

UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
Fundamentos de Computación y Programación (10110-1)


UdeSantiago

CLASE N°12

LISTAS Y CICLO FOR-IN

LISTAS


UdeSantiago

- Las listas son un tipo de dato en Python que, a diferencia de los que hemos visto hasta ahora, **se compone de datos más pequeños**
- Permiten almacenar una **secuencia de elementos**, incluso de distintos tipos, en una **única estructura**
- Utiliza una **indexación posicional** para acceder a los elementos que contiene

Lista

Posición (índice)	→	0	1	2	3	4	5	6
Elementos	→	"Juan"	"Pino"	4	2.3	3.7	4.2	4.9

LISTAS



- Para declarar una lista:

```
<nombre lista> = [<elem1>, <elem2>, ..., <elemN>]
```

- Para acceder a un elemento en particular:

```
<nombre lista>[<posición>]
```

- Donde la **posición** es un número entre 0 y el largo de la lista menos uno ($\text{len}(\text{<nombre lista>}) - 1$)

3

OPERADORES CON LISTAS



- Algunos **operadores** funcionan de manera particular cuando se trabaja con listas

Operador	Utilización	Función
+	<lista> + <lista>	Concatenar dos listas
*	<lista> * <int>	Repite n veces una lista
:	<lista>[<int>:<int>]	Selecciona una parte de una lista (rango dado)
in	<elemento> in <lista>	Entrega True si el elemento está en la lista, False si no está

4

FUNCIONES NATIVAS CON LISTAS



- Tenemos como opción varias funciones nativas para operar con listas

Llamado	Propósito
<code>min(<lista>)</code>	Entrega el mínimo valor en la lista
<code>max(<lista>)</code>	Entrega el máximo valor en la lista
<code>sum(<lista>)</code>	Suma todos los elementos de la lista
<code>len(<lista>)</code>	Entrega el número de elementos de la lista
<code>del <lista>[<posición>]</code>	Elimina un elemento de la lista
<code>lista.append(<elemento>)</code>	Agrega un elemento al final de la lista

5

SENTENCIA FOR - IN



- Permite definir un bloque de operaciones a realizar **para cada elemento de una secuencia**
 - La iteración termina al **acabarse** los elementos en la secuencia
 - Sintaxis:

```
for <identificador> in <lista de elementos>:  
    <operaciones a realizar>  
    # Normalmente con el valor en <identificador>
```

6

FUNCIÓN range()



- Para **generar listas** rápidamente podemos usar la función **range()**
 - Nos permite iterar un ciclo **for-in** una cantidad fija de veces
 - Tiene 3 parámetros, 1ero y 3ero son opcionales:
 - **valor inicial** de la secuencia a generar (0 si se omite)
 - **límite superior** de la secuencia, el que no es incluido
 - **Paso**, o **diferencia aritmética** entre **elementos consecutivos** de la secuencia (1 si se omite)

7

STRINGS



- Los **string**(que ya hemos usados para dar mensajes) y al igual que las listas, son un tipo de dato **compuesto**
 - A diferencia de las listas, están compuestos **sólo de un tipo de dato**: el **caracter (chr)**
 - Los string son **inmutables**, es decir, no podemos modificar sus caracteres
 - Sin embargo, esto no impide hacer operaciones de **búsqueda**, **conteo** o **comparación** sobre strings

8

PARA LA PRÓXIMA CLASE



- Profundizaremos las operaciones sobre strings
- Aprenderemos a alimentar el programa por vías distintas a la entrada por teclado
- Profundizaremos en la comunicación humano-computador
- Aunque primero podríamos hacer una autoevaluación

9



¿CONSULTAS?

10