

# Lesson 5: Keyboard input

So we've done keyboard input before: moving shapes on the screen using the keyboard.

## Quick refresher of the important keywords:

<code>keyPressed</code>	true when any key is pressed.
<code>key == 'a'</code>	true <b>only</b> when the lowercase a is pressed.

## Test your knowledge!

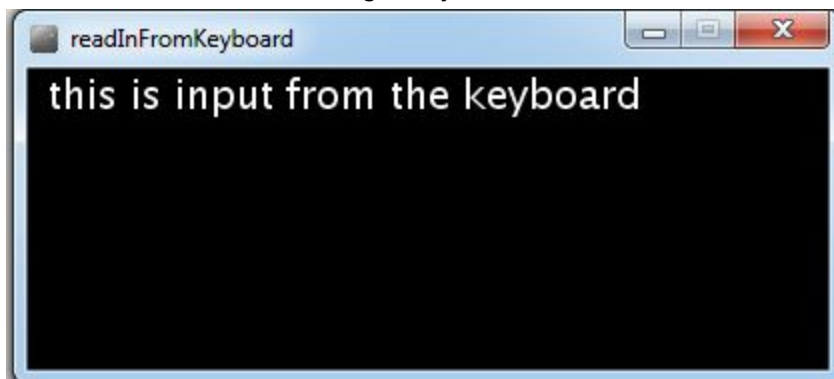
- Make a program in 5 mins that moves a square left and right using the (a) and (d) keys.

If you finish early:

- Add up (w) and down (s).
- Import a picture as the background.
- Change the shape from square to triangle.

## What are we going to do?

Make a program to show text from the keyboard on the screen, in the same way to everything from Microsoft Word to Google in your browser does!



## What do we need?

1. A **String** variable to store the keyboard input.
2. A way to show the message from the keyboard on the screen.
3. A way to add the pressed key to the **String** variable.

## A New Variable Type: String

**String** is a variable type - like **int** and **boolean**.

Rather than containing numbers (like **int**) or true/false (like **boolean**), **String** variables contain **text**. A **String** is everything between two quote marks. eg :

```
String myString = "This is 1 String";
text(myString , 50,50);
```

## 5 minute Task:

- Try the above code in a new program.
- Change the string that is shown on the screen.

## Void Keypressed

For part 3 we will be using something new - `void keyPressed()`

Why use `void keyPressed()` rather than `if(keyPressed)`?

`void keyPressed()`, is a separate **{codeBlock}** (like `void draw()` and `void setup()`) and only happens once per key release, that means every time we press a key we only get one new letter - rather than one letter every time void draw executes and we have our finger on that letter (even if you're super fast that is a LOT of times!)

### Example code:

```
String keyboardInput = ""; //New EMPTY string variable

void keyPressed() {
    keyboardInput += key;
}
```

### Task:

- In a new program copy the above code. (You need to add `void setup` and `draw`)
- Add a way to show the `String` to the screen to this program (as you did earlier)

### New lines!

1. Re-open you first program (that just showed a string). Now add a new line somewhere in that sentence by adding `"\n"` in the middle of the string, where you want a new line. `"\n"` is the symbol for new line. See?
2. Now add this into the keyboard program:
  - a. You need to check `if` the `key` pressed is equal to `ENTER`: then add a new line character (which we just learnt is `"\n"`). Otherwise add whatever key was pressed.

### Extra - if you want to add backspace add this check for new keys:

```
if (key == BACKSPACE && keyboardInput.length()>0) {
    keyboardInput= keyboardInput.substring(0,keyboardInput.length()-1);
}
else{ // do normal stuff for adding a new letter.
```

**Note** `keyboardInput` is whatever name you gave the the string variable

**If you are finished early:**

- Change the size of the text.
- Change colour of the text
- Add extra functionality using the mouse - to:
  - Make the text bigger on mouse RIGHT click
  - Make the text smaller on mouse LEFT click