ARRAY TRAVERSAL

To *traverse an array* means to loop through all the array's elements. A *for* statement is typically used, written in the following form:

```
Syntax to Traverse an Array using a for Statement

for ( index variable = 0; index variable < array size; index variable ++ )
{
    statement(s) to repeat
}
statement below
```

The *index variable* is some variable of data type *int* that can be used to subscript the array within the *statement(s) to repeat*.

The loop increments it after each cycle; when it equals the *array size*, the loop’s truth value is false and the loop quits.

If you have only one statement to repeat then the braces {} are optional.

**Example**

A *for* loop that prints each element within the array v.

```
double [] v = { 2, 13, -12, 5, 0, 10, 3 };
System.out.println( "Print Array" );
for ( int k=0; k<v.length; k++ )
    System.out.println( k + ": " + v[k] );
```

The code fragment outputs:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1.0</td>
<td>2.0</td>
<td>3.0</td>
<td>4.0</td>
</tr>
<tr>
<td>1</td>
<td>13.0</td>
<td>-12.0</td>
<td>5.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>10.0</td>
<td>3.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Array Traversal
You can adapt this technique to other applications simply by changing the *statement(s) to repeat* in the loop.

**Example**  
Count the negative values in v.

```java
double [] v = { 2, 13, -12, 5, 0, 10, 3 };  
System.out.println( "How many are negative?" );  
int negCount = 0;  
for ( int i=0; i<v.length; i++ )  
    if ( v[i] < 0 )  
        negCount++;  
System.out.println( negCount + " negative value(s)" );
```

**Example**  
Print the positions of the even values within v.

```java
double [] v = { 2, 13, -12, 5, 0, 10, 3 };  
System.out.println( "Print the positions of the even values" );  
for ( int j=0; j<v.length; j++ )  
    if ( v[j] % 2 == 0 )  
        System.out.println( "v[" + j + "] = " + v[j] + " is even" );
```

The code fragment outputs:

```
Print the positions of the even values  
v[0] = 2.0 is even  
v[2] = -12.0 is even  
v[4] = 0.0 is even  
v[5] = 10.0 is even
```
Example
Print all the array values along with a mark if the value is negative.

double [] v = { 2, 13, -12, 5, 0, 10, 3 };  
System.out.println("Mark the negative values");  
for ( int i = 0; i < v.length; i++ )  
{  
  System.out.print(i + ": " + v[i] );  
  if ( v[i] < 0 )  
    System.out.print("***");  
  System.out.println( );  
}

The code fragment outputs:

Mark the negative values
0: 2.0  
1: 13.0  
2: -12.0***  
3: 5.0  
4: 0.0  
5: 10.0  
6: 3.0

Example
Calculate the sum and average of the values within an array.

double [] v = { 2, 13, -12, 5, 0, 10, 3 };  
System.out.println("Find Sum & Average");  
double sum = 0;  
  for ( int k = 0; k < v.length; k++ )  
    sum += v[k];  
System.out.print ( sum + "/" + v.length );  
System.out.println( " = " + sum/v.length );

The code fragment outputs:

Find Sum & Average
21.0/7 = 3.0
**Example**
Find the smallest item in an array and its position.

double [] v = { 2, 13, -12, 5, 0, 10, 3 };  
System.out.println( "Find smallest item and its position" );  
int smallIsAt = 0;  
for ( int m = 1; m < v.length; m++ )  
    smallIsAt = (v[m] < v[smallIsAt]) ? m : smallIsAt;  
System.out.print( "v[" + smallIsAt + "] = " + v[smallIsAt] );  
System.out.println( " is the smallest" );

The code fragment outputs:

```
Find smallest item and its position  
v[2] = -12.0 is the smallest
```

**Example**
Print the array backwards.

double [] v = { 2, 13, -12, 5, 0, 10, 3 };  
System.out.println( "Print array from last to first" );  
for ( int k = v.length-1; k >= 0; k-- )  
    System.out.println( k + ": " + v[k] );

The code fragment outputs:

```
Print array from last to first  
6: 3.0  
5: 10.0  
4: 0.0  
3: 5.0  
2: -12.0  
1: 13.0  
0: 2.0
```
**Example**
Reach into the array and reverse its values.

double [] v = { 2, 13, -12, 5, 0, 10, 3 };  
System.out.println( "Reverse the elements in the array" );  
int lo, hi;  
double temp;  
for ( lo=0, hi=v.length-1; lo < hi; lo++, hi-- )  
{  
    temp = v[lo];  
    v[lo] = v[hi];  
    v[hi] = temp;  
}
Exercises

Fill in the values of the array $v$.

1. ```
   int [] v = { 1, 2, 3, 4, 5 }; 
   for ( int k=0; k < v.length; k++ ) 
       v[k] = v[k] - 1;
```  

   \[ v \]

   \[ \begin{array}{c}
   0 \\
   1 \\
   2 \\
   3 \\
   4 \\
\end{array} \]

2. ```
   int [] v = { 1, 2, 3, 4, 5 }; 
   for ( int k=1; k < v.length; k++ ) 
       v[k] = v[k-1];
```  

   \[ v \]

   \[ \begin{array}{c}
   0 \\
   1 \\
   2 \\
   3 \\
   4 \\
\end{array} \]

3. ```
   int [] v = { 1, 2, 3, 4, 5 }; 
   for ( int k=0; k < v.length; k++ ) 
       v[k] = 5 - v[k];
```  

   \[ v \]

   \[ \begin{array}{c}
   0 \\
   1 \\
   2 \\
   3 \\
   4 \\
\end{array} \]

What’s the final value of the variable $sum$?

4. ```
   int [] v = { 1, -2, 3, -4, 5, -6, 7, -8, 9 }; 
   int sum = 0; 
   for ( int k=0; k < v.length; k++ ) 
       sum += k;
```  

5. ```
   int [] v = { 1, -2, 3, -4, 5, -6, 7, -8, 9 }; 
   int sum = 0; 
   for ( int k=0; k < v.length; k++ ) 
       sum += v[k];
```  

6. ```
   int [] v = { 1, -2, 3, -4, 5, -6, 7, -8, 9 }; 
   int sum = 0; 
   for ( int k=0; k < v.length; k += 2 ) 
       sum += v[k];
```
What’s the final value of the variable `sum`?

7. ```
int [] v = { 1, -2, 3, -4, 5, -6, 7, -8, 9 };
int sum = 0;
for ( int k=0; k < v.length; k++ )
  if ( v[k] > 0 )
    sum += v[k];
```