

# Arrays



# What is an Array?

# What is an array?

An array is a group of items all of the same type which are accessed through a single identifier.

```
int[] nums = new int[10];
```

**0 1 2 3 4 5 6 7 8 9**

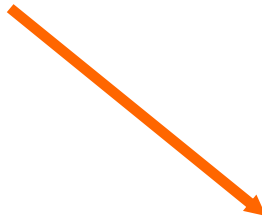
**nums**

<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

# Array References

```
int[] nums;
```

nums  
null



null

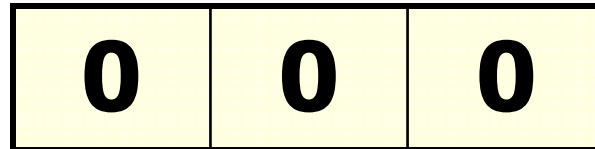
**nothing**

**nums is a reference to an integer array.**

# Array Instantiation

```
new int[3];
```

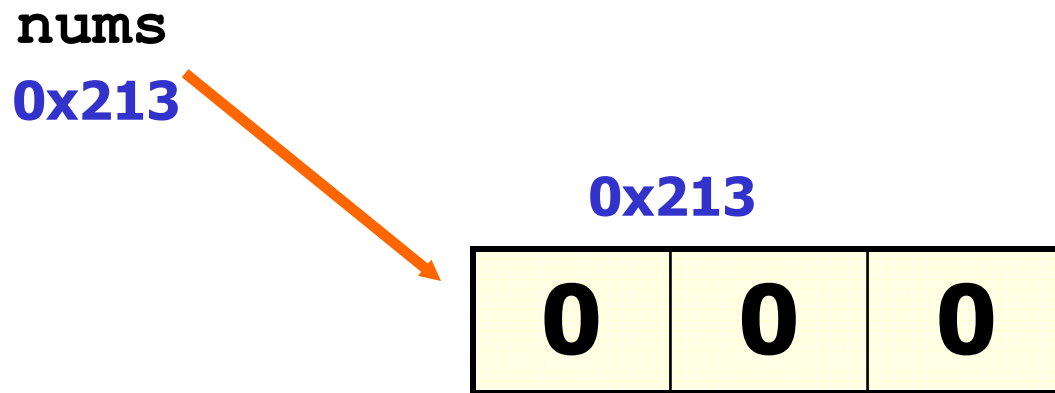
0x213



arrays are Objects.

# Arrays

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



**nums is a reference to an integer array.**

# Arrays

```
int[] nums = new int[10]; //Java int array
```

	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
<b>nums</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

**Arrays are filled with 0 values when instantiated. The exact value of each spot in the array depends on the specified type for the array.**

# Arrays

```
new int[10]; //Java int array
```

0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0

**Once an array object has been instantiated, the size many never change. To increase or decrease the size, a new array would need to be instantiated and all old value copied.**



# Arrays

```
int[] nums = {2,7,8,234,745,1245};
```

	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>nums</b>	<b>2</b>	<b>7</b>	<b>8</b>	<b>234</b>	<b>745</b>	<b>1245</b>

**An array can be initialized with values.**

# Indexes

	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
<b>nums</b>	<b>9</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

The **[spot/index]** indicates which value in the array is being manipulated.

**nums[0] = 9;**

The **0** spot is being set to **9**.

# Indexes

Java indexes must always be *integers* and the first index will always be 0.

	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
<b>nums</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>	<b>0</b>

# Printing Array Values

# Printing Array Values

```
int[] nums = {2,3,5,1,0,6,7};
```

```
out.println(nums[0]);  
out.println(nums[2]);  
out.println(nums[5]);
```

**OUTPUT**

**2**

**5**

**6**

	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>nums</b>	<b>2</b>	<b>3</b>	<b>5</b>	<b>1</b>	<b>0</b>	<b>6</b>	<b>7</b>

# Printing Array Values

```
int[] nums = {2,3,5,1,0,6,7};
```

```
out.println( nums[ 1 + 3 ] );  
out.println( nums[ 7 / 2 ] );  
out.println( nums[ 6 ] );
```

**OUTPUT**

**0  
1  
7**

	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>nums</b>	<b>2</b>	<b>3</b>	<b>5</b>	<b>1</b>	<b>0</b>	<b>6</b>	<b>7</b>

# Setting Array

# Spots

# Setting array spots

```
int[] nums = new int[10];
```

```
nums[0] = 231;
```

```
nums[4] = 756;
```

```
nums[2] = 123;
```

```
out.println(nums[0]);
```

```
out.println(nums[1]);
```

```
out.println(nums[4]);
```

```
out.println(nums[4/2]);
```

**OUTPUT**

**231**

**0**

**756**

**123**



# Setting array spots

```
double[] nums = new double[10];
```

```
nums[0] = 10.5;
```

```
nums[3] = 98.6;
```

```
nums[2] = 77.5;
```

```
out.println(nums[0]);
```

```
out.println(nums[3]);
```

```
out.println(nums[7]);
```

**OUTPUT**

**10.5**

**98.6**

**0.0**

# Setting array spots

```
String[] words = new String[10];
```

```
words[0] = "dog";
```

```
words[3] = "cat";
```

```
words[2] = "pig";
```

```
out.println( words[0] );
```

```
out.println( words[3] );
```

```
out.println( words[7] );
```

**OUTPUT**

**dog**

**cat**

**null**

# Accessing Arrays with Loops

# Accessing Arrays with Loops

```
int[] nums = {3,2,5,1,0,6};  
for(int spot=0; spot<nums.length; spot++)  
{  
    out.println(nums[spot]);  
}
```

length returns the # of elements/items/spots in the array!!!

## OUTPUT

3  
2  
5  
1  
0  
6

# Accessing Arrays with Loops

```
int[] nums = {3,2,5,1,0,6};  
for(int item : nums)  
{  
    out.println(item);  
}
```

	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>nums</b>	<b>3</b>	<b>2</b>	<b>5</b>	<b>1</b>	<b>0</b>	<b>6</b>

## OUTPUT

**3**  
**2**  
**5**  
**1**  
**0**  
**6**

# Accessing Arrays with Loops

```
int[] nums = new int[6];  
for(int spot=0; spot<nums.length; spot++)  
{  
    nums[spot] = spot*4;  
}
```

	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>nums</b>	<b>0</b>	<b>4</b>	<b>8</b>	<b>12</b>	<b>16</b>	<b>20</b>

# Accessing Arrays with Loops

```
String[] wrds = {"cat", "pig", "dog"};  
for(String item : wrds)  
{  
    out.println(item);  
}
```

**OUTPUT**

cat  
pig  
dog

	<b>0</b>	<b>1</b>	<b>2</b>
<b>wrds</b>	<b>cat</b>	<b>pig</b>	<b>dog</b>

# Counting Array

# Values



# Counting Array Values

**In order to count the number of occurrences of a particular value, you must use a loop to access all items in the array.**

**You must also include an if statement to check for the specified value and a variable with which to count each of the variable's occurrences.**

# Counting Array Values

loop through all array items

if current item == search value

increase the count by 1

# Counting Array Values

```
//assume nums is an array with values

int count = 0;
for( int item : nums )
{
    if ( item matches provided value )
        count = count + 1;
}

//return or print count
```

# Deleting Array

# Values

# Deleting Array Values

Once instantiated, the size of an array can never change.

```
int[] nums = {1,7,8,7,4,3,7};
```

# Deleting Array Values

**To delete values, a new array must be instantiated.**

```
int[] newRay = new int[ size ];
```

# Deleting Array Values

**Values must be copied from the old array to the new one.**

```
int[] nums = {1,7,8,7,4,3,7};  
int[] newRay = new int[ size ];
```

**loop through nums  
copy stuff to newRay**

# Deleting Array Values

```
int[] nums = {1,7,8,7,4,3,7};
```

## To delete all 7s

Count the 7s

Create an array set to count of non 7s

Copy all non 7s to new array



# Arrays as Instance Variables

# Instance Variables

```
public class Array
{
    private int[] nums;    //has the value null

    public Array(){
        nums = new int[10]; //sizes the array
    }

    //other methods not shown
}
```

# toString()

```
public class Array
{
    //instance vars and other methods not shown

    public String toString()
    {
        String output= "";
        for(int spot=0; spot<nums.length; spot++)
        {
            output=output+nums[spot]+" ";
        }
        return output;
    }
}
```

# toString()

```
public class Array
{
    //instance vars and other methods not shown

    public String toString()
    {
        String output= "";
        for( int val : nums )
        {
            output = output + val + " ";
        }
        return output;
    }
}
```