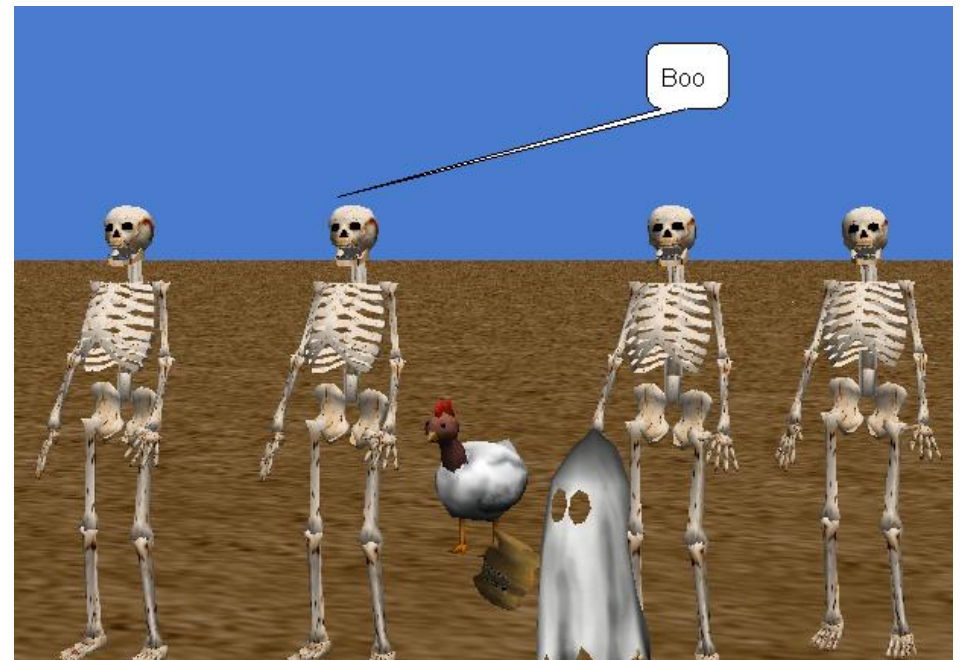


# CompSci 4

## Chap 9 Sec 1

Oct 30, 2007

Prof. Susan Rodger



# Announcements

- Read Chapter 9, Sec 2 for next time
- Assignment 6 due next Tuesday
- Today
  - Chapter 9, Section 1 – Lists
  - Show Halloween card....

# Collections

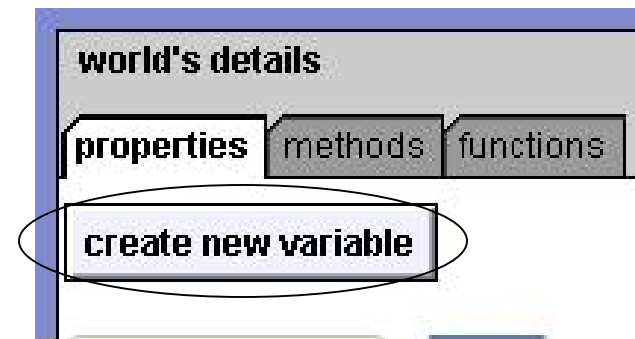
- In some animations, several objects must perform the same actions
  - Example: marching band marching
- It is convenient to collect all objects into a group (collection)
  - Major benefit – write code for all the objects in the group (rather than separate code for each object)

# List

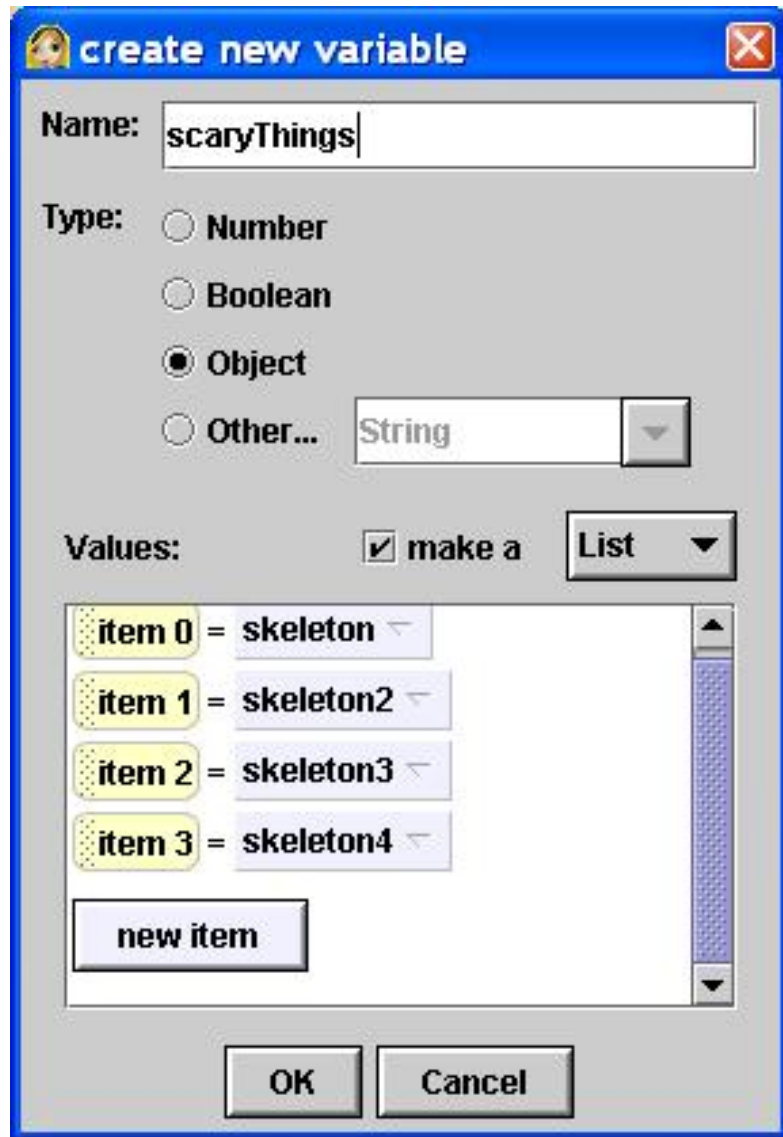
- A list - one way to organize objects into a collection
  - You may use lists to organize
    - Shopping list
    - Todo list
- In programming, a list is a collection of objects or information. We call an organizing structure a **data structure**.

# Creating Lists

- In Alice, a list can be a list of numbers, or a list of objects, or a list of colors, etc.
- Let's create a list of skeletons



# Create List (cont)



**create new variable**

Name:

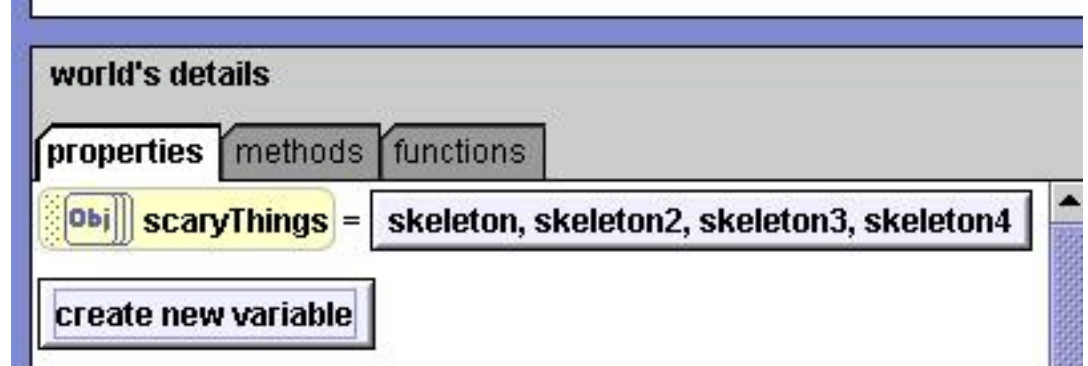
Type:

- ☐ Number
- ☐ Boolean
- ☒ Object
- ☐ Other...

Values: ☒ make a

item 0 = skeleton  
item 1 = skeleton2  
item 2 = skeleton3  
item 3 = skeleton4

- Type in name
- Select type
- Select “make a list”
- Add skeletons to list (click “new item” 4 times)
- Result is:



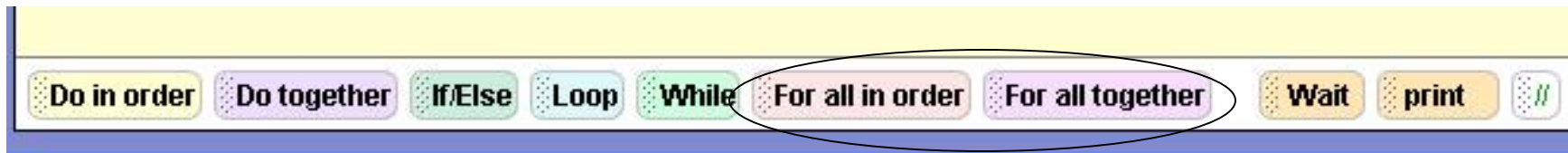
**world's details**

properties methods functions

scaryThings = skeleton, skeleton2, skeleton3, skeleton4

# Programming with a List

- Can “iterate through a list”
  - Do something to each item in the list
    - In order (use “For all in order”)
    - All together (use “For all together”)



# Example/Demo: Iteration in Order

For each skeleton in order

skeleton says “Boo”

For each skeleton in order

skeleton turns its head around

The image shows two Scratch code blocks. The first block is a 'For all' loop with 'world.scaryThings' as the variable, 'one' as the count, and 'item\_from\_scaryThings' as the object. The loop body contains a 'say Boo! more...' block. The second block is also a 'For all' loop with the same variable and count, but the object is 'item\_from\_scaryThings'. The loop body contains a 'turn left 1 revolution' block, with the text 's part named upperBody.neck.head' preceding the turn block.

```
[-] For all world.scaryThings, one [obj] item_from_scaryThings at a time  
  item_from_scaryThings say Boo! more...  
  
[-] For all world.scaryThings, one [obj] item_from_scaryThings at a time  
  item_from_scaryThings 's part named upperBody.neck.head turn left 1 revolution
```



# Applying a Part of an object

For all world.scaryThings , one Obj item\_from\_scaryThings at a time  
skeleton turn left 1 revolution more...

skeleton's details

properties methods functions

other

skeleton's current pose

skeleton's part named key

skeleton's variable named v

- Drag in skeleton turn
- Select part
- Drag over part
- Drag in item
- Type in part

For all world.scaryThings , one Obj item\_from\_scaryThings at a time  
skeleton 's part named turn left 1 revolution m

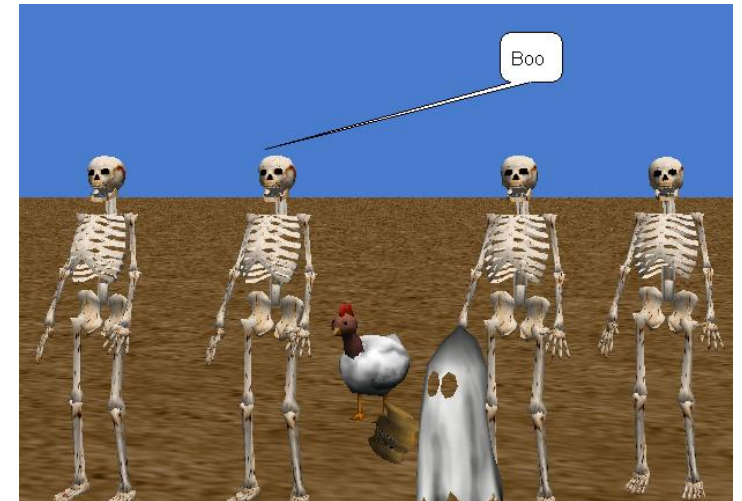
For all world.scaryThings , one Obj item\_from\_scaryThings at a time  
item\_from\_scaryThings 's part named turn left 1 revolution

For all world.scaryThings , one Obj item\_from\_scaryThings at a time  
item\_from\_scaryThings 's part named upperBody.neck.head turn left 1 revolution

# Example/Demo: Iteration Together

For all skeleton together  
skeleton says “Boo”

For all skeleton in together  
skeleton turns its head and neck around



```
[-] For all world.scaryThings ▾, every [Obj] item_from_scaryThings together
  item_from_scaryThings ▾ say I said BOO! ▾ more... ▾

[-] For all world.scaryThings ▾, every [Obj] item_from_scaryThings_#2 together
  item_from_scaryThings_#2 ▾ 's part named upperBody.neck.head ▾ ▾ turn left ▾ 1 revolution ▾
```

# List Questions

- What are differences between *For all in order* and *For all together*?
- When would you want to use each of them?
- What can you put in a list?
- When can you refer to a part of an object in a list?

# Classwork today

- Create a list of players
- Make them do several things.

