

TOCALE NOMÁS



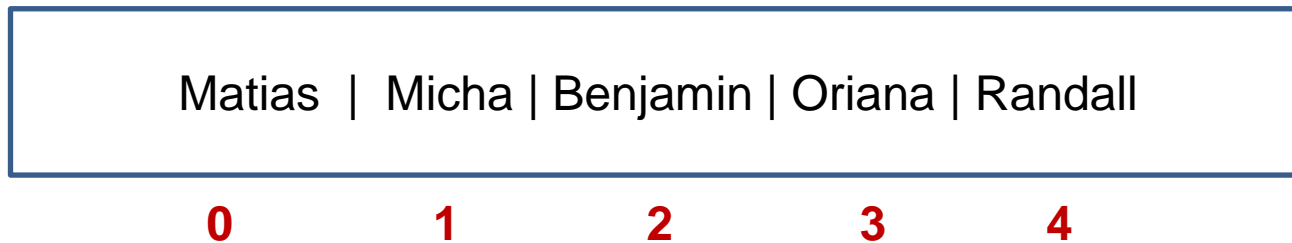
**DE PURO PROGRAMAR EN
PYTHON**

memegenerator.net

Crear un algoritmo que permita:

- Al usuario ingresar los nombres de los estudiantes de la asignatura de Programación y almacenarlos en una Lista llamada ESTUDIANTES.
- **NO PUEDEN SER INGRESADOS NOMBRES CONTENIDOS EN LA TUPLA VETADOS ('Jennifer', 'Luis', 'Rady', 'Jimmy')**

[Matias, Micha, Benjamin, Oriana, Randall]



Des pues de finalizar el ingreso de datos a la lista debe mostrar un menú con las siguientes opciones:

- **BUSCAR:** Permite al usuario ingresar un nombre y buscarlo en lista, indicando si existe o no. Si existe debe indicar en que posición de la lista se encuentra.
- **MOSTRAR TODO:** Permite Mostrar Todo el contenido de la lista .
- **ORDENAR:** Ordenar la lista
- **ELIMINAR:** Pedir al usuario el ingreso de un nombre y eliminarlo de la lista
- **SALIR:** Única forma de salir de la ejecución del menú

Estructura de datos que permite almacenar su contenido en forma de llave y valor

Son **dinámicos**, pueden crecer o decrecer, se pueden añadir o eliminar elementos.

Son **indexados**, los elementos del diccionario son accesibles a través del key.

Son **anidados**, puede contener a otro diccionario en su campo value.

```
mi_diccionario = {"key1":<value1>,"key2":<value2>,"key3":<value3>,"key4":<value4>}
```

```
prueba={'Jorge':50,'Diana':60,'Rodrigo':40,'Julia':70}
```

DICCIONARIOS

El constructor dict() crea un diccionario directamente desde secuencias de pares clave-valor:

```
dict([('sape', 4139), ('guido', 4127), ('jack', 4098)])
```

```
tel=dict([('sape', 4139), ('guido', 4127), ('jack', 4098)])
```

```
tel=dict(sape=4139, guido=4127, jack=4098)
```

```
tel = {'jack': 4098, 'sape': 4139}
```

```
tel['guido'] = 4127
```

```
{'jack': 4098, 'sape': 4139, 'guido': 4127}
```

```
del tel['sape']
```

```
{'jack': 4098, 'guido': 4127}
```

```
{'jack': 4098, 'guido': 4127}
```

```
tel['irv'] = 4127
```

```
print(tel)
```

```
{'jack': 4098, 'guido': 4127, 'irv': 4127}
```

```
Print(list(tel))
```

```
['jack', 'guido', 'irv']
```

```
Print(sorted(tel))
```

```
['guido', 'irv', 'jack']
```

```
print ( 'guido' in tel)
```

```
True
```

```
print ('jack' not in tel)
```

```
False
```

```
diccionario = {'nombre' : 'Carlos', 'edad' : 22,  
               'cursos': ['Python', 'Django', 'JavaScript'] }
```

```
print (diccionario['nombre'])→ Carlos  
print (diccionario['edad'])   → 22  
print (diccionario['cursos']) → ['Python', 'Django', 'JavaScript']
```

```
print (diccionario['cursos'][0]) → Python  
print (diccionario['cursos'][1]) → Django  
print (diccionario['cursos'][2]) → JavaScript
```


```
for key in diccionario:  
    print (key, ":", diccionario[key])
```

```
nombre : Carlos  
edad   : 22  
cursos : ['Python', 'Django', 'JavaScript']
```

•zip()

Recibe como parámetro dos elementos iterables, ya sea una cadena, una lista o una tupla. Ambos parámetros deben tener el mismo número de elementos. Se devolverá un diccionario relacionando el elemento i-esimo de cada uno de los iterables.

```
dic = dict(zip('abcd',[1,2,3,4]))
```

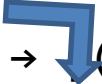


dic → {'a' : 1, 'b' : 2, 'c' : 3 , 'd' : 4}

•items()

Devuelve una lista de tuplas, cada tupla se compone de dos elementos: el primero será la clave y el segundo, su valor.

```
dic = {'a' : 1, 'b' : 2, 'c' : 3 , 'd' : 4}  
items = dic.items()
```




```
items → ('a',1),('b',2),('c',3),('d',4)]
```

•keys()

Retorna una lista de elementos, los cuales serán las claves de nuestro diccionario.

```
dic = {'a' : 1, 'b' : 2, 'c' : 3 , 'd' : 4}  
keys= dic.keys()
```



keys→ ['a','b','c','d']

•values()

Retorna una lista de elementos, que serán los valores de nuestro diccionario.

```
dic = {'a' : 1, 'b' : 2, 'c' : 3 , 'd' : 4}
```

```
values= dic.values()
```




values→ [1,2,3,4]

•clear()

Elimina todos los ítems del diccionario dejándolo vacío.

```
dic 1 = {'a' : 1, 'b' : 2, 'c' : 3 , 'd' : 4}
```

```
dic1.clean()
```



dic1 → { }

•copy()

Retorna una copia del diccionario original.

```
dic = {'a' : 1, 'b' : 2, 'c' : 3 , 'd' : 4}
```

```
dic1 = dic.copy()
```




dic1 → {'a' : 1, 'b' : 2, 'c' : 3 , 'd' : 4}

•get()

Recibe como parámetro una clave, devuelve el valor de la clave. Si no lo encuentra, devuelve un objeto none.

```
dic = {'a' : 1, 'b' : 2, 'c' : 3 , 'd' : 4}
```

```
valor = dic.get('b')
```



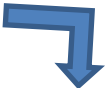
valor → 2

•pop()

Recibe como parámetro una clave, elimina esta y devuelve su valor. Si no lo encuentra, devuelve error.

```
dic = {'a' : 1, 'b' : 2, 'c' : 3 , 'd' : 4}
```

```
valor = dic.pop('b')
```



valor → 2

dic → {'a' : 1, 'c' : 3 , 'd' : 4}

```
diccionario = {"clave1": "valor1", "clave2": "valor2"}
```

```
diccionario.pop("clave3", "error")
```



error

•setdefault()

Funciona de dos formas. En la primera como get

```
dic = {'a' : 1, 'b' : 2, 'c' : 3 , 'd' : 4}
```

```
valor = dic.setdefault('a')
```

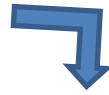


valor → 1

Y en la segunda forma, nos sirve para agregar un nuevo elemento a nuestro diccionario.

```
dic = {'a' : 1, 'b' : 2, 'c' : 3 , 'd' : 4}
```

```
valor = dic.setdefault('e',5)
```



dic → {'a' : 1, 'b' : 2, 'c' : 3 , 'd' : 4 , 'e' : 5}

•update()

Recibe como parámetro otro diccionario. Si se tienen claves iguales, actualiza el valor de la clave repetida; si no hay claves iguales, este par clave-valor es agregado al diccionario.

```
dic 1 = {'a' : 1, 'b' : 2, 'c' : 3 , 'd' : 4}
```

```
dic 2 = {'c' : 6, 'b' : 5, 'e' : 9 , 'f' : 10}
```

```
dic1.update(dic 2)
```



dic 1 → {'a' : 1, 'b' : 5, 'c' : 6 , 'd' : 4 , 'e' : 9 , 'f' : 10}

Recorriendo un Diccionario

```
# Recorremos el diccionario
for clave,valor in datos.items():
    print(clave,' ==> ',valor)
```

```
nombre ==> Marcos
apellido ==> Rojas
edad ==> 26
direccion ==> Balmaceda #1520
ciudad ==> La Serena
```

Crear un algoritmo que permita:

- Al usuario ingresar los nombres de los estudiantes de la asignatura de Programación y las notas de cada uno obtenida en la evaluación1
- almacenarlos en un diccionario llamado ESTUDIANTES.
- **NO PUEDEN SER INGRESADOS NOMBRES CONTENIDOS EN LA TUPLA VETADOS ('Jennifer', 'Luis', 'Rady','Jimmy')**

{'Matias':33, 'Magdiel':44, 'Benjamin':55, 'Oriana':66, 'Randall':20 }

Des pues de finalizar el ingreso de datos a la lista debe mostrar un menú con las siguientes opciones:

- **BUSCAR:** Permite al usuario ingresar un nombre y buscarlo , indicando si existe o no. .
- **MOSTRAR TODO:** Permite Mostrar Todo el contenido .
- **ELIMINAR:** Pedir al usuario el ingreso de un nombre y eliminarlo
- **SALIR:** Única forma de salir de la ejecución del menú