



Introduction to conversational software

Alan Nichol
Co-founder and CTO, Rasa



A short history

Conversational software is not a new idea!

- Dates back to at least 1960s
- First wave: command line applications
- ELIZA: 1966





Course content

- Implementing smalltalk, ELIZA style
- How to use regex and ML to extract meaning from free-form text
- Build chatbots that can
 - Query a database
 - Plan a trip
 - Help you order coffee
- Handling statefulness



EchoBot I

```
In [1]: USER: Hello!
Out[1]: BOT: I can hear you, you said: 'Hello!'
In [2]: USER: How are you?
Out[2]: BOT: I can hear you, you said: 'How are you?'
```



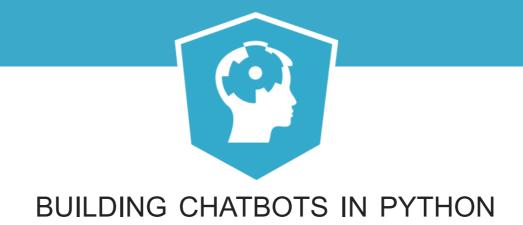
EchoBot II



EchoBot III

```
In [1]: import time
In [2]: time.sleep(0.5)
```





Let's practice!





Creating a personality

Alan Nichol
Co-founder and CTO, Rasa



Why personality?

- Difference between a command line app and a chatbot
- Makes chatbots and voice assistants more accessible and fun to use
- Your users will expect it!



Smalltalk



Including variables

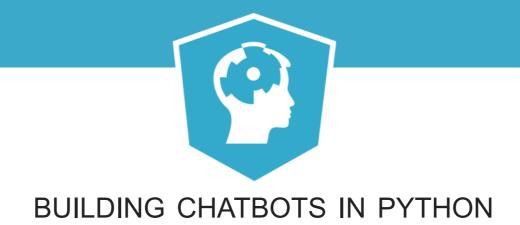


Choosing responses



Asking questions





Let's practice!





Text processing with regular expressions

Alan Nichol Co-founder and CTO, Rasa



Regular expressions

- Match messages against known patterns
- Extract key phrases
- Transform sentences grammatically



The regex behind ELIZA

USER: "do you remember when you ate strawberries in the garden?"

ELIZA: "How could I forget when I ate strawberries in the garden?"



Pattern matching



Extracting key phrases

```
In [1]: import re
In [2]: pattern = "if (.*)"
In [3]: message = "what would happen if bots took over the world"
In [4]: match = re.search(pattern, message)
In [5]: match.group(0)
Out[5]: 'what would happen if bots took over the world'
In [6]: match.group(1)
Out[6]: 'bots took over the world'
```



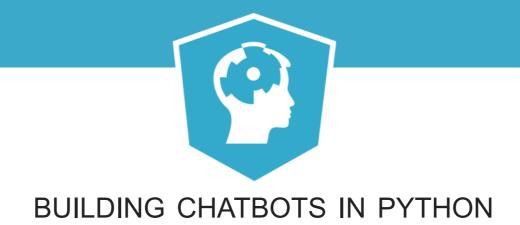
Grammatical transformation



Putting it all together

```
In [1]: pattern = 'do you remember (.*)'
In [2]: message = 'do you remember when you ate strawberries in the garden'
In [3]: phrase = pattern.search(pattern, message).group(1)
In [4]: phrase
Out[4]: 'when you ate strawberries in the garden'
In [5]: response = choose response(pattern)
In [6]: response
Out[6]: 'how could I forget {}'
In [7]: phrase = swap pronouns(phrase)
In [8]: phrase
Out[8]: 'when I ate strawberries in the garden'
In [9]: response.format(phrase)
Out[9]: 'how could I forget when I ate strawberries in the garden'
```





Let's practice!