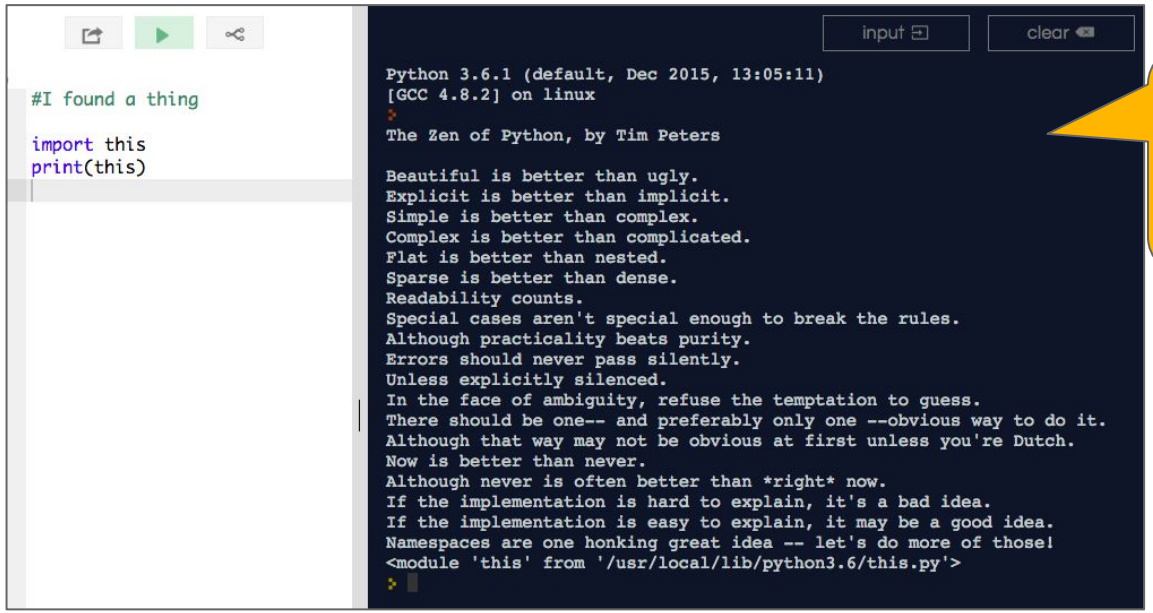
Chatbots with Personality

Notable Code!



I only post here what has been shared in the Slack channel. Please do share!

Question 1

What are 2 kinds of **errors** we should defend against?

Question 2

What does the **strip** function do?

Question 3

What could be wrong with this code?

```
if reply.lower().strip(" ") == "GOOD"
    print("Good!")
```

Question 4

What does this code output?

```
movies = ["Superman", "Frozen", "X-Men"]
print("x-men" in movies)
```

More on Chaining

A shortcut for code

```
1  # Chaining Example
2  # Author: Angelica Lim
3  # Date: Jan. 17, 2018
4
5  # Get the user reply and make it lowercase
6  # without extra characters
7  reply = input().lower().strip("!,.?")
8
9  # Print reply
10 print(reply)
```

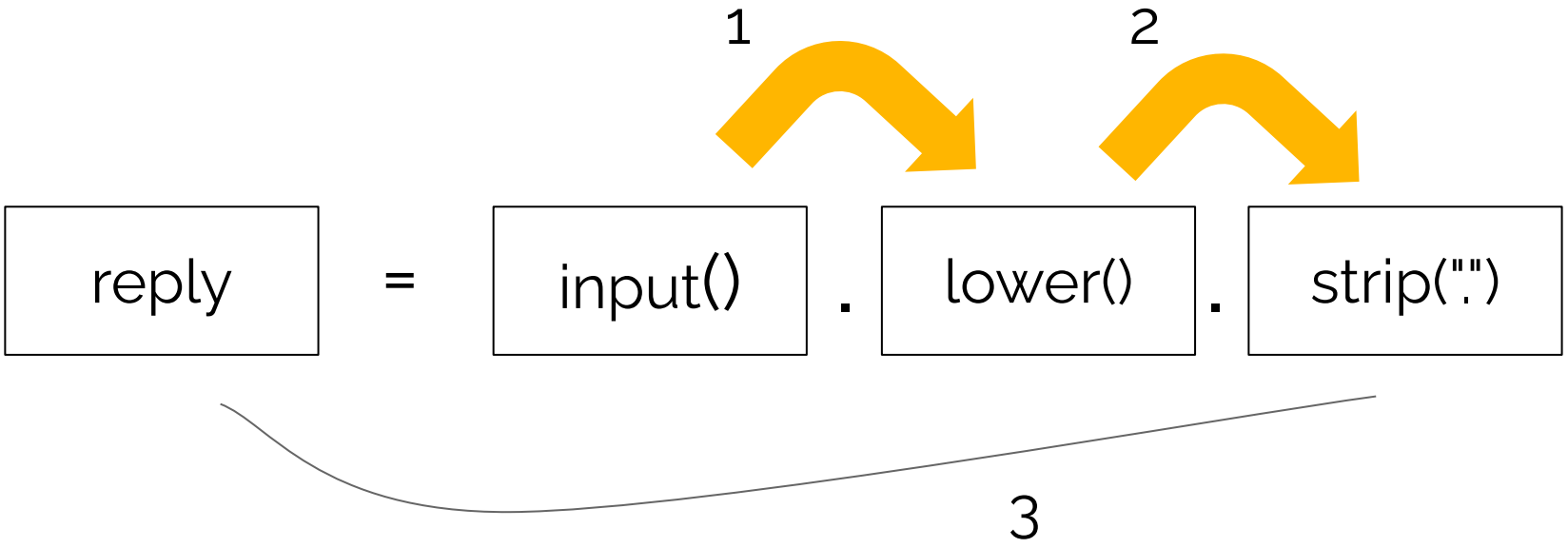
==

```
1  # Chaining Example
2  # Author: Angelica Lim
3  # Date: Jan. 17, 2018
4
5  # Get the user reply
6  reply = input()
7
8  # Make the reply lowercase
9  lowercase = reply.lower()
10
11 # Remove !,.? characters from the lowercased reply
12 stripped_lowercase = lowercase.strip("!,.?")
13
14 print(stripped_lowercase)
```

Both produce

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
>
 Hello!
hello
>
```

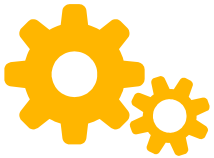
Here's how it works (left to right)



Chaining Examples

```
Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
 >> name = "Princess Anna!!"
 >> name.strip("!").lower()
=> 'princess anna'
 >> name.strip("!").upper()
=> 'PRINCESS ANNA'
 >> name.strip("!").lower().upper()
=> 'PRINCESS ANNA'
 >> name.strip("!").upper().lower()
=> 'princess anna'
 >> name.upper()
=> 'PRINCESS ANNA!!'
 >>
```

A Party Game Bot



This lesson

- Creating lists on-the-fly
- For loop (basic)

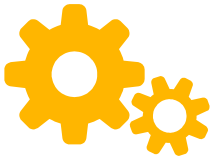
Mind Reader Game Host

This is a 2-player party game, hosted by your chatbot.

Rules

- The 1st player reads a **word**, and secretly enters 3 words they associate with it.
- The 2nd player must then try to guess at least one of the words. If it's a match, they win!

Mind Reader Game



```
1   # How Well Do You Know Me?
2   # Author: Angelica Lim
3   # Date: Jan. 16, 2018
4
5   # This is a 2-player game where you The 1st player reads a word,
6   # and secretly enters 3 words they associate with it.
7   # The 2nd player must then try to guess at least one of the words.
8   # If it's a match, they win!
9
10
11  # Introduce the game
12
13  # Ask the first player to enter 3 words associated with a given word
14
15  # Clear the screen
16
17  # Ask the 2nd player to guess an associated word
18
19  # Check if they match and tell them if they won!
20
```

Mind Reader Game

```
# Mind Reader
# Author: Angelica Lim
# Date: Jan. 16, 2018
# This is a 2-player game where you The 1st player reads a word,
# and secretly enters 3 words they associate with it.
# The 2nd player must then try to guess at least one of the words.
# If it's a match, they win!

import replit

# Introduce the game
print("Welcome to Mind Reader!")

# Ask the first player to enter 3 words associated with a given word
print("Player 1, enter 3 words you think when I say cat:")

# Get the 3 words
first = input("First word:")
second = input("Second word:")
third = input("Third word:")

# Clear the screen
replit.clear()

# Ask the 2nd player to guess an associated word
print("Player 2, what is one word you think Player 1 associates with cat?")
guess = input()

# Check if they match and tell them if they won!
if guess in [first, second, third]:
  print("You've got it!")
```

Try playing it with 2 people! :)

Now, can you think of a way to choose the word randomly?

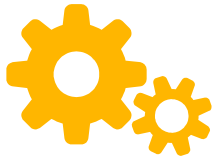
Yes, you can construct lists on-the-fly based on input!

Mind Reader Game

```
# Mind Reader Game
# Author: Angelica Lim
# Date: Jan. 16, 2018
# This is a 2-player game where you The 1st player reads a word,
# and secretly enters 3 words they associate with it.
# The 2nd player must then try to guess at least one of the words.
# If it's a match, they win!

import replit
import random

# Introduce the game
print("Welcome to Mind Reader!")

words = ["cat", "dog", "house", "apple"]
word = random.choice(words)

# Ask the first player to enter 3 words associated with a given word
print("Player 1, enter 3 words you think when I say " + word + "?")

# Get the 3 words
first = input("First word:")
second = input("Second word:")
third = input("Third word:")

# Clear the screen
replit.clear()

# Ask the 2nd player to guess an associated word
print("Player 2, what is one word you think Player 1 associates with " + word + "?")
guess = input()

# Check if they match and tell them if they won!
if guess in [first, second, third]:
  print("You've got it!")
```

What if they got it wrong?

Mind Reader Game



```
import replit
import random

# Introduce the game
print("Welcome to Mind Reader!")

words = ["cat", "dog", "house", "apple"]
word = random.choice(words)

# Ask the first player to enter 3 words associated with a given word
print("Player 1, enter 3 words you think when I say " + word)

# Get the 3 words
first = input("First word:")
second = input("Second word:")
third = input("Third word:")

# Clear the screen
replit.clear()

# Ask the 2nd player to guess an associated word
print("Player 2, what is one word you think Player 1 associates with " + word)
guess = input()

# Check if they match and tell them if they won!
if guess in [first, second, third]:
  print("You've got it!")
else:
  print("No match! They said " + first + ", " + second + " and " + third + "!")

```

How do we play the game multiple times?

Loops

For doing things over, and over, and over…!

http://interactivepython.org/runestone/static/thinkcspy/PythonTurtle/TheforLoop.html

Note: We removed the random statement here. Instead, we'll loop through all the words, one by one.

for **in**

This means: **for** every *word* **in** the *words* list, do the following (indented)

```
# This is a 2-player game where you the 1st player reads a word,
# and secretly enters 3 words they associate with it.
# The 2nd player must then try to guess at least one of the words.
# If it's a match, they win!

import replit

# Introduce the game
print("Welcome to Mind Reader!")

# Create a list of words
words = ["cat", "dog", "apple"]

# Loop through all the items in words
for word in words:

  # Ask the first player to enter 3 words associated with a given word
  print("Player 1, enter 3 words you think when I say " + word)

  # Get the 3 words
  first = input("First word:")
  second = input("Second word:")
  third = input("Third word:")

  # Clear the screen
  replit.clear()

  # Ask the 2nd player to guess an associated word
  print("Player 2, what is one word you think Player 1 associates with " + word)
  guess = input()

  # Check if they match and tell them if they won!
  if guess in [first, second, third]:
    print("You've got it!")

  else:
    print("No match! They said " + first + ", " + second + " and " + third + "!")
```

A Nosy Question Bot

Nosy Question Bot

```
# A Nosy Question Bot
# Author: Angelica Lim
# Date: Jan. 17, 2018
# Description: Asks you from a list of questions.

import random

# Introduction
print("Hello! This is Nosy Question Bot. I'd like to ask you some questions! Here we go.")

# Create questions list
questions = ["Given the choice of anyone in the world, whom would you want as a dinner guest?","Would
  you like to be famous? In what way?", "Before making a telephone call, do you ever rehearse what you
  are going to say? Why?", "What would constitute a “perfect” day for you?"]

# Make responses list
responses = ["Interesting.", "I see!", "Fascinating."]

# Make a loop that will ask the questions from the list
for question in questions:

  # Print the question
  print(question)

  # Get the response
  input()

  # Make a reply
  print(random.choice(responses))

# End it off
print("You are a super interesting person! Thanks for replying to my questions.")
```

Let's review some concepts

What are the **keywords** needed to make a **loop**?

In a loop, what do you need to do to the code that you want to repeat?

True or false? Methods can be **chained** from **left** to **right**.

Week 2 Exercise

Create a ChatBot with a Loop!

Create a chatbot with a loop based on the examples in class in http://repl.it. The chatbot should start when you click on the Run button. It must use at least **2 string methods** for robustness, and a **for loop**. Design your algorithm in English first, then translate it to Python code. Tips: Code bit by bit, and test small pieces as you go. Add your robustness at the end.

Extras

Integer

String, list, Boolean… and now Integer!

http://interactivepython.org/runestone/static/thinkcspy/SimplePythonData/ValuesandDataTypes.html
(Strings and Integers)

Keeping Score

The variable **score** is of type **integer**

```
9   import replit
10
11  # Introduce the game
12  print("Welcome to Mind Reader!")
13
14  words = ["cat", "dog", "apple"]
15  score = 0
16
17  for word in words:
18
19    # Ask the first player to enter 3 words associated with a given word
20    print("Player 1, enter 3 words you think when I say " + word)
21
22    # Get the 3 words
23    first = input("First word:")
24    second = input("Second word:")
25    third = input("Third word:")
26
27    # Clear the screen
28    replit.clear()
29
30    # Ask the 2nd player to guess an associated word
31    print("Player 2, what is one word you think Player 1 associates with " + word)
32    guess = input()
33
34    # Check if they match and tell them if they won!
35    if guess in [first, second, third]:
36      print("You've got it!")
37
38      # Add to their score
39      score = score + 1
40    else:
41      print("No match! They said " + first + ", " + second + " and " + third + "!")
42
43  # At the end, print out their score
44  print("You got " + str(score) + " right!")
45
```

This is how you can add to yourself, or "accumulate"

You can only concatenate strings with strings, so you need to convert your integer to a **string** type using **str()**

Nosy Question Bot

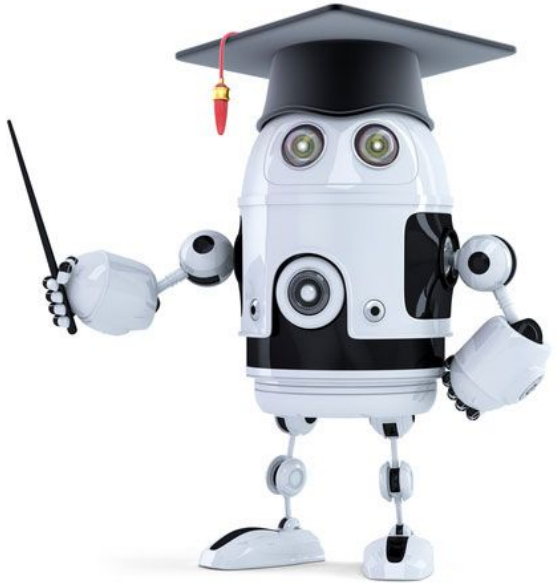
You can give the option to break out of the loop using an **if** statement and **break** keyword

```
# A Nosy Question Bot
# Author: Angelica Lim
# Date: Jan. 17, 2018
# Description: Asks you from a list of questions.

import random

# Introduction
print("Hello! This is Nosy Question Bot. I'd like to ask you some questions! Here we go.")

# Create questions list
questions = ["Given the choice of anyone in the world, whom would you want as a dinner guest?","Would
  you like to be famous? In what way?", "Before making a telephone call, do you ever rehearse what you
  are going to say? Why?", "What would constitute a “perfect” day for you?"]

# Make responses list
responses = ["Interesting.", "I see!", "Fascinating."]

# Make a loop that will ask the questions from the list
for question in questions:

  # Print the question
  print(question)

  # Get the response
  reply = input()

  # If they say stop, break out of the loop
  if reply == "stop":
    break

  # Make a reply
  print(random.choice(responses))

# End it off
print("You are a super interesting person! Thanks for replying to my questions.")
```

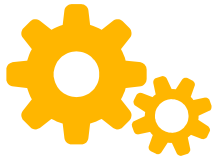
Flashcard Bot

So, you want to help yourself study for a test?

Make a flashcard bot that will test your knowledge and provide you encouragement!



A Training Bot



```
1   # A FlashCard Bot
2   # Author: Angelica Lim
3   # Date: Jan. 17, 2018
4
5   # Description: Asks you from a list of questions. If you type "Done", it
      will exit
6
7   # Introduction
8
9   # Create question list
10
11  # Make a loop that will ask the questions from the list
12
13  # If the user types Done, exit
14
```

Will this produce an error?

```
points = 5
print("You got " + points + "
right!")
```

How do we break out of a loop?