

Listas

(2ª parte)

Clase #17

IIC1103 – Introducción a la Programación

El plan de hoy es...

- Encuesta $\frac{1}{2}$ semestre (les llegó un mail)
 - Si tienen sugerencias de mejora, recuerden explicar bien para poder implementarlas 😊
 - Discutiremos resultados a la vuelta
- Próx semana: receso universitario (no hay clases)

Menti 🧐 🧠



List slices

- `lista = [0,1,2,3,4,5,6,7,8,9]`
 - `lista[1:3]`
 - `lista[:]`
 - `lista[:5]`
 - `lista[3:5]=['a','b']`
-
- Recordar que listas pueden combinar tipos de elementos (elementos pueden ser listas)

for en listas

- ```
L = ['buho', 'gato', 'delfin', 'jirafa', 'ave', 'perro', 'foca']
for i in range(len(L)):
 print(f"{i}=>{L[i]}")
print("*****")
for elem in L:
 print(f"{elem}")
```

# Problema #0

- Escribe una función que imprima una frase, con una palabra por línea
- Ejemplo:
  - `imprimefrase("Como estás hoy")`
  - imprime:
  - Como
  - estás
  - hoy

## String → Lista

- `s= "abracadabra pata de cabra, abracadabra!"`
- `lista_de_palabras = s.split()`
- `0`
- `lista_de_palabras = s.split(" ")` #recibe el delimitador

## String → Lista

- `s= “abracadabra pata de cabra, abracadabra!”`
- `lista_de_caracteres = list(s)`



## Lista → String

- `lista = ["abracadabra", "pata", "de", "cabra"]`
- `delimiter="."`
- `nuevo_string = delimiter.join(lista)`

# Problema 1

- Una función retorna un valor
- ... ¡pero ese valor puede ser una lista!
- `def palabras_empiezan_con(lista,x):`
- entrega una lista con todas las palabras que empiezan con la letra x
- ejemplo:
- `dias =`  
`["lunes","martes","miercoles","jueves","viernes","sabado","domingo"]`
- `palabras_empiezan_con(dias,"m")` entrega  
`["martes","miercoles"]`

# Solución

- **def** palabras\_empiezan\_con(lista,x):  
    nueva\_lista = []  
    **for** elemento **in** lista:  
        **if** elemento[0]==x:  
            nueva\_lista.append(elemento)  
    **return** nueva\_lista

# Resumen hasta el momento

- Dado x: elemento, l,m:lista, s:string, i,j:int
- `l.append(x)` → agrega x al final
- `s.split(d)` → entrega lista de string s, separado según d
- `d.join(l)` → entrega string de lista l, unido según d
- `list(s)` → entrega lista de cada carácter de string s
- `l[i:j]` → trozo de lista entre i y j-1
- `len(l)` → nº elementos de l
- `list(range(...))` → crear lista usando range