

Estructuras de Datos

Profesor
Sergio Gonzalez

Unidad 3: 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

Arreglos

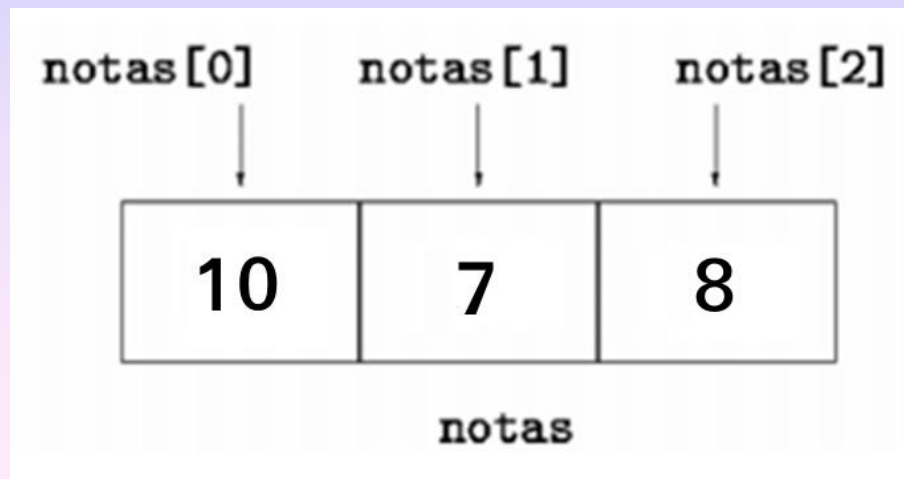
- Tipos según número de dimensiones
 - Unidimensionales (vectores)
 - Bidimensionales (matrices)
 - Multidimensionales

Arreglos en Python

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

Arreglos en Python

- Operaciones con arreglos
 - Acceder / Asignar (Usando indice / posicion):
 - `v[0]` = Primer elemento
 - `notas = np.array([10,7,8])`



Arreglos en Python

- Operaciones con arreglos
 - Acceder / Asignar (Usando indice / posicion):
 - `vector[1]` = Elemento posicion 1 (segundo)
 - `vector[1:4]` = Posiciones 1, 2 y 3
 - `matriz[2]` = Fila 2 completa
 - `matriz[2][1]` = Elemento en fila 2, columna 1

Arreglos en Python

- Operaciones con arreglos
 - Obtener tamaño:
 - `len(vector)` -> Cantidad de elementos
 - `matriz.shape` -> Dimensiones
 - `len(matriz)` -> Numero de filas
 - `len(matriz[0])` -> Numero de columnas

Arreglos en Python

- Recorrer arreglo:
 - Elementos directos
`for elemento in vector:`
 - Usando indices con range
`for indice in range(len(vector)):`
`for indice in reversed(range(len(vector))):`

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

Arreglos de caracteres vs String

- Arreglo de caracteres
 - `vc = np.zeros((5),str)`
 - `vc = np.array(['h' , 'o' , 'l' , 'a'])`
- str en Python
 - Son inmutables