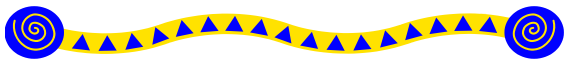
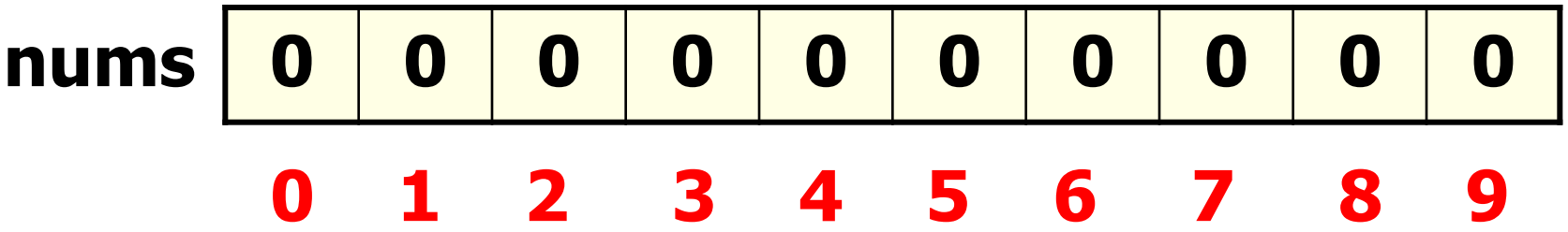
Arrays



What is an array?

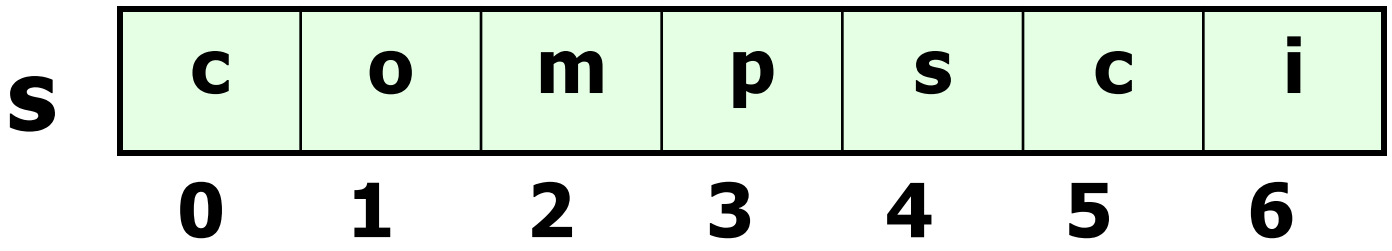
An array is a group of same-type values which are all accessed through a single identifier.

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



Strings are arrays

```
String s = "compsci";      //Strings are arrays
```



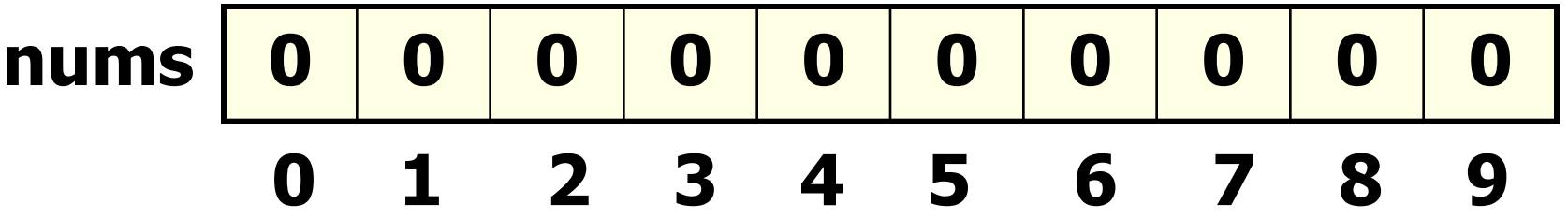
The first index position in a String is 0.

A String is a group of characters…

just like this class ;)

Arrays

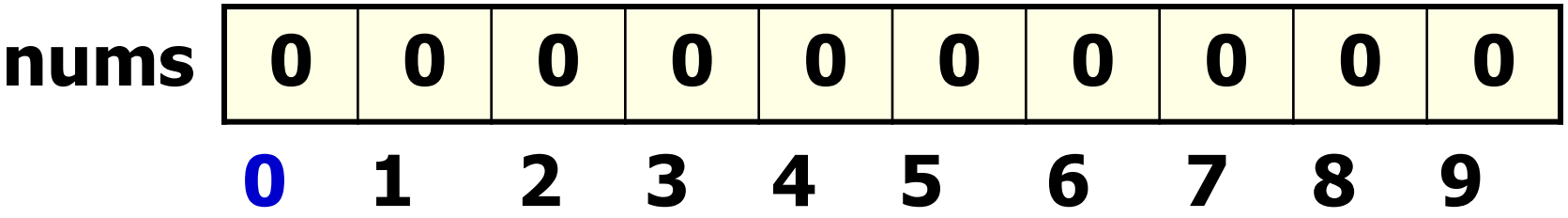
```
int[] nums = new int[10];        //Java int array
```



Usually, an int Array is filled with 0 values when instantiated.
The exact value in each spot in the array depends on the specified type for the array.

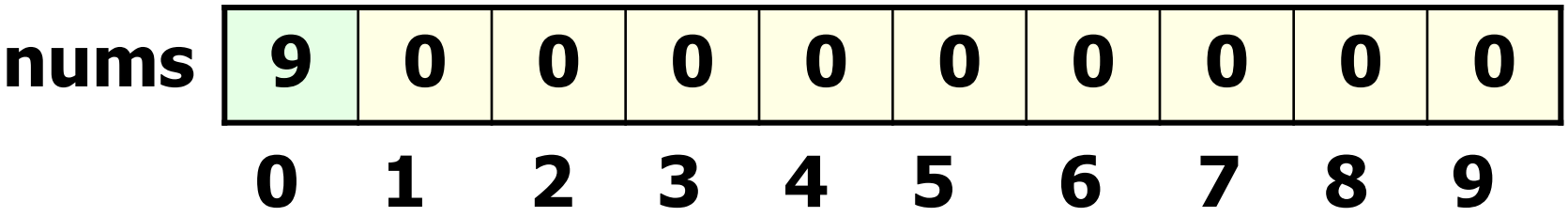
Arrays

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



The size must always be an int.

Indexes

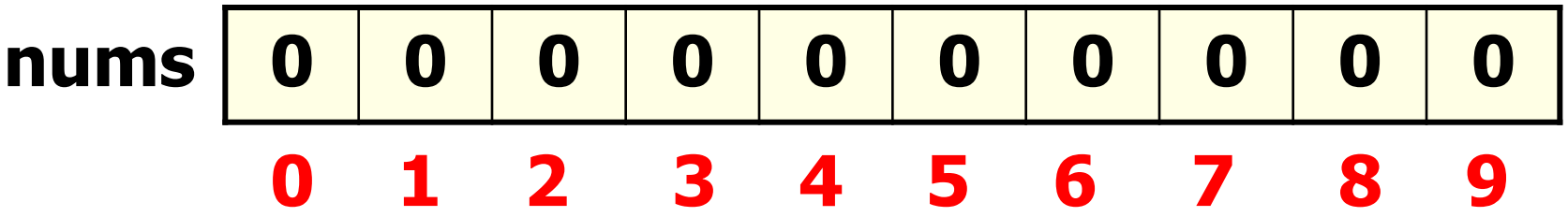


The index indicates which box/spot/element in the array is being manipulated.

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.



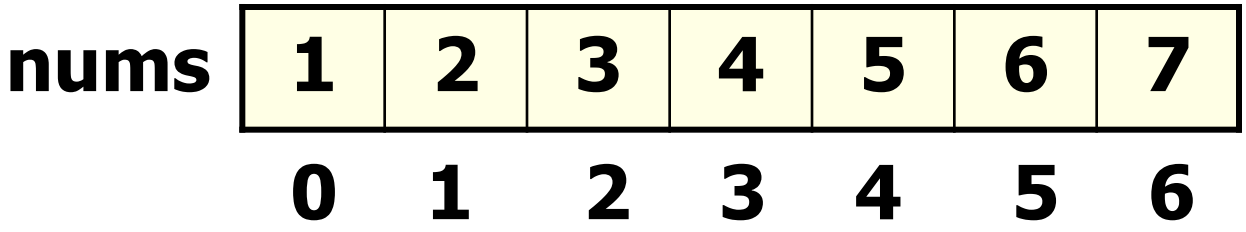
Printing Arrays

Printing array elements

```
int[] nums = {1,2,3,4,5,6,7};

System.out.println(nums[0]);
System.out.println(nums[1]);
System.out.println(nums[2]);
System.out.println(nums[5]);
```

OUTPUT

```
1
2
3
6
```



Printing arrays

```
int[] nums = {1,2,3,4,5,6,7};
for(int i=0; i < nums.length; i++)
{
    System.out.println(nums[i]);
}
```

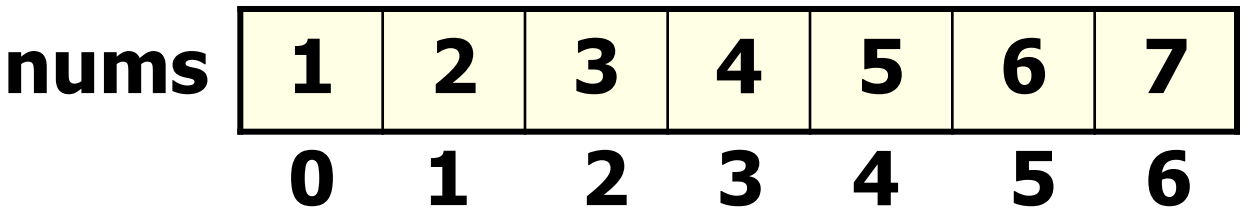
.length returns the # of elements/items in the array

OUTPUT

```
1
2
3
4
5
6
7
```

Printing arrays

```
int[] nums = {1,2,3,4,5,6,7};
for(int  num  :  nums)
{
    System.out.println(num);
}
```



OUTPUT

```
1
2
3
4
5
6
7
```

Setting Arrays

Assigning array elements

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

nums[0] = 231;
nums[4] = 756;
nums[2] = 123;

System.out.println(nums[0]);
System.out.println(nums[1]);
System.out.println(nums[4]);
System.out.println(nums[4/2]);
```

OUTPUT

```
231

0

756

123
```

Assigning array elements

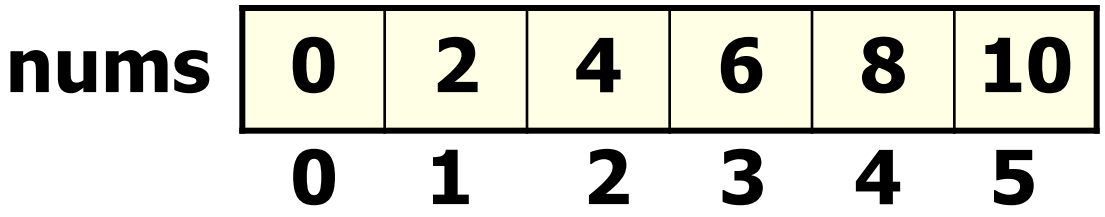
```
double[] nums = new double[10];

nums[0] = 10.5;
nums[3] = 98.6;
nums[2] = 77.5;

System.out.println(nums[0]);
System.out.println(nums[3]);
System.out.println(nums[7]);
```

OUTPUT

```
10.5
98.6
0.0
```

Assigning array elements

```
int[] nums = new int[5];
for(int i=0; i<nums.length; i++)
{
   nums[i] = i*2;
}
```



Entering Array Information

Input / Output

```
//prompt and read in size
int[] ray = new int[size];

for(int i=0; i < ray.length; i++ )
{
   System.out.print( "Enter integer " + i + " : " );
   ray[ i ] = sc.nextInt( );
}

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

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
}
```

class Arrays

Arrays

frequently used methods

Name	Use
sort(x)	puts all items in x in ascending order
binarySearch(x,y)	checks x for the location of y
equals(x,y)	checks if x and y have all the same values
fill(x, y)	fills all spots in x with value y

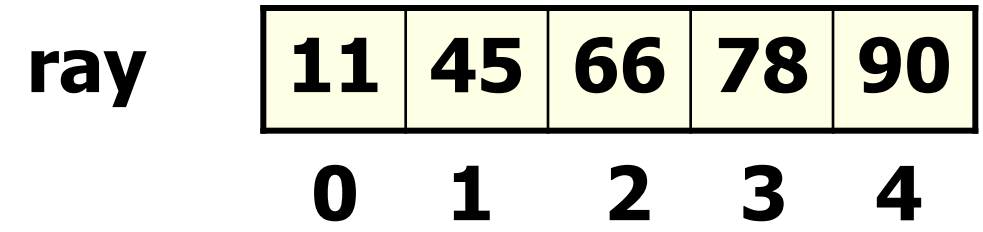
```
import  java.util.Arrays;
```

sort

```
int ray[] = {45,78,90,66,11};

Arrays.sort(ray);

for(int num : ray)
   System.out.println(num);
```

OUTPUT

```
11
45
66
78
90
```



search

```
int ray[] = {45,7,34,66,11};

Arrays.sort(ray);

for(int i=0; i<ray.length; i++)
   System.out.println(ray[i]);

System.out.println(Arrays.binarySearch(ray, 34));
System.out.println(Arrays.binarySearch(ray, 9));
```

OUTPUT

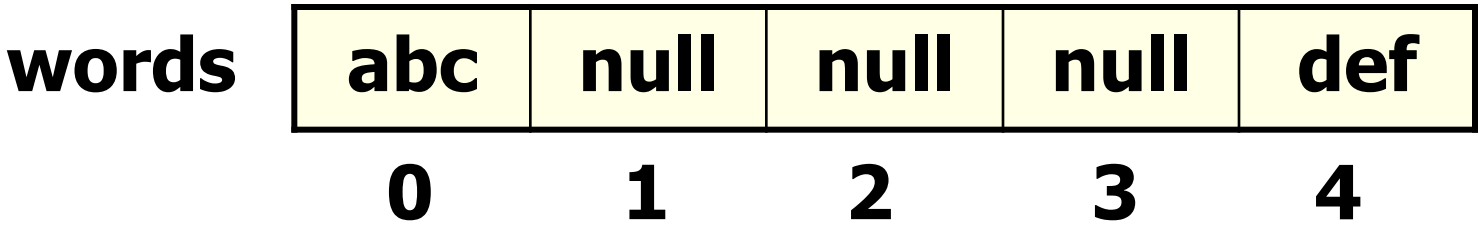
```
7
11
34
45
66
2
-2
```

String arrays

```
String[] words = new String[5];
words[0] = "abc";
words[4] = "def";
System.out.println(words[0]);
System.out.println(words[4]);
System.out.println(words[1]);
```

OUTPUT

```
abc
def
null
```



split

```
String s  = "one two four five";

String[] words = s.split(" ");

System.out.println(words[0]);
System.out.println(words[1]);
System.out.println(words[3]);
```

OUTPUT

```
one
two
five
```