# Arrays

**Creating an Array**

# Declaring an Array

```
int[] nums = new int[3];
```

**Key elements in the syntax**

• [] (read: "array")

• name (`nums`)

• `new` keyword

• size (3)

# Initializing an Array

- first declare, then initialize

```
int[] nums = new int[3];

nums = new int[]{3, -1, 17};
```

- initialization can be done without keyword new

```
nums = {3, -1, 17};
```

- declaring and initializing at once

```
int[] myNumbers = new int[]{3, -1, 17};

int[] myIntArray = {2, 0, -102};
```

```java
// there are more allowed ways to declare an array

int[] nums;

int [] nums;

int []nums;

int nums[];

int nums [];


// NOTE: the size of the array can only be on the right-hand side!!


int[3] nums = ...   // DOES NOT COMPILE
```

```java
// you can have multiple values in one declaration
// (not recommended, but possible)
int[] myNumbers, myIntValues;


// you can even declare int number and int array in the same line
int myNumbers[], a;


// arrays don't implement equals() method
int[] nums1 = new int[] {1, 2, 3};

int[] nums2 = new int[] {1, 2, 3};

System.out.println(nums1.equals(nums2));
  => false
```

```java
// printing out the array

int[] nums = new int[] {3, -1, 17};

System.out.println(nums);

  => [I@7a4f0f29

System.out.println(Arrays.toString(nums));

  => [3, -1, 17]




// every array has a property called "length" (it's not a method!)

int[] nums = new int[] {3, -1, 17};

System.out.println(nums.length);

  => 3
```