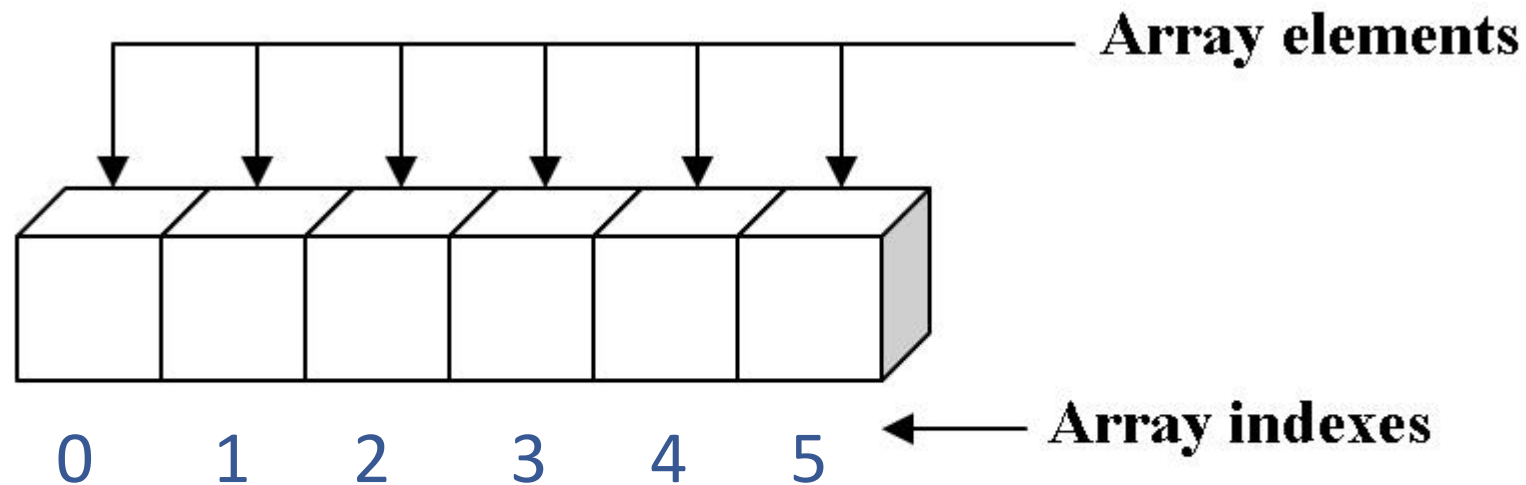


# Estructura de datos

Una estructura de datos es una forma particular de organizar información en un computador para que pueda ser utilizada de manera eficiente.



**One-dimensional array with six elements**

# Listas

- Tipo de dato utilizado para agrupar valores.
- Estos valores están separados por coma y dentro de un corchete.
- Se pueden tener valores de diferentes tipos, pero usualmente son del mismo tipo.
- Las listas se pueden indexar y segmentar.
- Son mutables, es posible cambiar su contenido (concatenar, agregar, editar, eliminar algún valor, vaciar).
- Anidación (Matrices).

Sintaxis: `lista = [ valor_1 , valor_2 , valor_3 , ... ]`

Nota: Para crear una tupla vacía se debe usar: `[]` o la función `list()`

# Tuplas

- Tipo de dato utilizado para agrupar valores.
- Estos valores están separados por coma (puede estar dentro de paréntesis).
- Se pueden tener valores de diferentes tipos, pero usualmente son del mismo tipo.
- Las tuplas se pueden indexar y segmentar.
- **NO** Son mutables, no es posible cambiar su contenido.

Sintaxis: `tupla_1 = ( valor_1 , valor_2 , valor_3 , ... )`  
`tupla_2 = valor_1 , valor_2 , valor_3 , ...`

Nota: Para crear una tupla vacía se debe usar: `()` o la función `tuple()`  
Para crear una tupla con un elemento: `tupla = "valor",`

# Conjuntos

- Tipo de dato utilizado para agrupar valores no ordenados y sin elementos repetidos.
- Estos valores están separados por coma y dentro de llaves.
- soportan operaciones matemáticas como la unión, intersección, diferencia, y diferencia simétrica.
- **No** se pueden indexar ni segmentar.
- **No** son mutables, no es posible cambiar su contenido.

Sintaxis: conjunto = { valor\_1 , valor\_2 , valor\_3 , ... }

Nota: Para crear un conjunto vacío se debe usar la función: `set()`

# Diccionario

- Conjunto de pares clave:valor. Con el requerimiento de que las claves sean únicas (dentro de un diccionario).
- Los pares clave:valor están separados por coma y dentro de llaves.
- Se pueden tener valores de diferentes tipos.
- Los diccionarios se indexan con claves (no mutables). No se puede segmentar.
- Son mutables, es posible cambiar su contenido (agregar, editar, eliminar alguna clave:valor).

Sintaxis: `diccionario = {"clave_1" : valor_1 , "clave_2" : valor_2 , "clave_3" : valor_3 , ... }`

Nota: Para crear un diccionario vacío se debe usar: `{}` o la función `dict()`

# Funciones de Listas

- **append( x )**: Agrega un ítem al final de la lista
- **extend( iterable )**: Extiende la lista agregándole todos los ítems del iterable.
- **insert( i , x )**: Inserta un ítem en una posición dada.
- **remove( x )**: Quita el prime ítem de la lista cuyo valor sea x. Lanza un ValueError si no existe tal ítem.
- **pop( [ i ] )**: Quita el ítem en la posición dada de la lista y lo retorna.
- **clear( )**: Quita el ítem en la posición dada de la lista y lo retorna.
- **index( x [ , start [ , end ] ] )**: Retorna el índice basado en cero del primer elemento cuyo valor sea igual a x. Lanza una excepción ValueError si no existe tal elemento.

# Funciones de Listas

- **count( x )**: Retorna el número de veces que x aparece en la lista.
- **sort( \*, key = None, reverse = False)**: Ordena los elementos de la lista in situ (los argumentos pueden ser usados para personalizar el orden de la lista).
- **reverse( )**: Invierte el orden de la lista.
- **copy( )**: Retorna una copia superficial de la lista. Equivalente a [ : ].

# Técnicas de iteración

- **items()**: obtenemos la clave y su respectivo valor. [ **diccionarios** ]
- **enumerate()**: obtenemos el índice posición junto a su valor correspondiente. [ **listas , tuplas , conjuntos** ]
- **zip()**: iterar sobre dos o más secuencias al mismo tiempo. [ **listas , tuplas , conjuntos , diccionarios** ]
- **reversed()**: iterar sobre una secuencia en orden inverso. [ **listas , tuplas , diccionarios** ]
- **sorted()**: itera sobre una secuencia ordenada. [ **listas , tuplas , conjuntos , diccionarios** ]
- **range()**: itera sobre una secuencia de números. [ **listas , tuplas , conjuntos , diccionarios** ]