ARRAY'S INTRO

That's kind of a pretty picture...
NEED FOR ARRAYS

- You want to make 1,000,000 variables.
- You want to make a variable number of variables.
- You want to make a bunch of related variables.
NEED FOR ARRAYS

How to deliver mail to the Carter's?
If you want an array of length 5 that holds ints (32 bits each), then you have the picture below:

```
  32  32  32  32  32
```

160 bits

Memory Address: 498000
If you want the data in the 3rd box, you go to the first box, and jump forward $32 \times 2 = 64$ bits

Memory Address: 49800

160 bits
To make a variable with no data:

```java
int x;
```

To make an undetermined number of variables:

```java
int[] y;
```
- To assign our variable the number 5:
  \[ x = 5; \]

- To create 5 variables:
  \[ y = \text{new int}[5]; \]

- To build the street (not the houses)
- Number of lots on the street
- Street name
- Type of houses
To make and assign a variable...

```java
int m = 4;
```

To create 4 variables:

```java
int[] n = new int[4];
```
WHAT'S IN THE ARRAY?

- In the beginning, an array contains "default values"

  ```java
  int[] a = new int[4];  // all 0's
  double[] b = new double[4]; // all 0.0's
  String[] c = new String[4]; // all null
  ```

- Null's are "bad" - we need to put things in the c array!
You can make an array with values in it to start!

```java
int[] a = new int[]{1, 2, 3, 8};

double[] b = new double[]{1.0, 2.0, 3.0, 8.0};

String[] s = new String[]{"a", "b", "x"};
```

Look, no number!
Array's are "0"-indexed

The maximum index is one less than the length
### Assigning

<table>
<thead>
<tr>
<th>Indexes:</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>8</td>
<td>2</td>
<td>6</td>
<td>-2</td>
<td>8</td>
</tr>
<tr>
<td>Length:</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```python
class array[1] = 12;
```
ASSIGNING

Indexes: 0 1 2 3 4

Values: 8 2 6 -2 8

Length: 5

array[1] = 12;
Assigning

Indexes: 0 1 2 3 4

Values: 8 2 6 -2 8

Length: 5

array[1] = 12;
# ACCESSING

<table>
<thead>
<tr>
<th>Indexes:</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>8</td>
<td>12</td>
<td>6</td>
<td>-2</td>
<td>8</td>
</tr>
<tr>
<td>Length:</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```java
int x = array[0];
```
**ACCESSING**

<table>
<thead>
<tr>
<th>Indexes:</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>8</td>
<td>12</td>
<td>6</td>
<td>-2</td>
<td>8</td>
</tr>
<tr>
<td>Length:</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```java
int x = array[0];
```
ACCESSING

Indexes: 0 1 2 3 4

Values: 8 12 6 -2 8

Length: 5

int x = array[0];

x: 8
<table>
<thead>
<tr>
<th>Indexes:</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values:</td>
<td>8</td>
<td>12</td>
<td>6</td>
<td>-2</td>
<td>8</td>
</tr>
<tr>
<td>Length:</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

array[3] = array[4];
### Indexes:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
</table>

### Values:

|   | 8 | 12 | 6 | -2 | 8 |

### Length:

5

```c
array[3] = array[4];
```
Indexes: 0 1 2 3 4

Values: 8 12 6 -2 8

Length: 5

array[3] = array[4];

8 12 6 8 8
String[] array = ...;

int len = array.length; // the length

- The length of an array is a field
int x = 2;

int y = array[2*x - 1];

double[] z = new double[x];

- Indexes may be any integer value
- If an index is out of bounds, your program will crash with an ArrayIndexOutOfBoundsException
INDEX ERROR

```java
int[] array = new int[5];
array[5] = 10;
```

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 5
  at Test.main(Test.java:6)
```

- This error tells you the index that is out of bounds (5)
- It also tells you the Class (Test) and the line (6) where the error occurred
int[] array = ...  
for(int i = 0; i <= array.length; i++)  
{  
    System.out.println(array[i]);  
}  

- What's wrong?
int[] array = ...
for(int i = 0; i <= array.length; i++)
{
    System.out.println(array[i]);
}

- i should only go up to array.length-1
- Remember: array.length is out of bounds
- Using an array field...

```java
public class Example {
    private int[] numbers;
    public Example() {
        numbers = new int[5];
        for(int i = 0; i < numbers.length; i++)
            numbers[i] = i;
    }
}
```

street with no lots ➔

set up 5 lots ➔

put houses on street ➔
Or...

```java
public class Example {
    private int[] numbers = new int[]{0, 1, 2, 3, 4};
}
```
Using an array field that is already created

```java
public class Example {
    private int[] numbers; // assigned elsewhere

    public int sum() {
        int value = 0;
        for (int i = 0; i < numbers.length; i++)
            value += numbers[i];
        return value;
    }
}
```
COMMON SITUATIONS

- Never!

    public class Example
    {
        private int[] numbers = new int[5];
        for(int i = 0; i < numbers.length; i++)
            numbers[i] = i;
    }

- You can only write variable **declarations** at the class level!