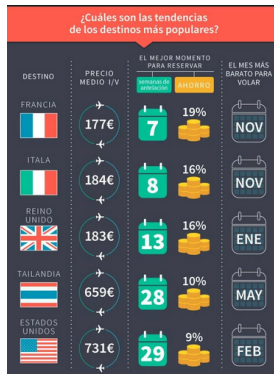


LISTAS - Sesión 8A

Computer Science

CS1100 - Introducción a Ciencia de la Computación



▲ Uke [Filter area]	▲ Subject	▼ Average UCAS points achieved	▼ % of employed with grad job
<input type="checkbox"/> UNIVERSITY OF CAMBRIDGE	Computer Science	580	100%
<input checked="" type="checkbox"/> UNIVERSITY OF WARWICK	Computer Science	430	100%
<input type="checkbox"/> UNIVERSITY OF YORK	Computer Science	460	100%
<input type="checkbox"/> UNIVERSITY OF BATH	Computer Science	440	100%
<input type="checkbox"/> DURHAM UNIVERSITY	Computer Science	440	100%
<input type="checkbox"/> IMPERIAL COLLEGE LONDON	Computer Science	510	98%
<input type="checkbox"/> UNIVERSITY OF BRISTOL	Computer Science	430	95%
<input type="checkbox"/> UNIVERSITY OF SOUTHAMPTON	Computer Science	450	95%
<input type="checkbox"/> UNIVERSITY OF SHEFFIELD	Computer Science	390	95%
<input type="checkbox"/> UNIVERSITY OF PRESTON	Computer Science	300	95%

Logro de la Sesión

Al finalizar esta sesión, estarás en la capacidad de:

- Desarrollar programas en Python en donde podrán aplicar adecuadamente los loops For/While y utilizar sublistas/matrices.

FOR vs WHILE

CADENA (STRING)

Si tenemos una Lista en Python, podemos iterar a través de los elementos con un loop FOR. Iterable es un objeto capaz de retornar sus miembros, uno por uno.

```
1 for fruta in ["manzana", "durazno", "uva"]:  
    print ("Comemos", fruta)
```

```
1 Comemos  manzana  
2 Comemos  durazno  
3 Comemos  uva
```

FOR vs WHILE

LISTAS

WHILE se utiliza cuando necesitamos repetir un conjunto de instrucciones hasta que se cumpla una condición en particular. FOR es utilizado para estructuras iterables, como Listas por ejemplo.

```
1  lineas = []
2  print('Ingrese líneas de texto')
3  print('Ingrese una línea vacía para
   finalizar.')
4
5  linea = input('Su línea: ')
6  while linea != "":
7      lineas.append(linea)
8      linea = input("Siguiente línea: ")
9  print('Sus líneas fueron:')
10 for linea in lineas:
```

FOR vs WHILE

LISTAS

OUTPUT:

```
1
2 Ingrese líneas de texto
3 Ingrese una línea vacía para finalizar.
4 Su línea: Primera línea
5 Siguiente línea: Segunda Línea
6 Siguiente línea: Tercera Línea
7 Siguiente línea:
8 Sus líneas fueron:
9 Primera línea
10 Segunda Línea
11 Tercera Línea
12
13 Process finished with exit code 0
```



¿Cómo se representa?

FOR vs WHILE

NO es muy común utilizar WHILE para iterar Listas. El FOR es más recomendable en estos casos, ya que las Listas son estructuras iterables.

```
1 lista = [23, 435, 345]
2 i = 0
3 while i < len (lista):
4     print (lista[i])
5     i = i + 1
```

```
1 lista = [23, 435, 345]
2 for i in range(len(lista)):
3     print (lista[i])
```

El resultado en ambos casos es el mismo:

LISTAS

ITERACION

Iteración por elemento

```
1 lista=[2,4,6,7,1]
2 for i in lista:
3     print (i)
4 print ()
```

Iteración basada en el índice

```
1 lista=[2,4,6,7,1]
2 for i in range(len(lista)):
3     print (lista[i])
```


LISTAS

ITERACION CON WHILE

Solicita ingreso de 3 vocales en mayúscula. Valida ingreso. Las vocales ingresadas se añaden a una lista hasta completar 3 elementos.

```
1 lista = []
2 for i in range(3):
3     r = False
4     while not r:
5         print("Ingrese una vocal: ")
6         caracter = input("")
7         if caracter.upper() in ['A', 'E', 'I', 'O',
8                                ', 'U']:
9             r=True
10            lista.append(caracter)
11 print(lista)
```

MATRICES

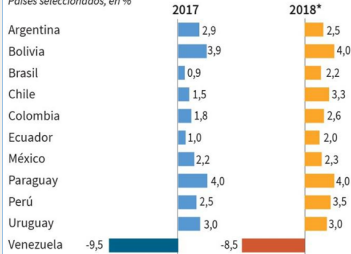
Es probable que necesitemos almacenar colecciones de valores que poseen una distribución de tabla bidimensional. Estos conjuntos, contienen valores tanto en filas como en columnas.

Imaginemos que necesitamos almacenar datos como los mostrados a continuación.

El crecimiento de América Latina

Variación del PIB

Países seleccionados, en %



Fuente: Cepal (abril 2018)

Evolución del PIB

América Latina y el Caribe, en %



*proyección © AFP

LISTAS

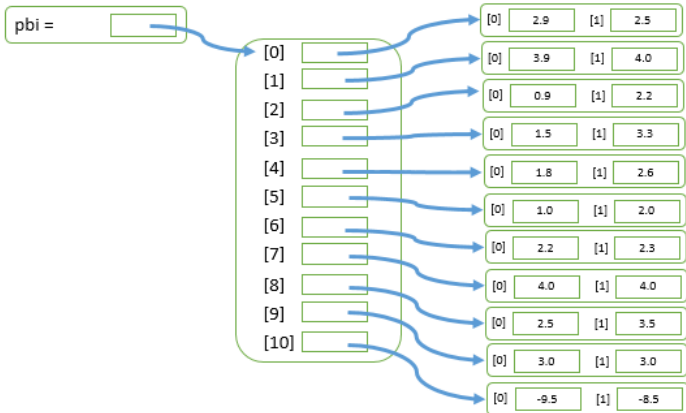
Si le otorgamos una estructura tabular a los datos de la imagen tendríamos lo siguiente:

	2017	2018
Argentina	2.9	2.5
Bolivia	3.9	4.0
Brasil	0.9	2.2
Chile	1.5	3.3
Colombia	1.8	2.6
Ecuador	1.0	2.0
México	2.2	2.3
Paraguay	4.0	4.0
Perú	2.5	3.5
Uruguay	3.0	3.0
Venezuela	-9.5	-8.5

CREACIÓN DE MATRICES

Python no dispone de un tipo de dato para crear tablas. Sin embargo, podemos crear una estructura tabular de dos dimensiones, utilizando las Listas de Python.

```
1 PAISES = 11
2 AÑOS = 2
3 paises = [ "Argentina", "Bolivia", "Brasil",
4            ", "Chile",
5            "Colombia", "Ecuador", "Mexico",
6            ", "Paraguay",
7            "Perú", "Uruguay", "Venezuela"
8            ]
9 pbi = [
10        [2.9 , 2.5],
11        [3.9 , 4.0],
12        [0.9 , 2.2],
13        [1.5 , 2.2]
```



MATRICES

Podría darse el caso, que tengamos que crear una matriz de dimensiones tan grandes, que complique la inicialización elemento por elemento. En tal situación, podemos proceder en dos pasos:

1. Crear una lista que servirá para almacenar las filas.

```
1 matriz = [ ]
```

2. Luego creamos una lista utilizando la replicación (con el número de columnas como tamaño) y añadiendo esta a la lista de filas.

MATRICES

```
1 FILAS = 4
2 COLUMNAS = 6
3
4 for i in range(FILAS) :
5     fila = [0] * COLUMNAS
6     matriz.append(fila)
```

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

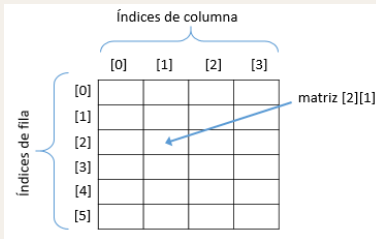
Output: [[0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0]]

MATRICES

Acceso a elementos

Para acceder a un elemento de la tabla, lo hacemos a través de los índices de la fila y la columna.

```
1 elemento = matriz [2][1]
```



MATRICES

Acceso a elementos

Para obtener acceso a todos los elementos de la tabla, utilizamos ciclos repetitivos anidados:

```
1 PAISES = 11
2 AÑOS = 2
3 paises = [ "Argentina", "Bolivia", "Brasil",
             ", "Chile", "Colombia", "Ecuador", "
             Mexico", "Paraguay", "Perú", "Uruguay",
             "Venezuela" ]
4 pbi = [ [2.9 , 2.5], [3.9 , 4.0], [0.9 ,
        2.2], [1.5 , 3.3],
5         [1.8 , 2.6], [1.0 , 2.0], [2.2 ,
        2.3], [4.0 , 4.0],
6         [2.5 , 3.5], [3.0 , 3.0], [-9.5 ,
        -8.5] ]
```

MATRICES

Acceso a elementos

	2017	2018
Argentina	2.9	2.5
Bolivia	3.9	4.0
Brasil	0.9	2.2
Chile	1.5	3.3
Colombia	1.8	2.6
Ecuador	1.0	2.0
Mexico	2.2	2.3
Paraguay	4.0	4.0
Perú	2.5	3.5
Uruguay	3.0	3.0
Venezuela	-9.5	-8.5

LISTAS

Cálculo de totales de Filas y Columnas

Es común calcular los totales por columna y por fila.

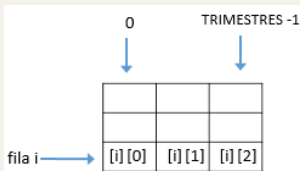
DIVISION	Trimestre 1	Trimestre 2	Trimestre 3
Línea Blanca	450	650	342
Electrodomésticos	340	487	767
Juguetería	134	212	354
Perecibles	180	464	565
Limpieza	647	324	232

MATRICES

Cálculo de totales de Filas

Cálculo del total de ventas perteneciente a la fila i (Una División en particular, Juguetería por ejemplo) en todos los trimestres.

```
1 TRIMESTRES = 3
2 DIVISIONES = 5
3
4 total = 0
5 for j in range(TRIMESTRES):
6     total = total + matriz[i][j]
```



MATRICES

Cálculo de totales de Filas

Cálculo del total de ventas perteneciente a la fila i (Una División en particular, Juguetería por ejemplo) en todos los trimestres.

```
1 DIVISIONES = 5
2 TRIMESTRES = 3
3 divisiones = ["Línea Blanca", "Electrodomés-
               sticos", "Juguetería", "Perecibles", "
               Limpieza"]
4
5 datos = [ [450, 650, 342], [340, 487,
               767], [134, 212, 354],
6           [180, 464, 565], [647, 324, 232],
7           ]
8 print ("          DIVISION          TRIM1
```

MATRICES

Cálculo de totales de Filas...

Cálculo del total de ventas perteneciente a la fila i (Una División en particular, Juguetería por ejemplo) en todos los trimestres.

```
1
2 suma = 0
3 i = 2
4 for j in range(TRIMESTRES) :
5     suma = suma + datos[i][j]
6 print("\nEl total de ventas de la División
    Juguetería es: ", suma)
```

MATRICES

Cálculo de totales de Filas

	Trim1	Trim2	Trim3
Línea Blanca	450	650	342
Electrodomésticos	340	487	767
Juguetería	134	212	354
Perecibles	180	464	565
Limpieza	647	324	232

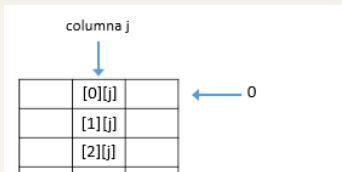
El total de ventas de la División Juguetería es: 700

MATRICES

Cálculo de totales de Columnas

Cálculo del total de ventas en un TRIMESTRE en particular en todas las DIVISIONES. En este caso, la variable i va desde el rango 0 hasta $DIVISIONES - 1$

```
1  
2  
3 total = 0  
4 for i in range(DIVISIONES) :  
5     total = total + matriz[i][j]
```



MATRICES

Cálculo de totales de Columnas

```
1 suma = 0
2 j = 1
3 for i in range(DIVISIONES) :
4     suma = suma + datos[i][j]
5 print("\nEl total de ventas del segundo
    Trimestre es: ", suma)
```

MATRICES

Cálculo de totales de Filas

Línea Blanca	450	650	342
Electrodomésticos	340	487	767
Juguetería	134	212	354
Perecibles	180	464	565
Limpieza	647	324	232
El total de ventas del segundo Trimestre es: 2137			

MATRICES

Cálculo de totales de Filas y Columnas

```
1 PAISES = 11
2 AÑOS = 2
3 paises = [ "Argentina", "Bolivia", "Brasil",
             ", "Chile", "Colombia", "Ecuador", "
             Mexico", "Paraguay", "Perú", "Uruguay",
             "Venezuela" ]
4 pbi = [ [2.9 , 2.5], [3.9 , 4.0], [0.9 ,
             2.2], [1.5 , 3.3],
5         [1.8 , 2.6], [1.0 , 2.0], [2.2 ,
             2.3], [4.0 , 4.0],
6         [2.5 , 3.5], [3.0 , 3.0], [-9.5 ,
             -8.5] ]
7 print ("                2017    2018")
8 for i in range(PAISES) :
```

MATRICES

Cálculo de totales de Filas y Columnas

	2017	2018
Argentina	2.9	2.5
Bolivia	3.9	4.0
Brasil	0.9	2.2
Chile	1.5	3.3
Colombia	1.8	2.6
Ecuador	1.0	2.0
Mexico	2.2	2.3
Paraguay	4.0	4.0
Perú	2.5	3.5
Uruguay	3.0	3.0
Venezuela	-9.5	-8.5

MATRICES

Ejemplo 1

Dada la siguiente tabla y el código Python que la genera:

País	Precio local	Tasa de cambio dólar	Precio BM dólares	Salario mínimo local	Salario mínimo dólares	Número de BigMacs
Costa Rica	2290	569,0	4,0	300255,0	527,7	131
Argentina	75	22,8	3,3	9500	416,7	127
Panamá	5,9	1,0	5,9	744,0	744,0	126
Chile	2600	634,0	4,1	276000,0	435,3	106
Uruguay	140	30,8	4,5	13430,0	436,0	96
Paraguay	26542	5601,0	4,7	2200000,0	392,8	83
Colombia	10900	2964,0	3,7	869453,0	293,3	80
Peru	11,5	3,3	3,5	850,0	257,6	74
Ecuador	5,5	1,0	5,5	386,0	386,0	70
Bolivia	34	7,0	4,9	2000,0	287,4	59
Brasil	16,5	3,6	4,6	965	268,1	58
Mexico	48	19,6	2,5	2687,0	137,2	56
Nicaragua	168	31,3	5,4	8445,0	269,6	50
Venezuela	168000	690854,0	0,2	1308000,0	1,9	8

MATRICES

Ejemplo 1

Dada la siguiente tabla y el código Python que la genera:

```
1 PAISES = 14
2 DATOS = 6
3 paises = [ "Costa Rica", "Argentina", "
             Panamá", "Chile", "Uruguay",
4             "Paraguay", "Colombia", "Perú"
             , "Ecuador", "Bolivia",
5             "Brasil", "México", "Nicaragua",
             "Venezuela" ]
6 datos = [ [2290, 569, 4 , 300255, 527.7,
             131],
7           [75, 22.8, 3.3, 9500, 416.7,
             127],
8           [5.9, 1, 5.9, 744, 744, 126],
```

MATRICES

Ejemplo 1...

```
1
2 for i in range(PAISES) :
3     print ("%10s"%países[i], end="    ")
4     for j in range(DATOS) :
5         print ("%12.2f" % (datos[i][j]), end= "
6             ")
7     print ()
```

Ejemplo 1

- a) solicite al usuario el ingreso de un país e imprima todos los datos que le corresponden.
- b) Totalice la cantidad de BigMacs que pueden adquirirse en América Latina.
- c) Imprima el nombre de los países donde es posible adquirir más de 100 BigMacs.

Ejemplo 1

Solución 1a

```
1 pais=input("\nIngrese país: ")
2 if pais in paises:
3     pos = paises.index(pais)
4     for j in range(DATOS):
5         print ("%15d" % datos[pos][j],end="
")
```

Ejemplo 1

Solución 1b

```
1 suma = 0
2 for i in range(PAISES):
3     suma = suma + datos[i][5]
4 print ("\nLa cantidad total de BigMacs es:
    ", suma)
```

Ejemplo 1









Solución 1c

```
1 print("Los países con ingreso mínimo  
   superior a 100 dólares son:")  
2 for i in range(PAISES):  
3     if (datos[i][5] >100):  
4         print(países[i], end= " ")
```

MATRICES

Ejercicio 1

Dada la siguiente tabla:

#	País				Total
1	 Estados Unidos (USA)	1022	794	704	2520
-	 Unión Soviética (URS)	395	319	296	1010
2	 Reino Unido (GBR)	263	295	289	847
3	 China (CHN)	227	163	153	543
4	 Alemania (GER)	219	246	269	734

- 1.1) Calcule e imprima el total de medallas obtenidas por China.
- 1.2) Calcule e imprima el total de medallas de bronce.
- 1.3) Calcule e imprima el total de todas las medallas obtenidas en la tabla.

Ejercicio 1

Solución

```
1 PAISES = 5
2 TIPOS_MEDALLA = 3
3 nombres_paises = [ "Estados Unidos", "Unió
    n Soviética", "Reino Unido",
4                     "China", "Alemania"]
5 medallas = [
6     [1022, 794, 704], [395, 319, 296],
7     [263, 295, 289],
8     [227, 163, 153], [219, 246, 269]
9 ]
10 print ("                ORO   PLATA
11         BRONCE")
12 for i in range(PAISES) :
```

Ejercicio 1

Solución 1a

```
1 total = 0
2 for j in range(TIPOS_MEDALLA) :
3     total = total + medallas[3][j]
4 print("\nEl total de medallas de China: ",
      total)
```

Ejercicio 1

Solución 1b

Cálculo del total de ventas de una columna i (medallas de bronce)

```
1 total = 0
2 for i in range(PAISES) :
3     total = total + medallas[i][2]
4 print("\nEl total de medallas de bronce: "
      , total)
```

Ejercicio 1

Solución 1c

```
1 suma = 0
2 for i in range(PAISES):
3     for j in range(TIPOS_MEDALLA):
4         suma = suma + medallas[i][j]
5 print ("\nLa cantidad total de medallas es
      : ", suma)
```

	ORO	PLATA	BRONCE
Estados Unidos	1022	794	704
Unión Soviética	395	319	296
Reino Unido	263	295	289
China	227	163	153
Alemania	219	246	269

El total de medallas de China: 543

El total de medallas de bronce: 1711

La cantidad total de medallas es: 5654

MATRICES

Your Turn!!!



Listas : Resumen

LISTAS

En la sesión de hoy, aprendimos a identificar la iteración con FOR y WHILE en Listas y a utilizar Matrices.

Aprendimos:

- a) Cuándo es recomendable iterar con FOR ó WHILE
- b) Calcular el total por columna en una matriz
- c) Calcular el total por fila en una matriz
- d) Calcular el total de todas las filas y columnas
- e) Accesar a una celda determinada de la tabla.
- f) Hicimos ejercicios en HackerRank sobre estos temas.

See you next week!

