Class Today

• Group elements together in an array
• Elements perform action one at a time
• Elements perform action all together
• Calculating the smallest in an array

Setup for Lecture today

• Put in four objects, all must be Bipeds:
  – Biped: teenPerson, bunny, cheshireCat, Hare

The Story

• Our group of four will do the following.
  – One at a time they turn all the way around and then kick their right leg up and down.
  – All together they turn all the way around and kick their right leg up and down in synch.
  – One at a time they each say how tall they are.
  – Tallest person says they are the tallest.
  – Change the sizes and again the tallest person should say they are the tallest.
Array for grouping items together

- Each element in the array is in a particular slot number.

<table>
<thead>
<tr>
<th>teenPerson</th>
<th>bunny</th>
<th>cheshireCat</th>
<th>hare</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

- Note size of array is 4, elements are in positions 0 to 3

Create a variable of type array (in myFirstMethod)

- Be sure to check the box that it is an array variable, which means you can store multiple values in it.
- Type is Biped, you can store lots of Bipeds in it.
- Give the variable a name – I gave it `club`
- Click on Custom Array and add in the four Bipeds
- (see next slide)

Biped variable named club

One at a time, turn and kick leg

- One way but lots of code!

- Need six more lines for cheshireCat and hare
- What if we had 10 creatures?
Better way – use “for each in”

- Apply commands to each item in an array

```
while _ : for each in _ if _
```

How to use – for each in

- for each Biped crew in club
  - Crew is a variable to represent each item one at a time in the array club

```
for each Biped crew in club:
  do in order
    crew turn RIGHT
    add detail

  crew getRightHip turn BACKWARD
  duration 0.25
  add detail

  crew getRightHip turn FORWARD
  duration 0.25
  add detail
```

Play it, they turn and kick one at a time

```
for each Biped crew in club:
  drop statement here
```

Turn and kick together

- Use “for each together”

```
if _ do together each in _ together variable_
```

- Create a new variable for this loop, I named mine crew2
- Have each one turn and kick together

```
each Biped crew2 in club do in order
  crew2 turn RIGHT
  add detail

  crew2 getRightHip turn BACKWARD
  duration 0.25
  add detail

  crew2 getRightHip turn FORWARD
  duration 0.25
  add detail
```
Play it and they turn together and kick together

For each person in club, have them say how tall they are
- Note: used another variable crew3!

How do you calculate the tallest person?
- When there are two creatures?
- When there are three creatures?
- When there are four creatures?
- When there are ten creatures?
- When there are 100 creatures?

How do we determine who is the tallest?
What if we add in more creatures?
**Algorithm**
- Assume first person is the tallestSoFar
- Compare tallestSoFar to second person,
- If second person is taller, update tallestSoFar to the second person
- Compare tallestSoFar to third person
- If third person is taller, update tallestSoFar…
Algorithm to find the tallest

• Create variable for tallestSoFar and set to first person in the array
• For each item in the array, if they are taller than the tallestSoFar, reset tallestSoFar to them

Scene Function tallest

• Note need a parameter that is an array of Biped
• Get first item in array as “creatures[0]”

Who’s the tallest?

• Have the tallest say, “I’m the tallest”
• Shrink the hare by 0.5
• Have the tallest say, “I’m the tallest”
• Increase the bunny by 4 times
• Have the tallest say, “I’m the tallest”

• See next slide
What happens if you add in two more bipeds in the array?

- **Add them** to the array

```
Custom Array
```

**myFirstMethod Code (part 1)**

```
myFirstMethod code (part 2)
```

This lecture covered

- Creating an array of objects
- For each object in the array (**one at a time**) execute code with “for each in “
- For each object in the array (**together**) execute code with “each in together”