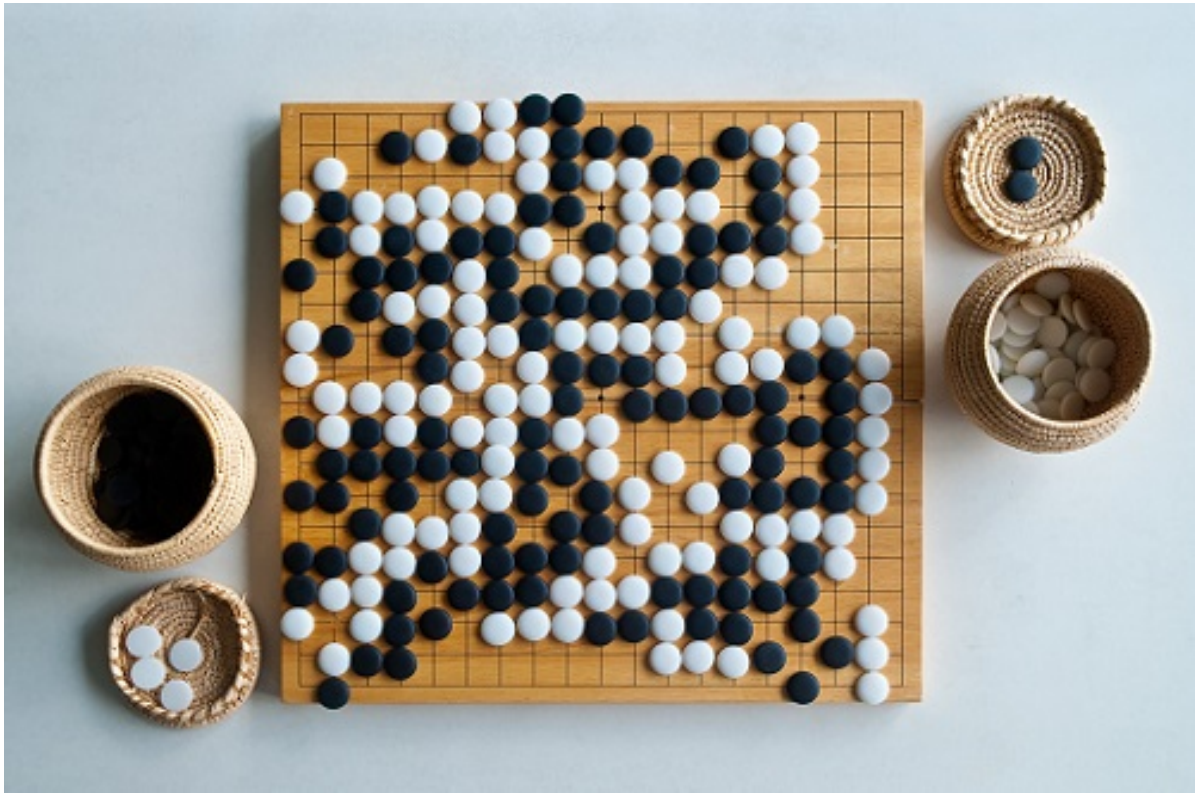
```
1 lista_2 = [['Mujer', 24, 68, 156], ['Hombre', 35, 75, 170]]
```

Una lista de listas en Python es un objeto de lista donde cada elemento de la lista es una lista en sí mismo.

Listas de Listas



Listas de Listas

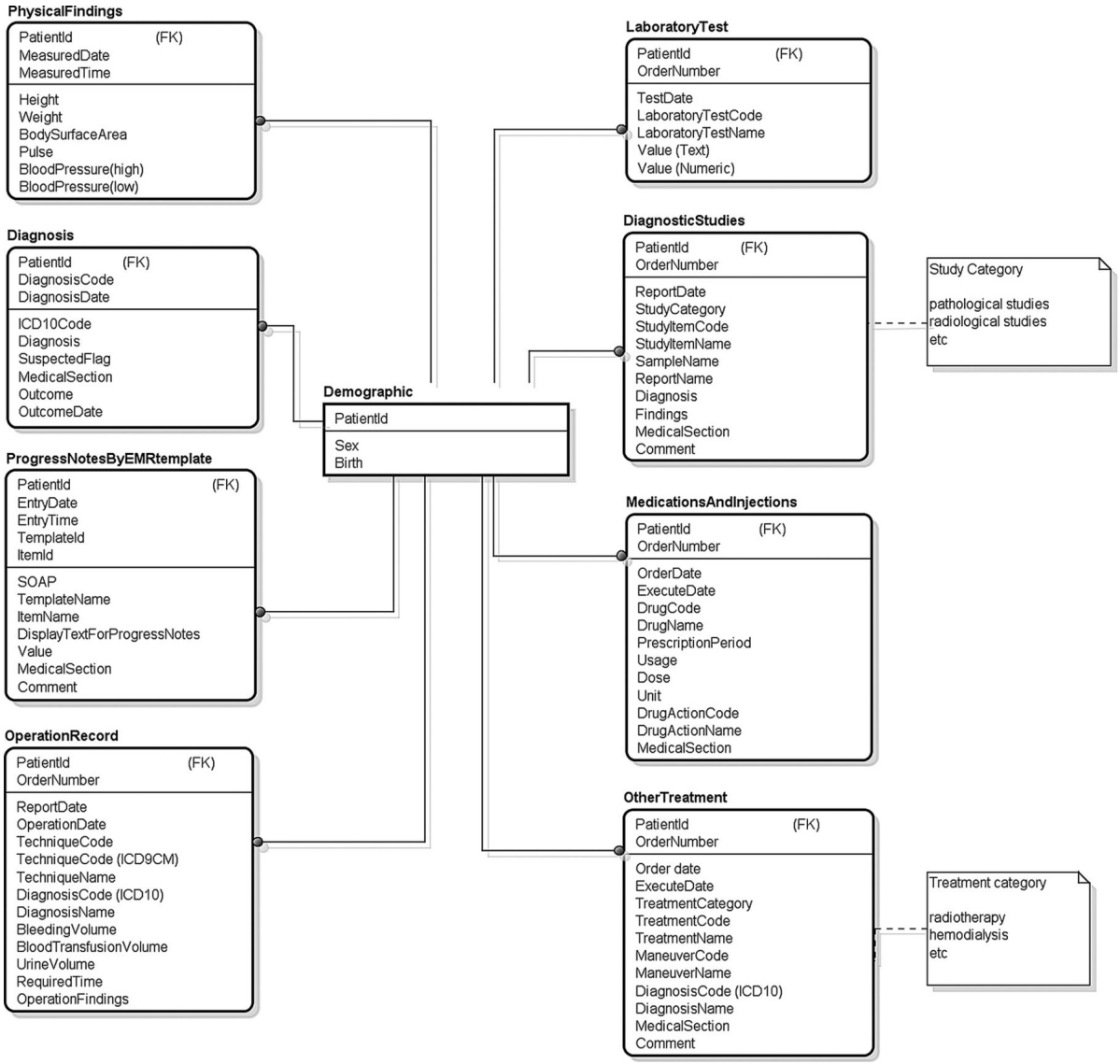


Listas de Listas

matrix dimensions = 11 × 11

$$\begin{pmatrix} -3 & 4 & 5 & 0 & -4 & 9 & 6 & -4 & -3 & -3 & -6 \\ 0 & 7 & 1 & 8 & -2 & -7 & 9 & 9 & 0 & 8 & -7 \\ 0 & 0 & 0 & 1 & -6 & 2 & -3 & 9 & -6 & -7 & 3 \\ 0 & 0 & 0 & 6 & 1 & -2 & 0 & 3 & -6 & -5 & 1 \\ 0 & 0 & 0 & 0 & 6 & -9 & 8 & -8 & -2 & 7 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & -2 & 4 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & -9 & -3 & 4 & 2 & -8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -5 & -4 & -9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -7 & -4 & -5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 5 & -5 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -7 \end{pmatrix}$$

Listas de Listas



Listas de Listas

```
1 df_paper.head()
```

	id	year	title	event_type	pdf_name	abstract	paper_text
0	1	1987	Self-Organization of Associative Database and ...	NaN	1-self-organization-of-associative-database-an...	Abstract Missing	767\n\nSELF-ORGANIZATION OF ASSOCIATIVE DATABA...
1	10	1987	A Mean Field Theory of Layer IV of Visual Cort...	NaN	10-a-mean-field-theory-of-layer-iv-of-visual-c...	Abstract Missing	683\n\nA MEAN FIELD THEORY OF LAYER IV OF VISU...
2	100	1988	Storing Covariance by the Associative Long-Ter...	NaN	100-storing-covariance-by-the-associative-long...	Abstract Missing	394\n\nSTORING COVARIANCE BY THE ASSOCIATIVE\n...
3	1000	1994	Bayesian Query Construction for Neural Network...	NaN	1000-bayesian-query-construction-for-neural-ne...	Abstract Missing	Bayesian Query Construction for Neural\nNetwor...
4	1001	1994	Neural Network Ensembles, Cross Validation, an...	NaN	1001-neural-network-ensembles-cross-validation...	Abstract Missing	Neural Network Ensembles, Cross\nValidation, a...

Listas de Listas



Listas de Listas

```
1 datos_medicos = [['Sexo', 'Edad', 'Peso', 'Altura'], ['
  Mujer', 24, 68, 156], ['Hombre', 35, 75, 170], ['Mujer',
  26, 62, 175], ['Hombre', 62, 61, 169], ['Mujer', 50, 93,
  180], ['Hombre', 31, 67, 171], ['Mujer', 44, 79, 182], ['
  Hombre', 30, 71, 159], ['Mujer', 41, 69, 160], ['Mujer',
  51, 68, 158], ['Mujer', 26, 72, 169], ['Hombre', 23, 73,
  178], ['Mujer', 28, 56, 168], ['Mujer', 25, 65, 159], ['
  Mujer', 30, 82, 166], ['Mujer', 44, 71, 164], ['Hombre',
  33, 66, 162], ['Mujer', 35, 79, 188], ['Mujer', 46, 76,
  187], ['Hombre', 32, 89, 173], ['Mujer', 43, 81, 160], ['
  Hombre', 47, 77, 181], ['Mujer', 37, 73, 178], ['Hombre',
  45, 71, 164], ['Mujer', 49, 72, 181], ['Hombre', 30, 64,
  184], ['Mujer', 32, 70, 162], ['Mujer', 40, 23, 168], ['
  Hombre', 41, 81, 187], ['Mujer', 38, 82, 177]]
```

¿Qué información podemos extraer de estos datos?

Listas de Listas

```
1  mapa = [['M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M'],
2  ['M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M'],
3  ['M', 'M', 'X', 'M', 'M', 'M', 'M', 'M', 'B', 'M'],
4  ['M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M'],
5  ['M', 'M', 'M', 'B', 'M', 'M', 'M', 'M', 'M', 'M'],
6  ['M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M'],
7  ['M', 'M', 'M', 'M', 'M', 'B', 'M', 'M', 'M', 'M'],
8  ['B', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M'],
9  ['M', 'M', 'M', 'B', 'M', 'M', 'M', 'M', 'B', 'M'],
10 ['M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M', 'M']]
```

¿Qué cosas entretenidas podemos hacer con este mapa?

Listas de Listas

¿Cómo creamos una lista de listas?

Con append():

```
1 lista_de_listas = []
2
3 lista_de_listas.append([0])
4 lista_de_listas.append([1, 2])
5 lista_de_listas.append([3, 4, 5])
```

```
[[0], [1, 2], [3, 4, 5]]
```

Listas de Listas

¿Cómo creamos una lista de listas?

Con ciclos:

```
1 lst = []
2 for i in range(3):
3     lst.append([])
4     for j in range(3):
5         lst[i].append(j)
6
7 print(lst)
```

```
[[0, 1, 2], [0, 1, 2], [0, 1, 2]]
```