1-D Array

- Stores a list of variables of the **same data type**

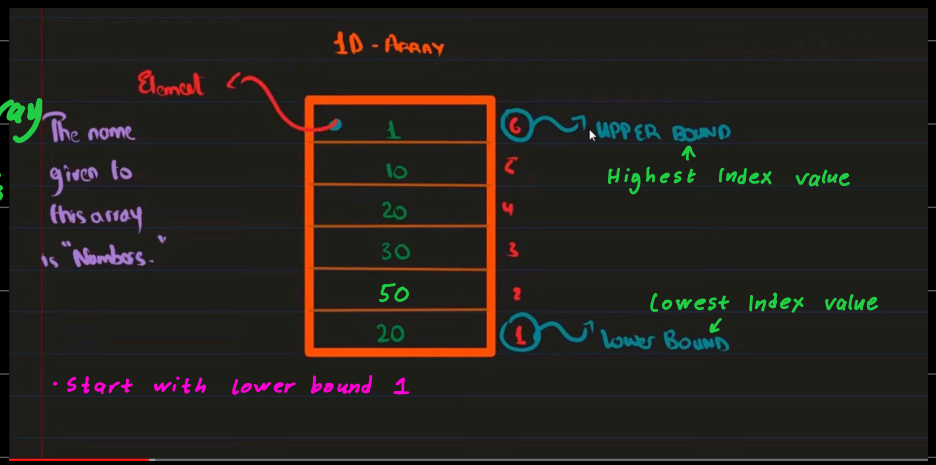
- Stores multiple values
- Same data type

Student 1 = "Ahmar Khan"
Student 2 = "Abdullah Ashfaq" — String
Student 3 = "Sheriyar"
Student 4 = "Hashim"
Student 5 = "Emad"
...
Student 30 = "..."

Solution Arrays
- Removes declaration and making of variables.
- Time-saving
↓
Efficient

Q- Difference b/w Element and index.

- Each data item in array is element and elements position is known as index.



Storing Values

Variable ↓ Position / Index ↓

Numbers [2] ← 50 → Value to be stored

- Must use square brackets '[ ]'

Declaration

DECLARE Name of Array : Array [Lowerbound : Upperbound] OF Datatype

- DECLARE Numbers : ARRAY [1 : 6] OF INTEGER

Q- write a pseudocode of a program in which input a number and store it in an array, position 2 (array name is Numbers)

```
DECLARE Numbers: ARRAY [1 : 6] OF INTEGER
DECLARE Num: INTEGER
```

```
OUTPUT "Enter Your Number : "
INPUT Num                          ]      INPUT Numbers[2]
Numbers[2] ← Num                   ]
```

Note: For storing a value always use FOR ... Loop.

Q- Construct an array of 565 elements and store names in it which user will input.

```
DECLARE Names: ARRAY[1:565] OF STRING
DECLARE Index : INTEGER

FOR Index ← 1 TO 565
    OUTPUT "Enter Name : "
    INPUT Names[Index]
END FOR
```

Q- Construct an array of 100 elements and store "No data" in all elements (Name of array= ResultArray)

```
DECLARE Index : INTEGER
DECLARE ResultArray : ARRAY [1:100] OF STRING

FOR Index ← 1 TO 100
    ResultArray [Index] ← "No data"
END FOR
```

Searching in an array

- Note: For searching in an array, you need to use selection statements.

Q- There is an array which contains names of 500 students search at which position "AHMAR" is stored. (N.O.A = 'Names')

```
DECLARE Names: ARRAY[1:500] OF STRING
DECLARE Index: INTEGER

  FOR Index ← 1 TO 500
      IF Names[Index] = "AHMAR"
          THEN
              OUTPUT "'AHMAR' is stored at Position:", Index
      ENDIF
  ENDFOR
```

Q- Already there is an array of name "SearchBox" with 500 elements of data type string. Search through the array and find how many times "Empty" was repeated.

```
DECLARE Index, Times : INTEGER
DECLARE SearchBox : ARRAY[1:500] OF STRING

Times ← 0
```

```
FOR Index ← 1 TO 500
    IF SearchBox[Index] = "Empty"
        THEN
            Times = Times + 1
    END IF
ENDFOR
OUTPUT "Number of times 'Empty' was repeated:", Times
```

Q- There is an array named "Numbers" of 500 elements, search through the array and find how many numbers are positive and how many are negative?

```
DECLARE Numbers: ARRAY[1:500] OF INTEGER
DECLARE Positive, Negative, Index : INTEGER
Positive ← 0
Negative ← 0
FOR Index ← 1 TO 500
```

```
    IF  Numbers[Index] >0
        THEN
            Positive ← Positive +1
        ELSE IF   Numbers[Index] < 0
                THEN
                    Negative ← Negative + 1
                END IF
    END IF
END FOR
OUTPUT "Number of Positive values: ", Positive
OUTPUT "Number of negative values: ", Negative
```

Index: A numerical indicator of data's position in an array.

Lower bound: The index of the first element in an array.

Upperbound: The index of the last element in an array.

Array is a data structure + data type