জাভাতে **অ্যারে (Array)** হল একই টাইপের একাধিক ডেটা স্টোর করার জন্য একটি ডেটা স্ট্রাকচার। অ্যারে ব্যবহার করে একই নামের অধীনে একাধিক ভ্যালু স্টোর করা যায় এবং ইনডেক্স (Index) ব্যবহার করে সেই ভ্যালুগুলো অ্যাক্সেস করা যায়। অ্যারে ফিক্সড সাইজের হয়, অর্থাৎ একবার সাইজ ডিফাইন করলে তা পরিবর্তন করা যায় না।

**অ্যারে ডিক্লারেশন এবং ইনিশিয়ালাইজেশন:**

জাভাতে অ্যারে ডিক্লার করার দুটি উপায় রয়েছে:

1. **অ্যারে ডিক্লারেশন এবং পরে ইনিশিয়ালাইজেশন**:

java

Copy

data\_type[] array\_name; // অ্যারে ডিক্লারেশন

array\_name = new data\_type[size]; // ইনিশিয়ালাইজেশন

উদাহরণ:

java

Copy

int[] numbers; // অ্যারে ডিক্লারেশন

numbers = new int[5]; // ইনিশিয়ালাইজেশন (সাইজ 5)

1. **অ্যারে ডিক্লারেশন এবং ইনিশিয়ালাইজেশন একসাথে**:

java

Copy

data\_type[] array\_name = {value1, value2, value3, ...};

উদাহরণ:

java

Copy

int[] numbers = {1, 2, 3, 4, 5}; // অ্যারে ডিক্লারেশন এবং ইনিশিয়ালাইজেশন

**অ্যারে অ্যাক্সেস করা:**

অ্যারের ভ্যালু অ্যাক্সেস করতে ইনডেক্স ব্যবহার করা হয়। ইনডেক্স 0 থেকে শুরু হয় এবং array\_length - 1 পর্যন্ত হয়।

java

Copy

int[] numbers = {10, 20, 30, 40, 50};

System.out.println(numbers[0]); // আউটপুট: 10

System.out.println(numbers[2]); // আউটপুট: 30

**অ্যারে মেথড এবং অপারেশন:**

জাভাতে অ্যারে নিয়ে কাজ করার জন্য কিছু বিল্ট-ইন মেথড এবং অপারেশন রয়েছে। নিচে সেগুলো ব্যাখ্যা করা হলো:

**১. অ্যারের লেন্থ (Length):**

অ্যারের লেন্থ বের করতে length প্রপার্টি ব্যবহার করা হয়।

java

Copy

int[] numbers = {1, 2, 3, 4, 5};

System.out.println(numbers.length); // আউটপুট: 5

**২. অ্যারে ট্রাভার্স (Traverse):**

অ্যারের সব ভ্যালু অ্যাক্সেস করতে লুপ ব্যবহার করা হয়।

java

Copy

int[] numbers = {10, 20, 30, 40, 50};

for (int i = 0; i < numbers.length; i++) {

System.out.println(numbers[i]);

}

**৩. অ্যারে সর্ট (Sort):**

অ্যারে সর্ট করতে Arrays.sort() মেথড ব্যবহার করা হয়।

java

Copy

import java.util.Arrays;

int[] numbers = {5, 3, 1, 4, 2};

Arrays.sort(numbers); // অ্যারে সর্ট করা

System.out.println(Arrays.toString(numbers)); // আউটপুট: [1, 2, 3, 4, 5]

**৪. অ্যারে কপি (Copy):**

একটি অ্যারে থেকে অন্য অ্যারেতে ভ্যালু কপি করতে Arrays.copyOf() বা System.arraycopy() ব্যবহার করা হয়।

java

Copy

import java.util.Arrays;

int[] numbers = {1, 2, 3, 4, 5};

int[] copyNumbers = Arrays.copyOf(numbers, numbers.length); // অ্যারে কপি করা

System.out.println(Arrays.toString(copyNumbers)); // আউটপুট: [1, 2, 3, 4, 5]

**৫. অ্যারে সার্চ (Search):**

অ্যারে থেকে কোনো ভ্যালু সার্চ করতে Arrays.binarySearch() মেথড ব্যবহার করা হয়। এই মেথড ব্যবহার করার আগে অ্যারেটি সর্টেড থাকতে হবে।

java

Copy

import java.util.Arrays;

int[] numbers = {1, 2, 3, 4, 5};

int index = Arrays.binarySearch(numbers, 3); // ভ্যালু 3 এর ইনডেক্স সার্চ করা

System.out.println(index); // আউটপুট: 2

**৬. অ্যারে ফিল (Fill):**

অ্যারের সব ভ্যালু একই ভ্যালু দিয়ে ফিল করতে Arrays.fill() মেথড ব্যবহার করা হয়।

java

Copy

import java.util.Arrays;

int[] numbers = new int[5];

Arrays.fill(numbers, 10); // অ্যারের সব ভ্যালু 10 দিয়ে ফিল করা

System.out.println(Arrays.toString(numbers)); // আউটপুট: [10, 10, 10, 10, 10]

**৭. অ্যারে তুলনা (Compare):**

দুটি অ্যারে তুলনা করতে Arrays.equals() মেথড ব্যবহার করা হয়।

java

Copy

import java.util.Arrays;

int[] numbers1 = {1, 2, 3};

int[] numbers2 = {1, 2, 3};

boolean isEqual = Arrays.equals(numbers1, numbers2); // অ্যারে তুলনা করা

System.out.println(isEqual); // আউটপুট: true

**৮. অ্যারে টু স্ট্রিং (Array to String):**

অ্যারে কন্টেন্ট প্রিন্ট করতে Arrays.toString() মেথড ব্যবহার করা হয়।

java

Copy

import java.util.Arrays;

int[] numbers = {1, 2, 3, 4, 5};

System.out.println(Arrays.toString(numbers)); // আউটপুট: [1, 2, 3, 4, 5]

**মাল্টি-ডাইমেনশনাল অ্যারে (Multi-Dimensional Array):**

জাভাতে মাল্টি-ডাইমেনশনাল অ্যারে (যেমন: 2D অ্যারে, 3D অ্যারে) ব্যবহার করা যায়। এটি টেবিল বা ম্যাট্রিক্সের মতো ডেটা স্টোর করতে সাহায্য করে।

**উদাহরণ:**

java

Copy

int[][] matrix = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

System.out.println(matrix[1][2]); // আউটপুট: 6 (2nd row, 3rd column)

**অ্যারের সুবিধা:**

1. একই টাইপের একাধিক ডেটা স্টোর করা যায়।
2. ইনডেক্স ব্যবহার করে দ্রুত ডেটা অ্যাক্সেস করা যায়।
3. অ্যারে সর্ট, সার্চ, কপি ইত্যাদি অপারেশন সহজে করা যায়।

**অ্যারের সীমাবদ্ধতা:**

1. অ্যারের সাইজ ফিক্সড, অর্থাৎ রানটাইমে সাইজ পরিবর্তন করা যায় না।
2. মেমোরি অপ্টিমাইজেশন না হলে মেমোরি অপচয় হতে পারে।

অ্যারে শেখা খুবই গুরুত্বপূর্ণ, কারণ এটি ডেটা স্ট্রাকচার এবং অ্যালগরিদমের বেসিক কনসেপ্ট। নিয়মিত অনুশীলন করলে অ্যারে ব্যবহার সহজেই আয়ত্ত করা যায়। 😊