

```
System.out.println( Integer.toString( 55, 2 ) );
```

prints out the binary representation of 55.

The `int` value that is represented by a `String` can be obtained by calling the method `Integer.parseInt`. This method generates an exception if the `String` does not represent an `int`. Exceptions are discussed in Section 2.5. More work is required to obtain a `double` from a `String`. Here are some examples:

```
int    x = Integer.parseInt( "75" );
double y = Double.valueOf( "3.14" ).doubleValue( );
```

2.4 Arrays

An *array* stores a collection of identically typed entities.

The *array indexing operator* `[]` provides access to any object in the array.

Arrays are indexed starting at zero. The number of items stored in the array is obtained by the *length* field. No parentheses are used.

An *aggregate* is a collection of entities stored in one unit. An *array* is the basic mechanism for storing a collection of identically typed entities. In Java the array is not a primitive type. Instead, it behaves very much like an object. Thus many of the rules for objects also apply to arrays.

Each entity in the array can be accessed via the *array indexing operator* `[]`. We say that the `[]` operator *indexes* the array, meaning that it specifies which object is to be accessed. Unlike C and C++, bounds checking is performed automatically.

In Java, arrays are always indexed starting at zero. Thus an array `a` of three items stores `a[0]`, `a[1]`, and `a[2]`. The number of items that can be stored in an array `a` can always be obtained by `a.length`. Note that there are no parentheses. A typical array loop would use

```
for( int i = 0; i < a.length; i++ )
```

2.4.1 Declaration, Assignment, and Methods

An array is an object, so when the array declaration

```
int [] array1;
```

To allocate an array, use `new`.

is given, no memory is yet allocated to store the array. `array1` is simply a name (reference) for an array, and at this point is `null`. To have 100 `ints`, for example, we issue a new command:

```
array1 = new int [ 100 ];
```

Now `array1` references an array of 100 `ints`.

There are other ways to declare arrays. For instance, in some contexts