

Think and Tell

Medal tally of the top five winning countries of 2016 Olympics has been displayed.

TOP COUNTRIES

RANK	COUNTRY	G	S	B	TOTAL
1	 USA United States	46	37	38	121
2	 CHN China	26	18	26	70
3	 GBR Great Britain	27	23	17	67
4	 RUS Russian Federation	19	18	19	56
5	 GER Germany	17	10	15	42

Source: https://en.wikipedia.org/wiki/2016_Summer_Olympics_medal_table



Medal Tally

TOP COUNTRIES						
RANK		COUNTRY				TOTAL
1		USA United States	46	37	38	121
2		CHN China	26	18	26	70
3		GBR Great Britain	27	23	17	67
4		RUS Russian Federation	19	18	19	56
5		GER Germany	17	10	15	42

How can you represent the number of Gold, Silver, and Bronze medals won by each of these countries in the memory as part of a program?

Medal Tally: Solution

G	S	B
46	37	38
26	18	26
27	23	17
19	18	19
17	10	15

Will single dimensional arrays work efficiently in this scenario?

```
int [] gold = {46,26,27,19,17};
int [] silver = {37,23,18,17,10};
int [] bronze = {38,17,26,20,15};
```

Medal Tally: Solution

```
int [] gold = {46,26,27,19,17};
```

```
int [] silver = {37,23,18,17,10};
```

```
int [] bronze = {38,17,26,20,15};
```

Can we represent the distribution of medals by using just one array instead of three different arrays?

Working with Multidimensional Arrays





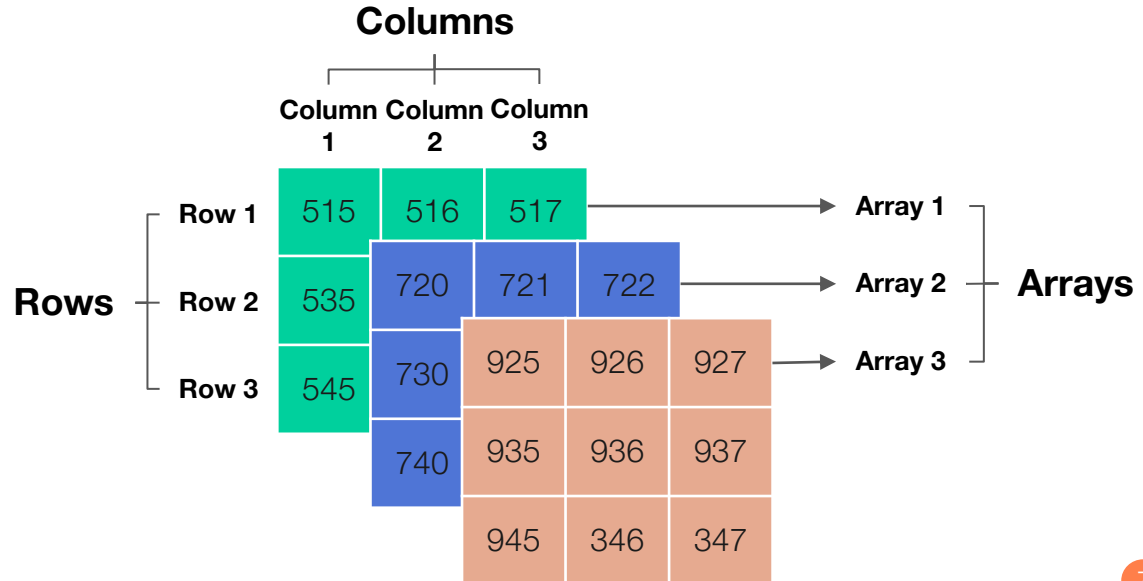
Learning Objectives

- Declare multidimensional arrays
- Initialize multidimensional arrays
- Iterate through multidimensional arrays
- Manipulate multidimensional arrays

Multidimensional Array

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

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



Creating Multidimensional Arrays

How can we create a multidimensional array to store data of the medals won by different countries?



Creating Multidimensional Arrays contd.

- **Steps to create a multidimensional array:**

- Declare an array
- Assign values to an array

- **Syntax to declare a multidimensional array:**

```
arraytype arrayname[][]= new arraytype[rowsize][columnsize];
```

Creating Multidimensional Arrays contd.

- Declare an array to store the medal tally of the given countries along with their names

USA	46	28	29
CHINA	38	28	22
INDIA	29	17	19

```
String medalTally[][] = new String[3][4];
```

Assigning Values to a Multidimensional Array

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

	col-index 0	col-index 1	col-index 2	col-index 3
row-index 0	USA	46	28	29
row-index 1	CHINA	38	28	22
row-index 2	GBR	29	17	19

medalTally[0][0]	=	"USA"
medalTally[0][1]	=	"46"
medalTally[0][2]	=	"28"
medalTally[0][3]	=	"29"
medalTally[1][0]	=	"CHINA"

Assigning Values During Declaration

- Assign values to the elements of a multidimensional array at the time of declaration

```
String medalTally[][] = new
String[][]{{"USA","46","28","29"}, {"CHINA","38","28","22"}, {
"GBR","29","17","19"}};
```

Accessing Multidimensional Arrays

How can we access the values assigned to the elements of an array?



Accessing Multidimensional Arrays

- **Syntax to access a multidimensional array:**

```
arrayname[row][column];
```

- **Displaying the elements of an array using the *for* loop:**

```
String medalTally[][] = new
String[][]{{"USA","46","28","29"}, {"CHINA","38","28","22"}, {
"GBR","29","17","19"}};

for(int i=0;i<3;i++){
    for(int j=0;j<4;j++){
        System.out.println(medalTally[i][j]);
    }
}
```

Accessing Arrays of Unknown Size contd.

- **If the size of an array is unknown:**
 - Use the length property of the array to traverse through it
- **Traversing through the array using the *for* loop and the length property:**

```
String medalTally[][] = new
String[][]{{"USA","46","28","29"}, {"CHINA","38","28","22"}, {
"GBR","29","17","19"}};

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

Using for-each with Multidimensional Arrays

- **Displaying the elements of a multidimensional array using the for-each loop:**

```
String medalTally[][] = new
String[][]{{"USA","46","28","29"}, {"CHINA","38","28","22"}, {"GB
R","29","17","19"}};




    for(String[] i: medalTally){
        for(String j:i){
            System.out.println(j); }}
```


Limitations of the for-each Loop

- The for-each loop cannot be used to:
 - Traverse through an array in reverse order
 - Assign elements to different positions in an array
 - Track the index position of the current element in an array
 - Access an element other than the current element on each pass

Interactive Demo

Write a program to accept and display the medal tally details of the given countries.

TOP COUNTRIES							
RANK		COUNTRY	G	S	B	TOTAL	
1		USA	United States	46	37	38	121
2		CHN	China	26	18	26	70
3		GBR	Great Britain	27	23	17	67



Interactive Demo

Write a program to accept the details given in the following medal tally and display the total number of Gold, Silver, and Bronze medals that were distributed.

G	S	B
46	37	38
26	18	26
27	23	17



Jagged Array

A Jagged array is an array of arrays in which the member arrays can be of different sizes. It creates a table with different sizes of columns in a row.

Syntax to declare a Jagged array is:

```
int[][] arr = new int[3][];
arr[0] = new int[5];
arr[1] = new int[3];
arr[2] = new int[4];
```

```
int[][] arr = { {11,10,6}, {5,5,6,22}, {12,16}, {70},
{11,21,40,40,60} };
```

Key Takeaways

- Create a multidimensional array and assign values to it
- Access and manipulate the elements of a multidimensional array
- Use the for-each loop with multidimensional arrays
- Declare a Jagged array





Thank you!