

# Estructuras de Datos

Profesor  
Sergio Gonzalez

# Unidad 2: Recursividad, Arreglos uni y multidimensionales

Profesor  
Sergio Gonzalez

# Arreglos

- Estructura de datos contiguas en memoria
- Tamaño fijo
- Conjunto de datos del mismo tipo
- Componentes = Elementos
- Modificable

# Arreglos

- Numero de elementos (tamaño) se especifica al declararlo.
- Numero de dimensiones
- Indices o subindices
  - nombreArreglo[2]

# Arreglos

- Tipos según número de dimensiones
  - Unidimensionales (vectores)
  - Bidimensionales (matrices)
  - Multidimensionales

# Arreglos en Python

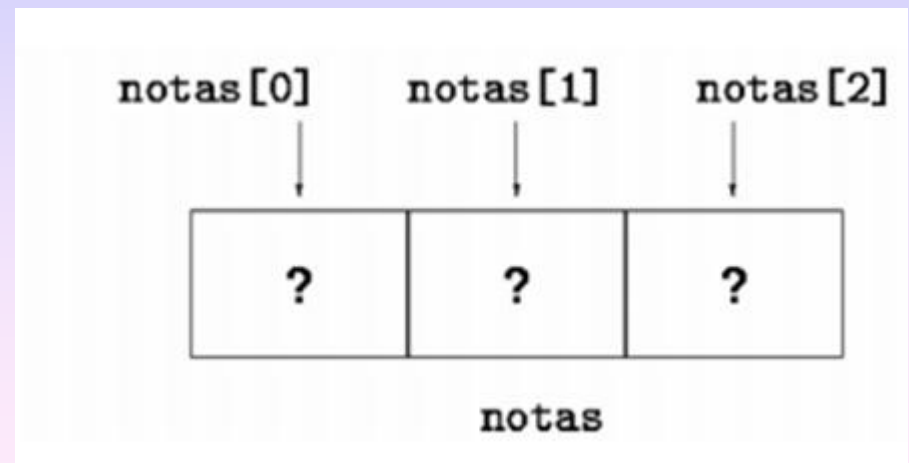
- Operaciones con arreglos
  - Tipo 'array' de paquete numpy
    - Declarar:
      - `vector = np.zeros(shape=(5),dtype=int)`
      - `vector = np.array([1 , 4 , 2 , 8 , 12])`
      - `matriz = np.zeros(shape=(5,3),dtype=int)`
      - `matriz = np.array([[1 , 4 , 2] ,[10 , 8 , 12]])`

# Arreglos en Python

- Operaciones con arreglos

- Acceder / Asignar:

- `vector[1] = 5`
    - `vector[1:4] = vector de 1, 2 y 3` (Segmento de vector)
    - `matriz[2] = Fila 2`
    - `matriz[2][1]=10`
  - `notas = np.array([10,7,8])`



# Arreglos en Python

- Operaciones con arreglos
  - Obtener tamaño:
    - `len(vector)`
    - `matriz.shape` -> Dimensiones
    - `len(matriz)` -> Numero de filas
    - `len(matriz[0])` -> Numero de columnas
  - Eliminar elemento
    - Mantener continuidad
  - Insertar elemento
    - Posición de inserción



# Arreglos en Python

- Cosas a tener en cuenta
  - Arreglos se pasan por parámetros por referencia
  - Al igualar arreglos no se crea uno nuevo
  - Cuidado con los índices negativos
  - Para recorrer:
    - for value in vector:
    - for index in range(len(vector)):
    - for reverseIndex in range(len(vector)-1,-1,-1):

# Cadenas de caracteres

- Almacenamos una palabra o frase
- Diferentes a arreglos de caracteres
  - `vc = np.zeros(shape=(5),dtype=str)`
  - `vc = np.array(['h' , 'o' , 'l' , 'a'])`
- `str` en Python
- No son modificables



# Cadenas de caracteres

- Declarar
  - `str = "palabra"`
- Concatenar
  - `str = str1 + str2 ;`
- Longitud
  - `len(str)`
- Extracción
  - `str[1]`
  - `str[inicio:fin+1]`