# Array:

## Properties:

* Continuous
* Fixed
* Same data type
* Ranging from 0 to n-1

## Memory of an array:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Advantage and disadvantages:

Advantages:

* Used for fixed variable operation
* Easy to create
* Easy to operate, access and update

Disadvantage:

* Use same data type.
* High space usage.
* Cannot update the size of array.

# Types of arrays:

1D, 2D(Matrix), 3D (Rubik cube)

## Creation of 1D array:

Creating of memory and values

## Accessing an array:

Accessing the value and its types

## Updating an array:

Updating the value and replacing the value

## How memory calculated for 1D array:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  | a[0] | a[1] | a[2] | a[3] | a[4] |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## Time for searching:

Insertion – O(1)

Update- O(1)

Searching- O(n)

# Creation of 2d array:

## Operations and time complexity:

**In-correct way:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a[0][0] | a[0][1] | a[1][0] | a[1][1] | a[2][0] |  |
|  | a[2][1] | a[3][0] | a[3][1] | a[4][0] | a[4][1] |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

**Correct way:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a[0][0] | a[0][1] | a[1][0] | a[1][1] | a[2][0] | a[2][1] |
| a[3][0] | a[3][1] | a[4][0] | a[4][1] |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

## Time for searching:

Insert - 0(1) Update - 0(1) Traverse - 0(m\*n) Searching – 0(m\*n)

## When to use and when not to use.