An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together. This makes it easier to calculate the position of each element by simply adding an offset to a base value, i.e., the memory location of the first element of the array (generally denoted by the name of the array). The base value is index 0 and the difference between the two indexes is the offset.

For simplicity, we can think of an array as a fleet of stairs where on each step is placed a value (let's say one of your friends). Here, you can identify the location of any of your friends by simply knowing the count of the step they are on. Remember: "Location of next index depends on the data type we use".

```
                    MEMORY lOCATION
                    ---------------
      20   21   22   23   24   25   26   27   28  29   30
     ['a']['b']['c']['d']['e']['f']['g']['h']['i']['j']['k']
      0    1    2    3    4    5    6    7    8   9    10
```

The above image can be looked at as a top-level view of a staircase where you are at the base of the staircase. Each element can be uniquely identified by its index in the array (in a similar way as you could identify your friends by the step on which they were on in the above example).

Advantages of using arrays:
----

1) Arrays allow random access to elements. This makes accessing elements by position faster.
2) Arrays have better cache locality that makes a pretty big difference in performance.
3) Arrays represent multiple data items of the same type using a single name.

Disadvantages of using arrays:

You can't change the size i.e. once you have declared the array you can't change its size because of static memory allocation. Here Insertion(s) and deletion(s) are difficult as the elements are stored in consecutive memory locations and the shifting operation is costly too.
Now if take an example of implementation of data structure Stack using array there are some obvious flaw.
Let's take the POP operation of the stack. The algorithm would go something like this.

    1) Check for the stack underflow
    2) Decrement the top by 1

So there what we are doing is that, the pointer to the topmost element is decremented means we are just bounding our view actually that element stays there talking up of the memory space. If you have any primitive datatype then it might be ok but the object of an array would take a lot of memory.

Applications on Array:
----

1) Array stores data elements of the same data type.
2) Arrays can be used for CPU scheduling.

3) Used to Implement other data structures like Stacks, Queues, Heaps, Hash tables, etc.