**Documentation:**

**Array** - In computer science, an **array data structure**, or simply an **array**, is a data structure consisting of a collection of *elements* (values or variables), each identified by at least one *array index* or *key*. In simple words, an array is a data structure that stores values of the same data type. An array is stored such that the position of each element can be computed from its index tuple by a mathematical formula. The simplest type of data structure is a linear array, also called a one-dimensional array.

**Elements** - Each item in an array is known as Element

**Index** - Each location in an array has a numerical index which is used to find the element

**Structure of an Array:**

Elements   [12, 64, 92, 72, 3, 20, 32, 7, 12, 28, 100]

Index           0     1    2    3     4     5 6 7 8 9 10

* By default, the index starts at 0
* Here the length of the array is 6, i.e. its storing 6 elements.

**Basic Array operations:**

**Traverse** - Print all elements of an array

**Insert** - Add an element in the array at the given index

**Update** - Updates an element in the array at the given index

**Search**- Searches an element at the given index

**Delete** - Deletes an element at the given index

In this project, we were working on two arrays and the following tasks were being performed without built-in functions:

* create
* size
* insert
* update
* delete
* traverse
* sort

Advanced Array Function:

|  |  |
| --- | --- |
| **Method** | **Description** |
| [append(](https://www.w3schools.com/python/ref_list_append.asp))\_ | Adds an element at the end of the list |
| [clear()](https://www.w3schools.com/python/ref_list_clear.asp) | Removes all the elements from the list |
| [copy()](https://www.w3schools.com/python/ref_list_copy.asp) | Returns a copy of the list |
| [count()](https://www.w3schools.com/python/ref_list_count.asp) | Returns the number of elements with the specified value |
| [extend()](https://www.w3schools.com/python/ref_list_extend.asp) | Add the elements of a list, to the end of the current list |
| [index()](https://www.w3schools.com/python/ref_list_index.asp) | Returns the index of the first element with the specified value |
| [insert()](https://www.w3schools.com/python/ref_list_insert.asp) | Adds an element at the specified position |
| [pop()](https://www.w3schools.com/python/ref_list_pop.asp) | Removes the element at the specified position |
| [remove()](https://www.w3schools.com/python/ref_list_remove.asp) | Removes the first item with the specified value |
| [reverse()](https://www.w3schools.com/python/ref_list_reverse.asp) | Reverses the order of the list |
| [sort()](https://www.w3schools.com/python/ref_list_sort.asp) | Sorts the list |

**Array Operations in Action:**

**Importing Module:**

To use perform operations on arrays we imported “\*” from the array module and the “re” (Regular Expression) module.

from array import \*

import re

**Creating the Array:**

def CreateArray(choice):

if choice == 1:

global filledIndexArray1, isArray1Full, isArray1Empty, filledIndexArray1, isArray1Created

filledIndexArray1 = 0

isArray1Full = False

isArray1Empty = True

isArray1Created = True

print("Array 1 created ")

print("-------------------")

**Size of an Array:**

**def SizeOfArray(choice):**

**if choice == 1:**

**if isArray1Created:**

**choice = userInput("Enter size of array between 1 to 50: ")**

**if choice > 50 or choice < 1:**

**print("Enter valid size for array")**

**print("----------------------")**

**else:**

**global sizeOfArray1, filledIndexArray1, emptyIndexArray1**

**sizeOfArray1 = choice**

**filledIndexArray1 = 0**

**emptyIndexArray1 = choice**

**print(f"size of array 1 is decleared to be: {sizeOfArray1}",)**

**else:**

**print("Array 1 is not created")**

**print("----------------------")**

**Insert an Element:**

**def insertElement(choice):**

**global arr1, emptyIndexArray1, filledIndexArray1, isArray1Empty**

**if choice == 1:**

**if isArray1Created and sizeOfArray1 > 0:**

**choices = userInput("how many values do you want to insert? ")**

**if choices <= emptyIndexArray1:**

**isArray1Empty = False**

**insertMenu()**

**choice = userInput("enter your choice ")**

**if choice == 1:**

**if filledIndexArray1 > 0:**

**for i in range(0, choices):**

**elementAlreadyInserted()**

**else:**

**insertElementAtFirstIndex(choices)**

**elif choices == 2:**

**insertAtLastIndex(choices)**

**print("data Inserted")**

**print("-------------")**

**elif choices == 3:**

**startingindex = userInput(**

**"At what index you want to add elements: ")**

**if filledIndexArray1 > 0:**

**for i in range(startingindex, choices):**

**elementAlreadyInserted()**

**else:**

**insertElementAfterCertainIndex(choices)**

**else:**

**print("Wrong choice")**

**print("-------------")**

**menu()**

**else:**

**print("Either array not created and memory size is not defined.")**

**Traverse:**

**def traverse(choice):**

**if choice == 1:**

**if isArray1Created:**

**showinfo(arr1, sizeOfArray1,**

**emptyIndexArray1, filledIndexArray1)**

**else:**

**print("Create array first to view")**

**elif choice == 2:**

**if isArray2Created:**

**showinfo(arr2, sizeOfArray2,**

**emptyIndexArray2, filledIndexArray2)**

**else:**

**print("Choose between 1 and 2 array only")**

**def showinfo(arr, sizeOfArray, emptyIndexArray, filledIndexArray):**

**print(f"size of array is {sizeOfArray}")**

**print(f"empty spaces of array is {emptyIndexArray}")**

**print(f"filled spaces of array is {filledIndexArray}")**

**for i in range(0, filledIndexArray):**

**print(f"{arr[i] }", end=" "),**

**Delete:**

**def DeleteArray(choice):**

**global isArray1Created, arr1**

**global isArray2Created, arr2**

**if choice == 1:**

**if isArray1Created:**

**del arr1**

**print('Array 1 is removed successfully')**

**print("----------")**

**elif choice == 2:**

**if isArray2Created:**

**del arr2**

**print('Array 2 is removed successfully')**

**print("----------")**

**else:**

**print("Array does not exist")**

**print("----------")**

**Sort:**

Sorting include Bubble Sort and Merge Sort by pressing 7 we get a choice to get our array sorted by bubble method or merge in ascending order.

**def sort(choice):**

**if choice == BubbleSort:**

**BubbleSort()**

**elif choice == MergeSort:**

**MergeSort()**

**def bubbleSort(arr):**

**n = len(arr)**

**for i in range(n):**

**for j in range(0, n-i-1):**

**if arr[j] > arr[j+1]:**

**arr[j], arr[j+1] = arr[j+1], arr[j]**

**print("Sorted array is:")**

**for i in range(len(arr)):**

**print("%d" % arr[i], end=" ")**

**def mergeSort(arr):**

**if len(arr) > 1:**

**mid = len(arr)//2**

**L = arr[:mid]**

**R = arr[mid:]**

**mergeSort(L)**

**mergeSort(R)**

**i = j = k = 0**

**while i < len(L) and j < len(R):**

**if L[i] < R[j]:**

**arr[k] = L[i]**

**i += 1**

**else:**

**arr[k] = R[j]**

**j += 1**

**k += 1**

**while i < len(L):**

**arr[k] = L[i]**

**i += 1**

**k += 1**

**while j < len(R):**

**arr[k] = R[j]**

**j += 1**

**k += 1**

**for i in range(len(arr)):**

**print("%d" % arr[i], end=" ")**