

# Tecnologías de la Información II

## Tuplas, sets y diccionarios

Daniela Opitz, Diego Caro  
[dopitz@udd.cl](mailto:dopitz@udd.cl)



Basada en presentaciones oficiales de libro Introduction to Programming in Python (Sedgewick, Wayne, Dondero).

Disponible en <https://introcs.cs.princeton.edu/python>

# Outline

- Listas anidadas
- Tuplas
- Sets
- Diccionarios

# Listas Anidadas

- Una lista anidada es una lista que aparece como un elemento en otra lista
- Son útiles para representar matrices
- Una matriz  $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$  por ejemplo en python quedaría así:

`[[a11, a12, a13], [a21, a22, a23], [a31, a32, a33]]`

# Listas Anidadas

```
1 from random import random
2
3 rows = 3
4 cols = 4
5 matrix = []
6 for i in range(rows):
7     matrix.append([0]*cols)
8
9 print(matrix)
```

Número de filas

Número de columnas

```
1 def fillrandom(matrix, rows, cols):
2     for i in range(rows):
3         for j in range(cols):
4             matrix[i][j] = random()
5
6 fillrandom(matrix, rows, cols)
7 print(matrix)
```

## Matriz de ceros

Número de filas

matrix[0] → [0, 0, 0, 0]  
matrix[1] → [0, 0, 0, 0]  
matrix[2] → [0, 0, 0, 0]

Número de columnas

# Listas Anidadas

- Cómo llamo a cada elemento de la lista anidada en python?

<code>matrix[0][0]</code>	<code>matrix[0][1]</code>	<code>matrix[0][2]</code>	<code>matrix[0][3]</code>
<code>matrix[1][0]</code>	<code>matrix[1][1]</code>	<code>matrix[1][2]</code>	<code>matrix[1][3]</code>
<code>matrix[2][0]</code>	<code>matrix[2][1]</code>	<code>matrix[2][2]</code>	<code>matrix[2][3]</code>

# tuple()

- Una tupla es una secuencia de valores agrupados. Sirve para agrupar en un único valor, valores que deben ir juntos.
- El tipo de datos que representa una tupla se llama tuple().
- Las tuplas se crean con valores separados por comas y entre paréntesis (elem1, elem2)
- La estructura de datos tuple() es parecida a estructura de lista pero a diferencia de esta, tuple() es INMUTABLE.

# set()

- Un set es una colección no ordenada ni indexada de elementos.
- Los elementos son únicos y no se pueden repetir.
- El tipo de datos que representa un set es set()
- Los sets pueden crear con el operador corchete {elem1, elem2}
- Sirven para:
  - chequear si existe un elemento
  - realizar operaciones de unión, intersección, diferencia entre conjuntos
  - eliminar duplicados de una secuencia

# Tuplas y sets



```
1 A = {[1, 2, 3, 4], list(['a', 'b', 'c'])}  
2 print('A:', A)  
3 print('(1, 2, 3, 4) in A?:', (1, 2, 3, 4) in A)
```

```
$ python3 badset.py  
Traceback (most recent call last):  
  File "badset.py", line 1, in <module>  
    A = {[1, 2, 3, 4], ['a', 'b', 'c']}  
TypeError: unhashable type: 'list'
```



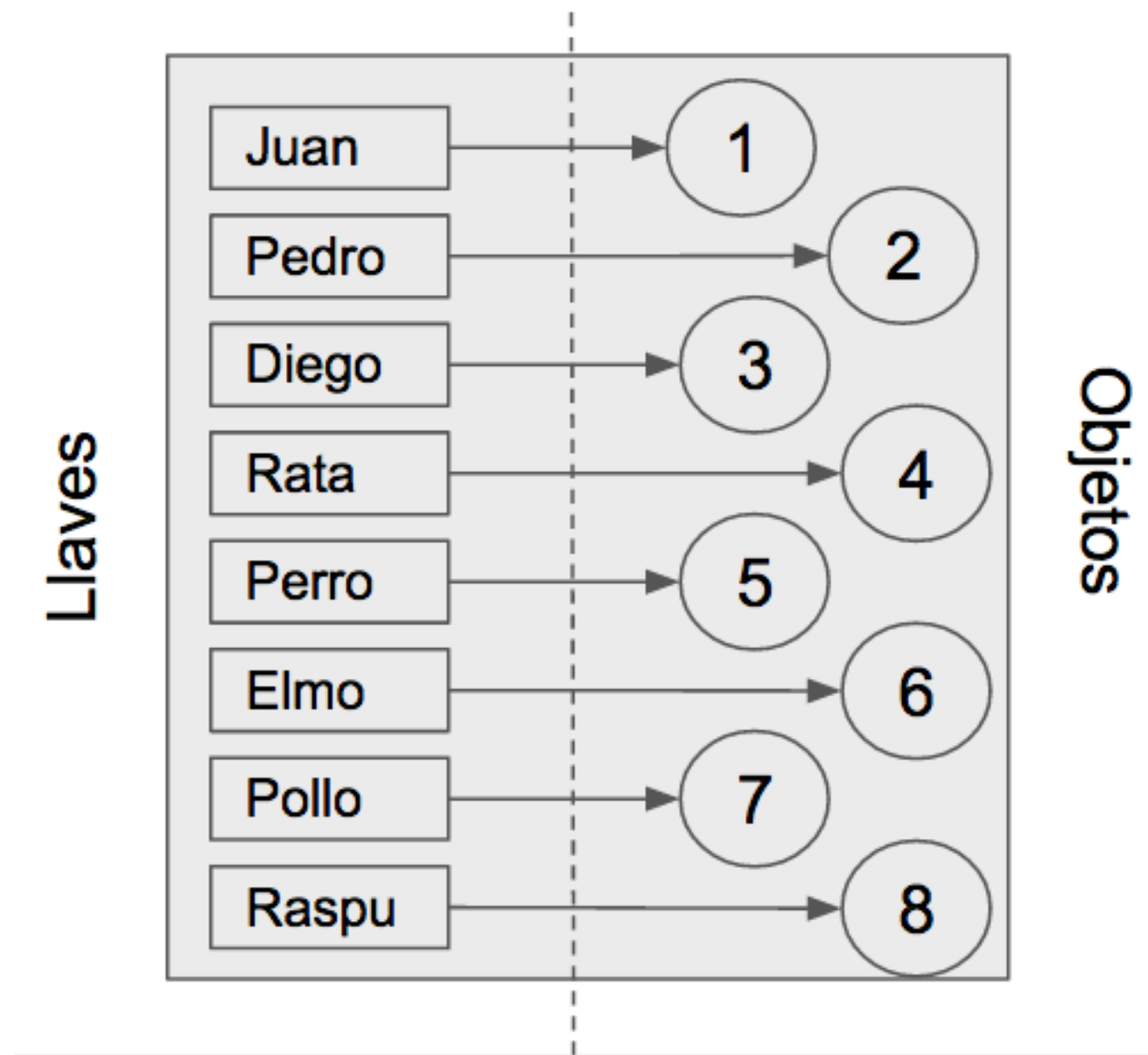
```
1 A = {(1, 2, 3, 4), tuple(['a', 'b', 'c'])}  
2 print('A:', A)  
3 print('(1, 2, 3, 4) in A?:', (1, 2, 3, 4) in A)
```

```
$ python3 tupleset.py  
A: {(1, 2, 3, 4), ('a', 'b', 'c')}  
(1, 2, 3, 4) in A?: True
```



# dict()

- Un diccionario es una colección de datos que no esta ordenada pero está indexada
- El tipo de datos de los diccionarios es dict()
- Un diccionario asocia un valor a una clave (key)
- La llave debe ser inmutable (de tipo int(), str(), tuple())

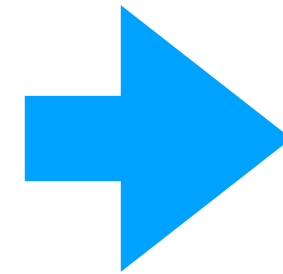


# Aplicaciones de los diccionarios

<i>application</i>	<i>key</i>	<i>value</i>
contacts	name	phone number, address
credit card	account number	transaction details
file share	name of song	computer ID
dictionary	word	definition
web search	keyword	list of web pages
book index	word	list of page numbers
cloud storage	file name	file contents
domain name service	domain name	IP address
reverse DNS	IP address	domain name
compiler	variable name	value and type
internet routing	destination	best route
...	...	...

# dict()

Dominio	Dirección IP
udd.cl	201.221.123.142
ingenieria.udd.cl	201.221.123.142
google.com	64.233.190.101
pokemongo.com	13.33.131.6
<b>llave (key)</b>	<b>valor</b>



```
1 ipaddress = dict()
2 ipaddress['udd.cl'] = '201.221.123.142'
3 ipaddress['ingenieria.udd.cl'] = '201.221.123.142'
4 ipaddress['google.com'] = '64.233.190.101'
5 ipaddress['pokemongo.com'] = '13.33.131.6'
6
7 print('Dirección IP de udd.cl:', ipaddress['udd.cl'])
```

```
1 ipaddress = {'udd.cl': '201.221.123.142',
2              'google.com': '64.233.190.101'}
3 ipaddress['ingenieria.udd.cl'] = '201.221.123.142'
4 ipaddress['pokemongo.com'] = '13.33.131.6'
5
6 print('Dirección IP de udd.cl:', ipaddress['udd.cl'])
```

# Ejemplo: Contando Palabras

```
1 def word_count(message):
2     counts = dict()
3     words = message.split()
4
5     for word in words:
6         if word in counts:
7             counts[word] += 1
8         else:
9             counts[word] = 1
10
11     return counts
12
13
14 mambo = '''Desde lima vengo a mi machaguay
15 Desde Lima vengo a mi machaguay
16 A bailar el mambo de mi machaguay
17 A bailar el mambo de mi machaguay'''
18
19 print(word_count(mambo))
```



```
{'A': 2,
 'Desde': 2,
 'Lima': 1,
 'a': 2,
 'bailar': 2,
 'de': 2,
 'el': 2,
 'lima': 1,
 'machaguay': 4,
 'mambo': 2,
 'mi': 4,
 'vengo': 2}
```

# Resumen

## Estructuras de datos

- `tuple()`: colección inmutable y ordenada
- `set()`: conjunto de elementos, no-ordenada
- `dict()`: tabla de símbolos clave:valor

## Conceptos

- **Inmutable**: que no se puede modificar una vez creada.

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

[https://docs.python.org/3/reference/lexical\\_analysis.html](https://docs.python.org/3/reference/lexical_analysis.html)

		Built-in Functions		
abs()	delattr()	hash()	memoryview()	set()
all()	dict()	help()	min()	setattr()
any()	dir()	hex()	next()	slice()
ascii()	divmod()	id()	object()	sorted()
bin()	enumerate()	input()	oct()	staticmethod()
bool()	eval()	int()	open()	str()
breakpoint()	exec()	isinstance()	ord()	sum()
bytearray()	filter()	issubclass()	pow()	super()
bytes()	float()	iter()	print()	tuple()
callable()	format()	len()	property()	type()
chr()	frozenset()	list()	range()	vars()
classmethod()	getattr()	locals()	repr()	zip()
compile()	globals()	map()	reversed()	__import__()
complex()	hasattr()	max()	round()	

<https://docs.python.org/3/library/functions.html>