

# Lesson 61: Magpie Lab #1 (W20D4)

Balboa High School

Michael Ferraro

January 14, 2016

# Do Now

- 1 Open the *student manual* [here](#). You'll be referring to it for the duration of this lab.
- 2 Read the introduction section on p2.
- 3 Read the *Current Work in NLP* section on p11.

Students will learn about chatbots and reinforce their knowledge of Strings by way of the College Board's Magpie lab.

# Natural Language Processing

- A “chatbot” (virtual chat partner) can be programmed (taught) to answer specific kinds of questions; some can learn from conversations and apply learned responses later (artificial intelligence)

---

<sup>1</sup>computer having a conversation with a human, and the human believing it's conversing with another person

# Natural Language Processing

- A “chatbot” (virtual chat partner) can be programmed (taught) to answer specific kinds of questions; some can learn from conversations and apply learned responses later (artificial intelligence)
- Chatbots can't yet interact with humans in a convincing way over the range of topics that might come up; can't yet pass the Turing Test<sup>1</sup>

---

<sup>1</sup>computer having a conversation with a human, and the human believing it's conversing with another person

# Natural Language Processing

- A “chatbot” (virtual chat partner) can be programmed (taught) to answer specific kinds of questions; some can learn from conversations and apply learned responses later (artificial intelligence)
- Chatbots can’t yet interact with humans in a convincing way over the range of topics that might come up; can’t yet pass the Turing Test<sup>1</sup>
- Real world applications
  - Some companies use speech recognition + NLP to route callers more efficiently
  - Apple’s [Siri](#)

---

<sup>1</sup>computer having a conversation with a human, and the human believing it's conversing with another person

# Some Chatbots

- Let's have a brief conversation with Rose:  
<http://brilligunderstanding.com/rosedemo.html>
- Chatbot vs. Chatbot

## Activity 2

- Find a partner with whom you'll work on this activity. **Share a computer.**
- Prepare Eclipse
  - Create project Magpie
  - Import the classes found [here](#), Magpie2 and a driver/runner
- Work on Activity 2 on pp3-4



## Activity 2

- What was the major limitation of using `indexOf()` to detect the presence of a word like “no”?

## Activity 2

- What was the major limitation of using `indexOf()` to detect the presence of a word like “no”?
- Activity 3 addresses misdetected keywords

# Activity 3

- Read the 2 introductory paragraphs for Activity 3 (top of p5)
- Import the sources from [here](#) (3 classes)
- With your partner:
  - read over the `StringExplorer` class
  - predict the printed values
  - run the class to validate your predictions

## Activity 3

- The driver class for Magpie3 is essentially the same as the prior driver
- A new, overloaded method has been added to the Magpie3 class:  
`findKeyword()`
  - read over the method with your partner, try to make sense of it
  - we'll review as a class next time

- Add your Magpie2 customizations to Magpie3
- Per instructions on p7, make an iteration table to show the execution of this method call:

```
findKeyword("yesterday is today's day before.", "day", 0);
```

- Try out Magpie3 to make sure, e.g., “know” isn't misdetected as “no”