Arrays and ArrayList in Java.

① → Need → To store multiple values, if don't have to initialize variables again & again.

② → For dynamic Array. (Arraylist).

② → Syntax → `int[] Name = new int[size];`

(Stack) Declaration — creation of obj, memory allocation in heap.

* All datatype is same in an array.

- `int[]` → datatype of ele inside array.
- Name → ref variable/Name
- new → to create an obj
- `int[size]` → no: of elements.

③ → Declaration happens at compile time
Memory allocation happens at runtime.

→ WKT, Stack → local var & methods
heap → inst var & objects.

→ As there is no concept of pointers in C, C++, in Java an array does not have continuous memory allocation. i.e depends on JVM.

④ Null → special literal, can only be assigned to non-primitive datatypes. (Assigned to nothing)

⑤ Input →

```
for(----){
    arr[i] = sc.nextInt();
}
```

Taken as
1 2 3 4 5
Not
1
2
3
4
5

⑥ → `Arrays.toString(arr)` → converts arr to String. (uses StringBuilder in internal working)

`swap(arr, i, j);`

`Arrays.sort()` → For sorting the array.

⑦ Strings are immutable but arrays are mutable.

⑦ Multidimensional Arrays (2D or matrix).

→ `int[][] arr = new int[3][3];` (Arrays of Arrays)

→ Size of arrays is variable } optional.
i.e column is optional }

Input → 2 for loops used. (in second loop condition is `i < arr[row].length;`)

Output → same steps i.e 2 for loops.
↳ one line to be added in outer for.

---

⑧ Dynamic Arrays

→ Arraylist is used in java as dynamic array.

```
for(int[] a : arr)
{ Sy...(a);}
```
↳ one more way.

→ Syntax → `ArrayList<Integer> list1 = new ArrayList<>();`
(can take a initial capacity)
Initializes a constructor.

→ One of the class of collections framework.

→ `list1.add(x);`
`System.out.println(list1);` } for loop not required.

→ `list1.contains(x);` → true/false

→ `list1.set(index of x, y);` .remove, .clear().

→ `arr[0]` means → `list1.get(0)`

→ now dynamic → doubles when reaches the end. [O(1)]
(1 → 2 → 4 → 8 → 16 → 32 - - - - -)
& copies old list to the new longer list.

⑨

→ `ArrayList<ArrayList<Integer>>` → for 2D Arrays. Others same.

Arrays finished.

For 2D Dynamic array input.

```
for(---){
  for(----){
    list1.get(i).add(sc.nextInt);
  }
}
```