# Many adults can't understand text message abbreviations

- Your challenge is to write a program that will take a text message and translate it into words your grandparents could understand. For example:
  - So funny LOL ROTFL

    becomes
  - So funny laughing out loud rolling on the floor laughing

- For this scenario we will start with a file that lists all the text message abbreviations and the translations

# First we have to break the problem into logical steps

1. Have the user enter the text message

2. Get a list of all the words in the text message

3. Search the translation file for each word in the message and get the corresponding translation

4. Display the translation to the user

# Step 1 – Have the user enter the text message

How do you ask a user to enter a value?

```
input
```

# Step 2 – Get a list of all the words in the text message

Uh oh, we never learned how to do that!
Well we have the whole text message stored in a string variable.

- **Try a Bing search "break sentence into words Python"**
- **Or Bing "Python string methods" to get a list of all the built-in methods for manipulating strings.**

Apparently there is a split method that takes a string and returns a list of all the words in that string, that sounds useful!

```
messageWords = message.split()
```

# Step 3: Search the translation file for each word in the message and get the corresponding translation

This will be the tough part so let's break this down to smaller steps

a. Open the file which contains our translations

b. Extract all the abbreviations from the file and put them in a list

c. Extract all the translations from the file and put them in a list

d. For each word in the text message, go through the list of abbreviations looking for a match

e. When match found get the corresponding translation

# Step 3a: Open the file which contains our translations

How do you open a file?

```
open(fileName)
```

Tip: Any time you are working with files you should open the file and look at it so you know how it is formatted, this will affect how you write the code to read the file.

Our sample file "Translations.txt" looks like this

```
LOL – Laughing out loud
ROFL – Rolling on the floor laughing
```

# Step 3b) Extract all the abbreviations from the file and put them in a list

You will need a loop to read each row from the file

```
for line in iter(fileObject):
#read the file line by line
```

Now we can use the split function again to get the individual words from the line in the file

```
wordsInTheLine = line.split()
```

The file line "LOL – laughing out loud"

Returns the list ["LOL","-","laughing","out","loud"]

# Step 3b) Extract all the abbreviations from the file and put them in a list

- Now, we have the words in the line stored in a list
  - ["LOL","-","laughing","out","loud"]

- The first word is the abbreviation

- How do you ask for the first value in a list?
  - Index position 0

- Add that value to your list of abbreviations

**`allAbbreviations.append(wordsInTheLine[0])`**

# Step 3c) Extract all the translations from the file and put them in a list

- Hmmm we have a challenge here. Because our translation can be made up of multiple words?

  ["LOL",”-”,”laughing”,”out”,”loud”]

- We will need a for loop that retrieves all the words starting from the 3$^{rd}$ word to the end of the list, then concatenate those words together to get the translation.

- How do you know the total number of words in the list?
  - You can use the len() function to get the size of the list containing the words

- You have a list of the words in the message

- Create a loop that will run once for each word in the message

- For each ~~word in the message, look at the position in the list of each~~ word

```
matchPosition = allAbbreviations.index(currentWord)
```

Step 3e) When a match is found display the corresponding translation

- If you found a match in the list of abbreviations, get the value from the translation list that has the same index position

- Add that translation to a string that will contain your final translated message

# Step 4. Display the translated message to the user

- This part you know how to do!
- Use a print statement

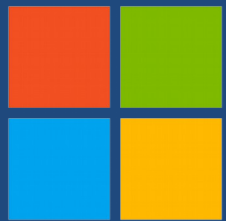# DEMO

Walkthough completed solution

# Are you completely lost?

- Try coding the solution one step at a time

- Test the code as you go

- If you get stuck on a step, open the solution file, read through the code. It is broken into the same steps. See if you can understand the code. Then go back and try to write it yourself.

- You will discover that at first you will need to read a lot of examples, you will get better at writing your own code the more you practice!

# Congratulations!

- You now know how to break a complex problem into parts
- Now you can code and test one part at a time so you can create elaborate programs to solve complex programs