

## For loops

Lecture #8 - COMP 364 February 3, 2010, updated 2012 Derek Ruths Fireworks light up the London skyline and Big Ben just after midnight on Jan. 1, 2012 in London, England. Thousands of people lined the banks of the River Thames in central London to see in the New Year with a spectacular fireworks display. (Dan Kitwood/Getty Images)

### Containers

- Lists are **ordered** containers ("sequences") that can be modified and contain many datatypes.
- Many other container types in Python: tuple (a frozen list), set and dictionary.
- Containers are perfectly suited for what we need to do!

#### Having fun with lists!

```
names = ["Dave", "Mark", "Ann", "Phil"]
a = names[2]  # Access different elements.
names[0] = "Jeff" # Change some elements in the list.
```

names.insert(2, "Thomas") # Insert something in a list.

```
names.append("Peter")  # Adding to the end of a list.
```

```
alist = [[1,2,3], [4,5,"Mark"], [7,8,9]]
```

# How do we access number 8 in "alist"?

## Task 1: Working with all the elements of a list

- Often we will want to do something to each element in a list
  - In the case of "gene <start>..<end>", extract <start> and <end> and subtract them to get the length of a gene
  - In the case of a bunch of data points, add them together and divide by the number of data points
- Version 1: Print out all the elements
- Version 2: Print out all the elements, one element per line
- Version 3: Compute the sum of a list of numbers
- Version 4: Compute the average of a list of numbers

## The for loop: handling one element at a time

- In a for loop we define a set of actions that are done exactly once on each element in a list.
- All the steps must be identically indented under the "for..." statement

the variable that contains the value of the list element



# Exercise: computing the sum of a list of numbers

## Exercise: counting the length of a list

# Exercise: computing the average of a list of numbers

### Other containers: strings and files!

• Strings and files also implement the container interface to a certain extent!

s = "Hello World"
 What is s[2]? Intuitively, what do s[0:5] and s[3:] represent?

```
f = open('species.v9.0.txt')
for line in f:
    print line[:5]
```

for c in "Mus Musculus": # What are we iterating over?
 print c.lower()

### Other ways to obtain lists

- Lists are everywhere: obtaining the argument list for a program, obtaining the list of lines in a file, creating a new list through iteration...
- Some functions that will give us lists:
  - split(...) will deconstruct a line according to a specified delimiter
  - f.readlines()