

# Lists

# What is a List?

A **list** in programming is a linearly ordered set of values, in which each value is referenced by an **index value**.

First, demo in Scratch ...

... storing user entered values in a list

... finding the largest element in a list

# Motivation

Suppose a program was needed to read in a list of grades for a class of up to 25 students. If we had one variable for each of the students for holding their grade, we would end up with something like,

```
grade1 = input('Enter grade of first student: ')  
grade2 = input('Enter grade of second student: ')  
grade3 = input('Enter grade of third student: ')  
etc.
```

## Motivation (cont.)

If we wanted to calculate the average of the grades, we would have,

$$\text{avg} = (\text{grade1} + \text{grade2} + \text{grade3} + \text{grade4} + \text{grade5} + \text{grade6} + \dots + \text{grade23} + \text{grade24} + \text{grade25}) / 25$$

This is rather inelegant. Also, it can become completely infeasible if there were to be 1,000 values entered, for example.

# Usefulness of Lists

With the use of lists, there can be ONE name given to the list as a whole, with each individual element of the list accessed by its position in the list,

grades	
<b>0</b>	<b>86</b>
<b>1</b>	<b>92</b>
<b>2</b>	<b>75</b>
<b>3</b>	<b>83</b>
<b>4</b>	<b>97</b>
<b>5</b>	<b>71</b>

grades[0] → 86  
grades[1] → 92  
grades[2] → 75  
etc.  
97  
71

# Creating Lists

In Python programming, a an empty list can be created as follows,

```
grades = []          -- []
```

Items can be added to a list,

```
grades.append(86)    -- [86]
```

```
grades.append(92)    -- [86, 92]
```

```
grades.append(75)    -- [86, 92, 75]
```

Can determine the length of a list,

```
len(grades) → 3
```

Can sort a list,

```
sort(grades)         -- [75, 86, 92]
```

## Using Lists

Now if we wanted to calculate the average of the grades, we would have,

$$\begin{aligned} \text{avg} = & (\text{grades}[0] + \text{grades}[1] + \text{grades}[2] + \\ & \text{grades}[3] + \text{grades}[4] + \text{grades}[5] + \\ & \cdot \\ & \cdot \\ & \text{grades}[22] + \text{grades}[23] + \text{grades}[24]) / 25 \end{aligned}$$

Which an improvement over the use of 25 individual variables.

**Really?!**

## Lists and Iteration

The REAL usefulness of lists is that the index value can be given as a variable,

`grades[i]`

Where the item referenced depends on the CURRENT value of the index variable, in this case variable `i`.

## Lists and Iteration (cont.)

Thus, to add up (and average) all the items in the list, we only need do the following,

```
i = 0
sum = 0
while i < len(grades):
    sum = sum + grades[i]
    i = i + 1

avg = sum / len(grades)
```

# “Blackboard Challenge”

Looking for a volunteer to develop the logic for finding the largest element in a list, without using the sort function.

One extra point on next exam ...