



Explore | Expand | Enrich



ARRAY'S IN JAVA



- An array is a container object that holds a fixed number of values of a single type
- The length of an array is established when the array is created
- In fact most of the collection types in Java which are the part of java.util package use arrays internally in their functioning
- Since Arrays are objects, they are created during runtime .The array length is fixed



FEATURES OF ARRAYS

- They can even hold the reference variables of other objects
- They are created during runtime
- They are dynamic, created on the heap
- The Array length is fixed



ARRAY DECLARATION

The preceding program declares an array (named anArray) with the following line of code

```
int[] anArray;
```

- An array's type is written as `type[]`, where `type` is the data type of the contained elements
- the brackets are special symbols indicating that this variable holds an array
- The size of the array is not part of its type (which is why the brackets are empty)



CREATING, INITIALIZING, AND ACCESSING AN ARRAY

One way to create an array is with the new operator

```
arrayRefVar = new dataType[arraySize];
```

- It creates an array using new dataType[arraySize]
- It assigns the reference of the newly created array to the variable arrayRefVar



CREATING, INITIALIZING, AND ACCESSING AN ARRAY

- If this statement is missing, then the compiler prints an error like the following, and compilation fails

```
anArray = new int[10]; // create an array of integers
```

- ArrayDemo.java:4: Variable anArray may not have been initialized



INITIALIZING AN ARRAY

The next few lines assign values to each element of the array

```
anArray[0] = 100;  
// initialize first element  
anArray[1] = 200;  
// initialize second element  
anArray[2] = 300; // and so forth
```



ACCESSING AN ARRAY

Each array element is accessed by its numerical index

```
System.out.println("Element 1 at index 0: " + anArray[0]);  
System.out.println("Element 2 at index 1: " + anArray[1]);  
System.out.println("Element 3 at index 2: " + anArray[2]);
```



ACCESSING AN ARRAY

Each array element is accessed by its numerical index

```
System.out.println("Element 1 at index 0: " + anArray[0]);  
System.out.println("Element 2 at index 1: " + anArray[1]);  
System.out.println("Element 3 at index 2: " + anArray[2]);
```



ARRAY INITIALIZATION -2

```
int[] age = {12, 4, 5, 2, 5};
```

- This statement creates an array and initializes it during declaration
- The length of the array is determined by the number of values provided which is separated by commas. In our example, the length of age array is 5



EXAMPLE TO INITIALIZE AND PRINTING THE ARRAY

```
class Main {  
    public static void main(String[] args) {  
        int[] age = {12, 4, 5, 2, 5};  
        for (int i = 0; i < 5; ++i) {  
            System.out.println("Element at index " + i + ": " + age[i]);  
        }  
    }  
}
```

```
Element at index 0 :12  
Element at index 1: 14  
Element at index 2: 5
```



THANK YOU

