

Think and Tell

A sports academy wishes to analyze the performance of 11 players of a Soccer team on the basis of their age.

How do you think a programmer will store data containing the age details of all the 11 athletes?



Ages of Players: Solution 1



Is this an ideal solution?

Can we declare 11 different variables with unique names?
`int ageOfPlayer1, ageOfPlayer2, ageOfPlayer3, ... ageOfPlayer11`

Ages of Players: Solution 2



Declare an array with 11 items indexed from 0 to 10:
Int ageOfPlayer[] = new int[11]

Working with Single Dimensional Arrays



Learning Objectives

- Define arrays
- List the different types of arrays
- Create single dimensional arrays
- Assign values to single dimensional arrays
- Traverse and manipulate single dimensional arrays



Interactive Demo

Write a program to accept and display the ages of the players of a soccer team.



Age Group of the Soccer Players

Average age group of the soccer players

ageOfPlayer	18	20	21	20	21	18	19	22	20	21	18	19	22
-------------	----	----	----	----	----	----	----	----	----	----	----	----	----

What Is an Array?

- An array is a collection of homogenous elements of a single data type, stored in adjacent memory locations
- It is a static data structure
- It can be accessed by specifying the name and the subscript number of the array
- The subscript number:
 - Specifies the position of an element within the array
 - It is also called the index of the element

Types of Arrays

- **Single dimensional array:**

- A collection of elements with a single index value
- It can have multiple columns but only one row

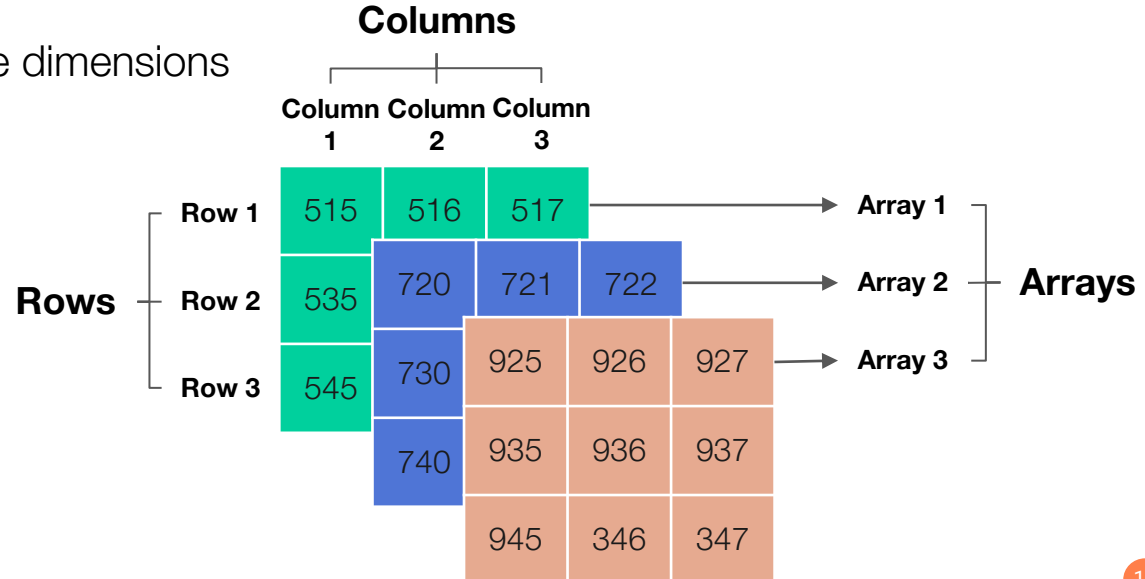
	Index 0	Index 1	Index 2	Index 3	Index 4
athleteName	Jessica	Allyson	Carl	Lebron	Serena
	athleteName[0]	athleteName[1]	athleteName[2]	athleteName[3]	athleteName[4]

	Index 0	Index 1	Index 2	Index 3	Index 4
athleteAge	16	17	16	20	21
	athleteAge[0]	athleteAge[1]	athleteAge[2]	athleteAge[3]	athleteAge[4]

Types of Arrays contd.

- Multidimensional array:
 - A collection of elements with a multi-index value
 - It can have multiple rows and columns
 - It can have two or more dimensions

Country Name	Gold	Silver	Bronze
USA	46	28	29
CHINA	38	28	22
INDIA	29	17	19



Creating Arrays

- **Creating a one-dimensional array:**

- Declare an array
- Assign values to an array

- **Declaring a one-dimensional array:**

```
arraytype arrayname[] = new arraytype[size] ;
```

- **Declaring an array to store 10 string values:**

```
String athleteName[] = new String[5];
```

```
int athleteAge[] = new int[5];
```

Assigning Values to the Array Elements

- Use index numbers to assign values to different elements of an array:

	Index 0	Index 1	Index 2	Index 3	Index 4
athleteName	Jessica	Allyson	Carl	Lebron	Serena
	athleteName[0]	athleteName[1]	athleteName[2]	athleteName[3]	athleteName[4]

	Index 0	Index 1	Index 2	Index 3	Index 4
athleteAge	16	17	16	20	21
	athleteAge[0]	athleteAge[1]	athleteAge[2]	athleteAge[3]	athleteAge[4]

athleteAge[0]	=	16
athleteAge[1]	=	17
athleteAge[2]	=	16
athleteAge[3]	=	20
athleteAge[4]	=	21

athleteName[0]	=	"Jessica"
athleteName[1]	=	"Allyson"
athleteName[2]	=	"Carl"
athleteName[3]	=	"Lebron"
athleteName[4]	=	"Serena"

Assigning Values to the Array Elements contd.

- Assign values to the elements of an array at the time of declaration:

```
String athleteName[] ={"Jessica","Allyson","Carl","Lebron","Serena"};
```

	Index 0	Index 1	Index 2	Index 3	Index 4
athleteName	Jessica	Allyson	Carl	Lebron	Serena
	athleteName[0]	athleteName[1]	athleteName[2]	athleteName[3]	athleteName[4]

```
int athleteAge[] ={16,17,16,20,21};
```

	Index 0	Index 1	Index 2	Index 3	Index 4
athleteAge	16	17	16	20	21
	athleteAge[0]	athleteAge[1]	athleteAge[2]	athleteAge[3]	athleteAge[4]

Interactive Demo

Write a program to accept the name and age of all the players of a soccer team and represent them in the memory using two different single dimensional arrays.

Task 1: Create arrays

Task 2: Assign values to the arrays



Traversing Array Elements

How do we traverse through the elements of an array?



Accessing Arrays

- **Syntax to access a single dimensional array:**

```
arrayname[index];
```

- **Use the *for* loop to display the elements of an array:**

```
String athleteName[] ={"Jessicca","Allyson","Carl","Lebron","  
Serena"};
```

```
for(int i=0;i<5;i++){  
    System.out.println(athleteName[i]);}
```


Accessing Arrays

When we declare an array to store different values, all the elements in that array may not be filled up with values.

How can we handle this situation while traversing through an array?



Accessing Arrays contd.

- Traversing through an array using the *for* loop and the length property:

```
String athleteName[] ={"Jessica","Allyson","Carl","Lebron","Serena"};
```

```
    for(int i=0;i< athleteName.length;i++){  
        System.out.println(athleteName[i]);}
```

Accessing Arrays Using for-each

- **The for-each loop of Java:**
 - Is used to iterate through an array
 - Helps in simplifying the code and increasing its readability
- **The syntax of the for-each loop used in an array is:**

```
for(type var: arrayobject)
```

Accessing Arrays Using for-each contd.

- **Display the elements of an array using the for-each loop:**

```
String athleteName[] ={"Jessica","Allyson","Carl","Lebron","  
Serena"};  
  
    for(int athleteAge:athleteAge){  
        System.out.println(athleteAge);}
```

Interactive Demo

Write a program that accepts the age of 11 players of a soccer team and displays the total number of players within the age group of 18 to 20.



Key Takeaways

- Different types of arrays
- Create single dimensional arrays
- Assign values to arrays
- Traverse through single dimensional arrays
- Access Arrays using the for-each loop





Thank you!