# CMPSC 100 Computational Expression

## Reminders and announcements

Due 13 April 2020:

- Practical 09
- Project Proposal

Due 17 April 2020:

- Quiz 2 due

Due 20-24 April 2020:

- Project updates/demos

## Recap: Data Structures and `ArrayList`

- A data structure is an object that allows us to store multiple pieces of data in one place so that we can interact with them later
- One example is an `ArrayList`

## Recap: ArrayList

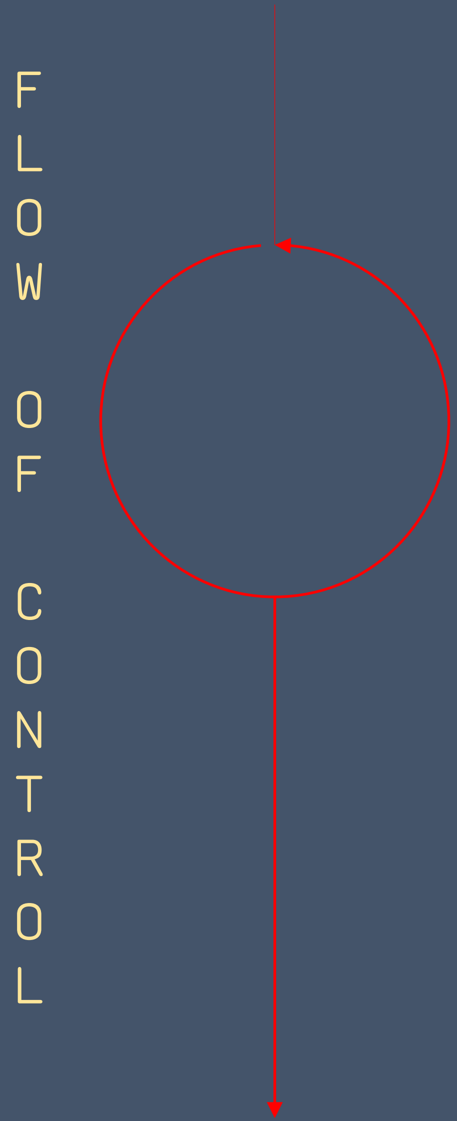
Indexes

| | |
|---|---|
| The Boss | 0 |
| Snooze Magoo | 1 |
| Ulysses | 2 |
| Mr. U | 3 |
| The Mane Man | 4 |

```
ArrayList<String> catNames = new ArrayList<String>();
catNames.add(“The Boss”);
catNames.add(“Snooze Magoo”);
catNames.add(“Ulysses”);
catNames.add(“Mr. U”);
catNames.add(“The Mane Man”);
```

Srsly. Stop sharing my personal infoz! I’ll share ur social security number, next!

## Recap: ArrayList

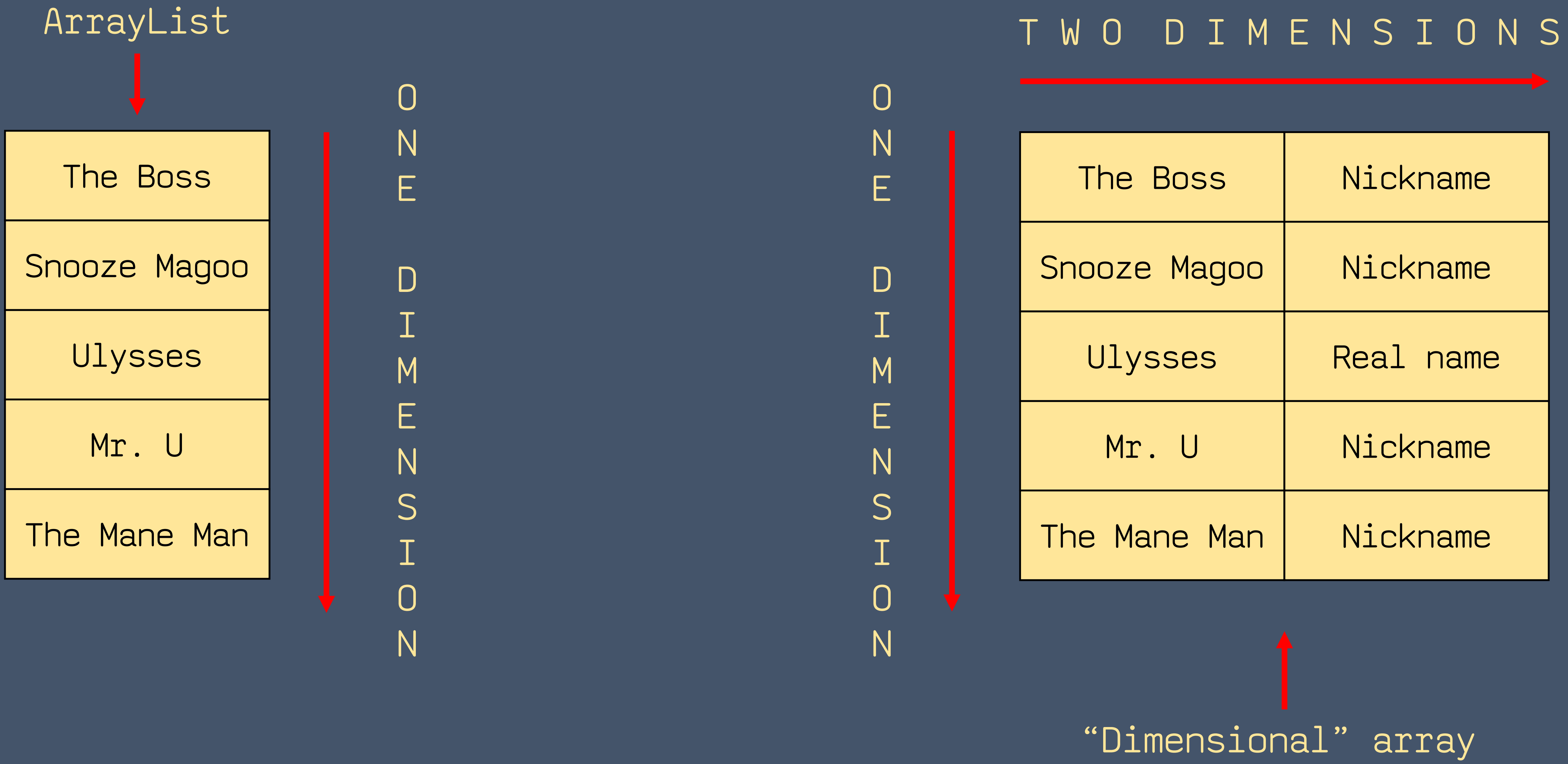


```
int index = 0;
String name;

while (index < catNames.size()){
  name = catNames.get(index);
  System.out.println(name);
  index++; // Don’t forget to increment your index!
}
```

```
> The Boss
Snooze Magoo
Ulysses
Mr. U
The Mane Man
```

# Data Structures (cont'd)



| |
|---|
| The Boss |
| Snooze Magoo |
| Ulysses |
| Mr. U |
| The Mane Man |

| | |
|---|---|
| The Boss | Nickname |
| Snooze Magoo | Nickname |
| Ulysses | Real name |
| Mr. U | Nickname |
| The Mane Man | Nickname |

# Data Structures (cont'd)

|   | 0 | 1 |
|---|---|---|
| 0 | The Boss | Nickname |
| 1 | Snooze Magoo | Nickname |
| 2 | Ulysses | Real name |
| 3 | Mr. U | Nickname |
| 4 | The Mane Man | Nickname |

# Data Structures (cont'd)

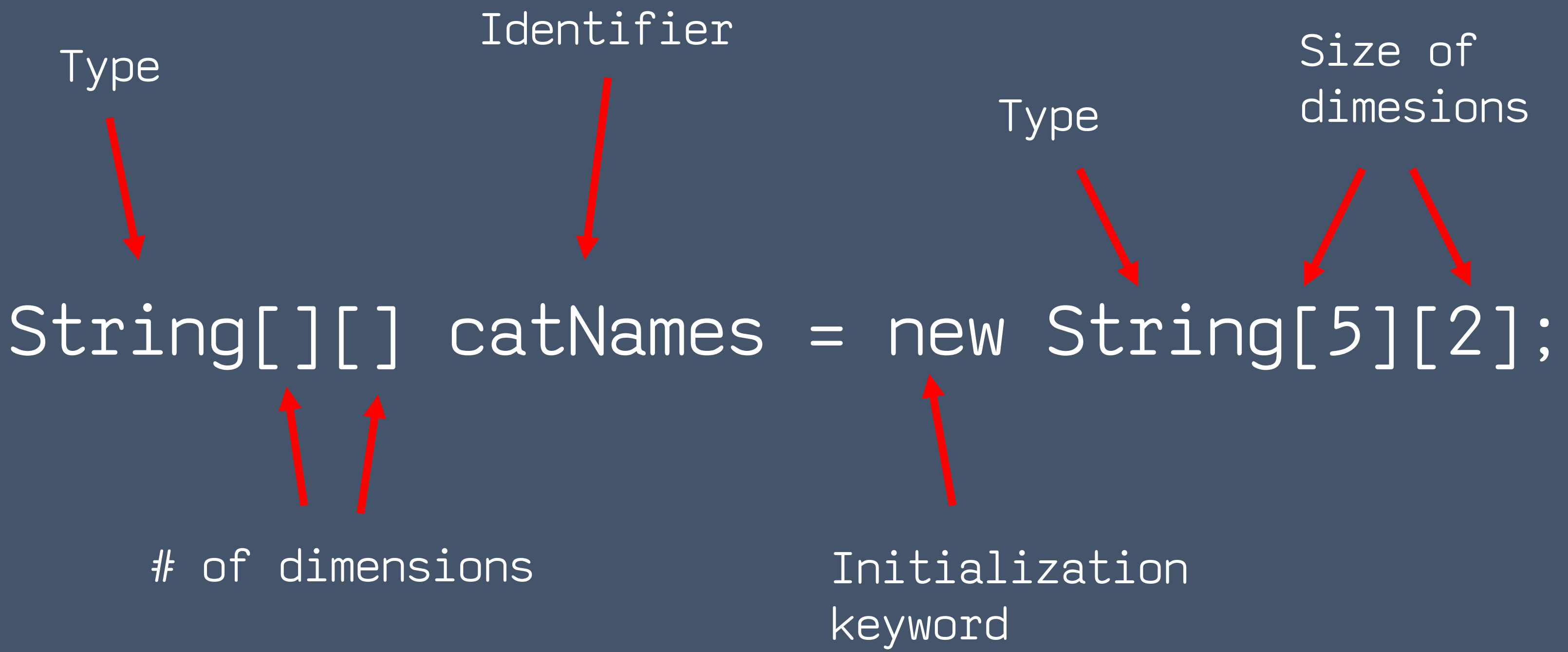
|   | 0 | 1 |
|---|---|---|
| 0 | 0,0 | 0,1 |
| 1 | 1,0 | 1,1 |
| 2 | 2,0 | 2,1 |
| 3 | 3,0 | 3,1 |
| 4 | 4,0 | 4,1 |

## Data Structures (cont’d)

| | 0 | 1 |
|---|---|---|
| 0 | The Boss | Nickname |
| 1 | Snooze Magoo | Nickname |
| 2 | Ulysses | Real name |
| 3 | Mr. U | Nickname |
| 4 | The Mane Man | Nickname |

| | 0 | 1 |
|---|---|---|
| 0 | 0,0 | 0,1 |
| 1 | 1,0 | 1,1 |
| 2 | 2,0 | 2,1 |
| 3 | 3,0 | 3,1 |
| 4 | 4,0 | 4,1 |

## Data Structures: multi-dimensional arrays



```
String[][] catNames = new String[5][2];
```

## Data Structures (cont'd)

| | 0 | 1 |
|---|---|---|
| 0 | The Boss | Nickname |
| 1 | Snooze Magoo | Nickname |
| 2 | Ulysses | Real name |
| 3 | Mr. U | Nickname |
| 4 | The Mane Man | Nickname |

We describe this as a:

- 2D (two dimensional)
- String array
- Having 5 "rows"
- Having 2 "columns"

We express this as:

```
String[][] catNames = new String[5][2];
```

## Data Structures (cont'd)

| | 0 | 1 |
|---|---|---|
| 0 | The Boss | Nickname |
| 1 | Snooze Magoo | Nickname |
| 2 | Ulysses | Real name |
| 3 | Mr. U | Nickname |
| 4 | The Mane Man | Nickname |

We "access" these values with the following notation:

"row"

```
System.out.print(catNames[1][0]);
```

"column"

```
> Snooze Magoo
```

"row"

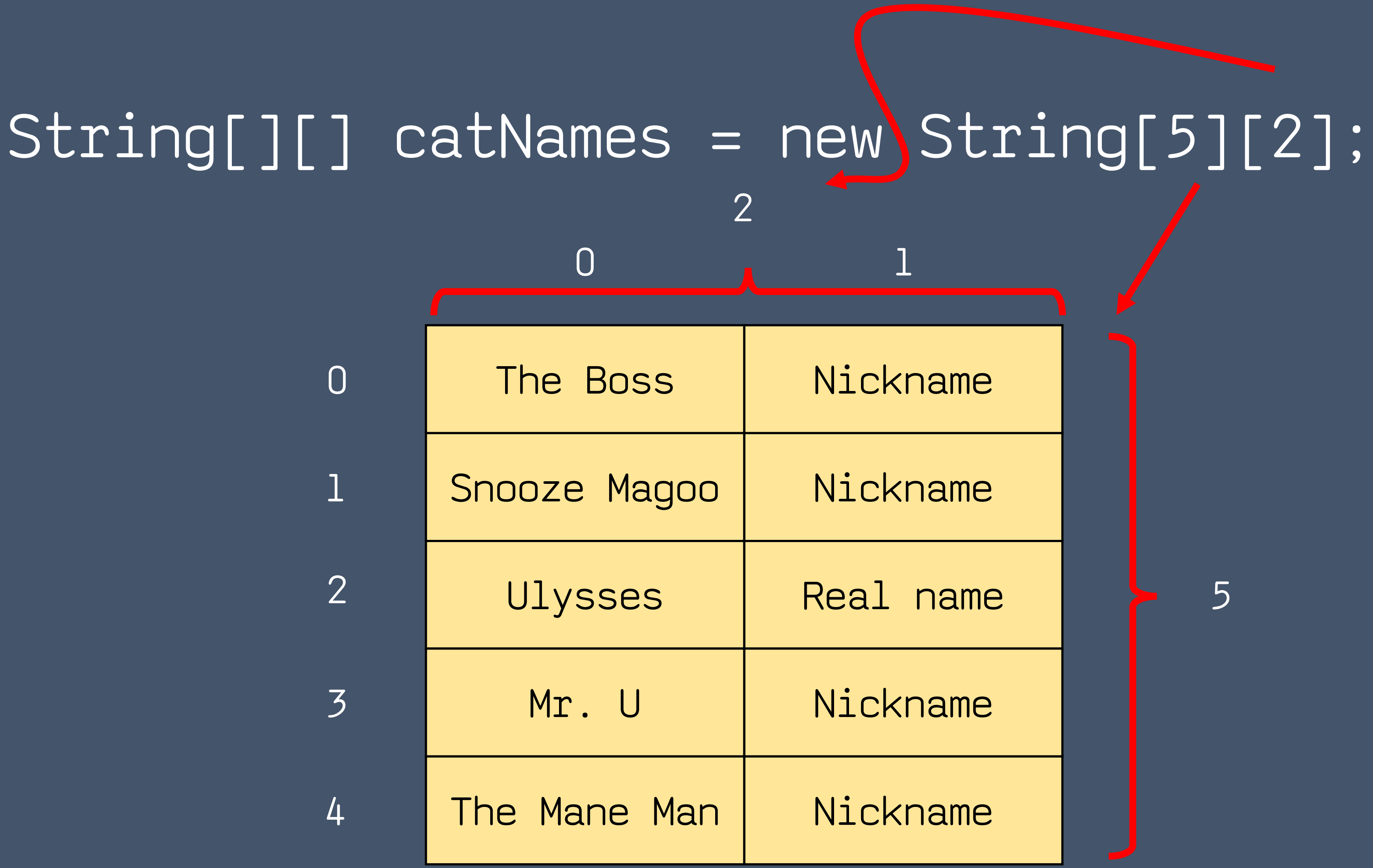
```
System.out.print(catNames[1][1]);
```

"column"

```
> Nickname
```

## Data Structures: multi-dimensional arrays

```
String[][] catNames = new String[5][2];
```



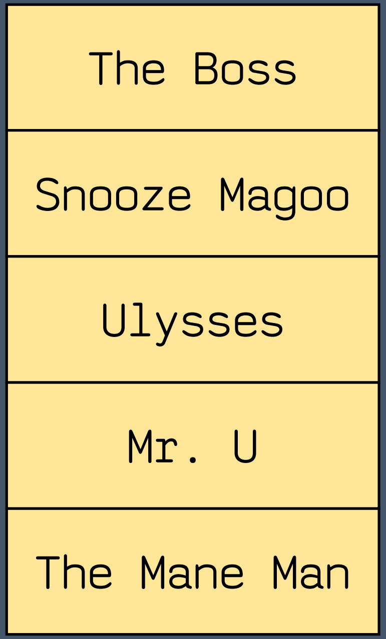
|   | 0 | 1 |
|---|---|---|
| 0 | The Boss | Nickname |
| 1 | Snooze Magoo | Nickname |
| 2 | Ulysses | Real name |
| 3 | Mr. U | Nickname |
| 4 | The Mane Man | Nickname |

## Data Structures: multi-dimensional arrays

What’s the deal with multiple dimensions?

```
ArrayList<String> catNames = new ArrayList<String>();
```

Pretty good for our original list.



## Data Structures

Starts to look pretty rough for something like this.

| | |
|---|---|
| The Boss | Nickname |
| Snooze Magoo | Nickname |
| Ulysses | Real name |
| Mr. U | Nickname |
| The Mane Man | Nickname |

```
ArrayList<ArrayList<String>> catNames = new ArrayList<ArrayList<String>>();
```

## Data Structures

```
String[][] catNames = new String[5][2];
```

|  |  |
|---|---|
| The Boss | Nickname |
| Snooze Magoo | Nickname |
| Ulysses | Real name |
| Mr. U | Nickname |
| The Mane Man | Nickname |

```
String[][][] catNames = new String[5][3];
```

|  |  |  |
|---|---|---|
| The Boss | Nickname | Likes it |
| Snooze Magoo | Nickname | Unknown |
| Ulysses | Real name | Ignores it |
| Mr. U | Nickname | For polite company |
| The Mane Man | Nickname | Power energy |

## Data Structures: `ArrayList` vs. `array`

| ArrayList | Multidimensional Array |
|---|---|
| Is “dynamic” (can expand in size) | Is “static” or “fixed” (once declared, it cannot expand in size) |
| Needs to be imported from java.util | Does not need to be imported; is provided by java.lang (Java API) |
| Cannot store primitive types | Can be made of any type |
| Has various “services” including: .size(), .get(), .add() … | Has *no* services; must be accessed entry by-entry. |
| Can be used with a while or for loop | Is best used with for loops |
| Can easily store only 1 dimension | Can easily store 1 or more dimensions |

## array and loops

- Arrays require a more complex way to use indexes
- Here, while or do…while loops don’t quite do the job
- We need a structure that can create more numbers to track more dimensions, and we don’t want them to hang around
  - That would get confusing

## array and loops

```
// Here, r is short for “row”
```

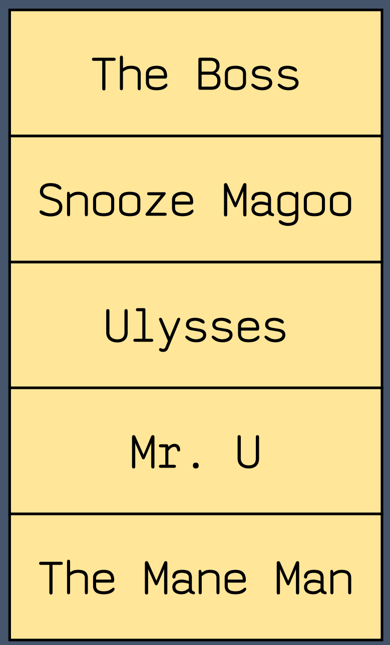
assignment

condition

increment/
decrement

```
for (int r = 0; r < catNames.length; r++) {
  System.out.println(catNames[r]);
}
```

catNames as
dimensional
array



## array and loops

```
// Here, r is short for “row”

// At this point, r doesn’t exist!

for (int r = 0; r < catNames.length; r++) {
  System.out.println(catNames[r]);
}

// r doesn’t exist here, either!
```

## array and loops

catNames as
dimensional
array

```
// Here, r is short for “row”
for (int r = 0; r < catNames.length; r++) {
  // Here, c is short for “column”
  for (int c = 0; c < catNames[r].length; c++) {
    System.out.println(catNames[r][c] + “\t”);
  }
}
```

| | |
|---|---|
| The Boss | Nickname |
| Snooze Magoo | Nickname |
| Ulysses | Real name |
| Mr. U | Nickname |
| The Mane Man | Nickname |

## Activity

cd to your class activities folder

Perform a git pull download master

cd to the activity-13 folder

## Activity

To make a linear equation (aka a straight line between two points) we need to remember the following form:

For any two+ points:

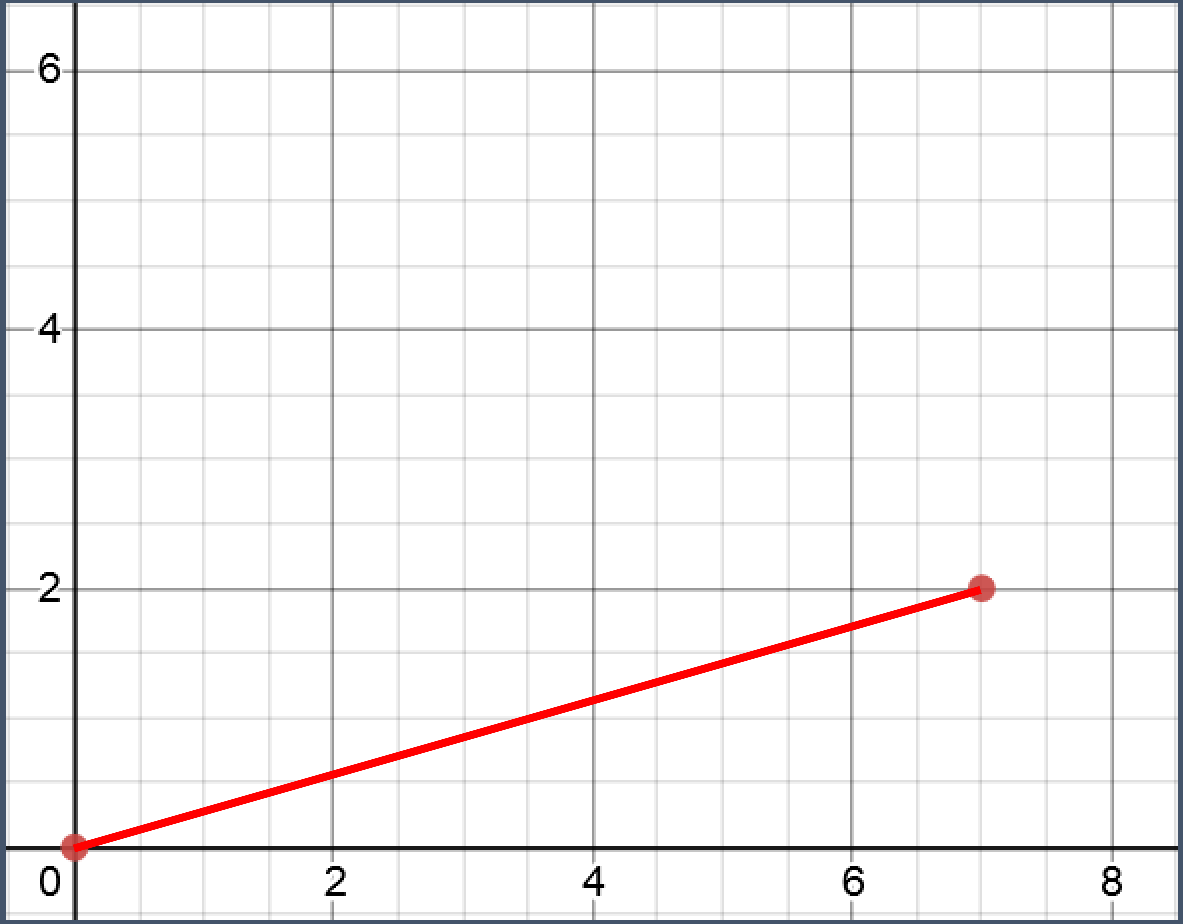
$$y = \underbrace{m}x + b$$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

(cue algebra nightmares now)

## Activity

| | |
|---|---|
| 0 | 0 |
| 7 | 2 |



## Activity

| | |
|---|---|
| 0 | 0 |
| 7 | 2 |

$$\frac{2-0}{7-0}$$

.285…