

CMPS 10 Lecture Notes: Lecture 15 (2-23-2016)

Week 8: Lists

Part 1: Last week remainder slides

Part 2: Static list Coding Example

Part 3: Dynamic List Coding Example

Part 4: Map Function

Part 5: Keep Desired Elements

Part 6: Week 8 Homework Assignment.

Part 1: Last Week Bouncing Directions

- 90 is right
- -90 is left
- 0 is up
- 180 is down
 - You can use the "point in direction" block and "pick random" block to choose a random range to bounce back from.
 - If you want your ball to bounce "upwards" you would probably pick something between negative 85 and positive 85. You won't want negative to positive 90 because then your ball might "bounce" horizontally!

Part 2: List

- A LIST is a way to combine several values of the same thing.
- A list can contain anything you want! Nouns! Adverbs! Any thing you want!
- You use a list block that looks similar to the 'script variable' block. And then you can drag that list block into a SET block!
 - So now your variable isn't just a single thing! Your variable is a whole list of things!
- You can also dynamically add elements into a list.
- There is a block: "Add THING to list" (under variables tab in Snap)
 - BUT before you use it, you must TEACH Snap that you are dealing with a list.
 - You do that with a set block (so you can say set localList to an "empty list" (you say "empty list" by dragging the "list" block into the value part of the set block).
- You can also have the sprite say the content of the list by dragging the list variable into the say block, as you would any other variable.
- When you have a script variable (as opposed to a global variable), it can sometimes be harder to see it displayed.
- So one thing that you can do is set one list equal to another list, so you can make a new global list and set it to be equal to your localList.
- There is another block "replace item X of list with thing"
 - You'll have to drag your list variable into the little orange icon on that "replace item X..." block so that it knows what specific list you are dealing with.

In Snap:

- List BEGINS with 1. So the list begins with item 1.
- And so if you use the 'replace item X of LIST with THING' block, you can easily swap out a certain element in your list with a new thing!

There is also another very useful block: "length of LIST"

- That returns a number that is the length of the list.
- This is very useful because you don't always know what the length of your list is going to be ahead of time.
 - So you are replacing STATIC knowledge with DYNAMIC knowledge, because the length function figures it out for you. You don't have to figure it out yourself.

- * In general: Dynamic knowledge is great. It makes your code much more flexible!
- You sometimes have to be careful when dealing with an empty list, as it won't have any length.

Let's write a function that outputs whether a particular name is in the globalList.

- There is already a block that exists "LIST contains THING"
- You can check to see if your list contains an element that you already know!
 - Combine that with if else statements to do certain things based on the contents of your list!

Example of capitalizing the first letter of every element in a list.

- There is a "For each THING in LIST" block that will loop through every element in the list.
- In this example, "sum" isn't exactly a sum as in the result of addition. It is going to be each letter of the word.
- Now we want to check to see if it is capitalized or not.
 - And the way we are doing that for now is we are just checking each individual letter. So checking if this is a small b, then we replace it with big B. And then we need to do that for every letter.
 - We'll make a new concatenate variable, and initialize it to just have the value of our single capital letter.
 - And then once we do that, we then want to loop through the rest of the letters, and we do that using a for loop, starting at TWO (because we've already dealt with the first letter) and going until the end (which we can do using the "length of" block).
 - * Inside of here, we want to set concatenate to whatever concatenate was before, JOINED with the letter located at position i.

OK, there is something that is going to help us generalize the above: Unicode. AKA Ascii code. You can find this information if you search online.

- Each letter of the alphabet is assigned to a letter.
- The difference between a lower case letter and an upper case letter is always going to be 32.
- So we can do two checks. We can double check that we are dealing with a lower case letter by seeing that we are dealing with something with a unicode value of 97 or greater, AND if we are, then subtract 32 from it!
- There is the block "Unicode as Letter" that will help us out with this!
- So we can revise our code to check if unicode of firstLetter is greater than 96 by using an if statement
 - And then if it is, then we set firstLetter to "unicode (Unicode of first letter - 32) as letter." – Imagine that the parenthesis in this bullet point represent an embedded "unicode of" block INSIDE of a "unicode as letter" block.
 - And then VOILA! We have made it so that we can dynamically change ANY lowercase letter into an upper case one, in a very generalized way.
- And to wrap this up, we want to actually update the global list.
- And to do that, we'll actually to set up another new variable in our scriptVariable tab, we'll call it listIterate.
 - And we'll replace item listIterate of globalList with concatenate, and then use a Change By block to change listIterate by 1.
 - * TA DA! We did it!

Part 3: Map Function

Think of map as an element wise report function

- Map automatically iterates through a list for you!
- It does an operation on every element of a list and report it.
- So you have some operation thing, and if there is a blank spot in your operation, it will fill it in with a value in the map. NOTE: In these notes, a 'blank' will be depicted with an underscore, like: _
- With map, you drag out some operator (so, maybe with a join operator)
 - So, "map join hello _ over globalList" will make hello Ana, hello Paris, hello Sarah, hello John, etc. It joins "hello" with each item in the global list!
 - And "map letter 2 of blank over globalList" will get you the second letter of each word in the global list.

- And "map letter length of _ of _ over globalList" would get you the LAST letter of each word in the global list (because you are getting the letter located at the final position (i.e., the length)).
 - * Map made our lives so much easier!

Another example: Count number of elements that begin with the letter a, OR KEEPING items that meet some special criteria

- "Keep Items such that A = Letter 1 of _ from globalList"
 - This will get a SUBLIST from the original list (the original list being globalList in this case)
 - You'll only get the elements from the original list that meet your criteria (in this case, elements that begin with an A)

Part 4: Grading Schema

- You are going to generate your own sentences.
- When you go to the website to get your assignment, you are going to get some starting code!
- TRY TO GO TO LAB SECTION this week, as this assignment is a little tricky!
 - If you do that you'll be able to get some extra help!