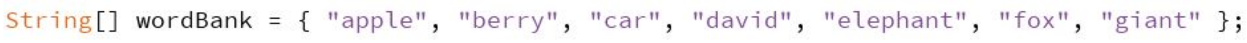
Typing Game

This game pulls together a lot of what you've learned in previous lessons. The goal is to create a game where the player has to type the words that appears on screen. They have a limited amount of time to guess as many words as possible.

You will need to work a lot of the details out yourself. Don't forget to refer back to earlier lessons if you can't remember how to do something!

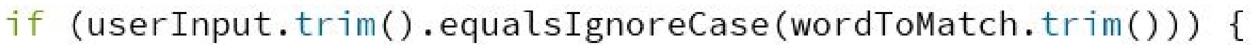
Details

1. You will need a String array to store your words in. These are the words the player will have to guess. Ultimately you will want to load these words from a file (look back to **lesson 4** for a reminder), but this will get you started:



```
String[] wordBank = { "apple", "berry", "car", "david", "elephant", "fox", "giant" };
```

2. Use "void keyTyped()" to allow your player input text (look back at **lesson 2** for a reminder on how this works).
3. You should have a "wordToType" string. Use a random number to select a random word from your wordBank - this will be your first wordToType. This should display on the screen, for the player to type.
4. When the player hits enter you will need to compare the userInput with the wordToMatch - if they're the same then add to the player's score (you will need a **playerScore** int) and pick a new word. Comparing can be tricky - what if the player types a capital letter, or adds an extra space? If you like, you can ignore them using this code as a starting point (**trim()** removes spaces at the start or end of a string):



```
if (userInput.trim().equalsIgnoreCase(wordToMatch.trim())) {
```

5. Once you have the starting point in place you can add a timer (look back at **lesson 6** if you need a reminder on how this works).
6. Add in the file loading mentioned in point 1 (you will need to find a list of words somewhere - try googling "list of words text file", and see if you find anything useful!)