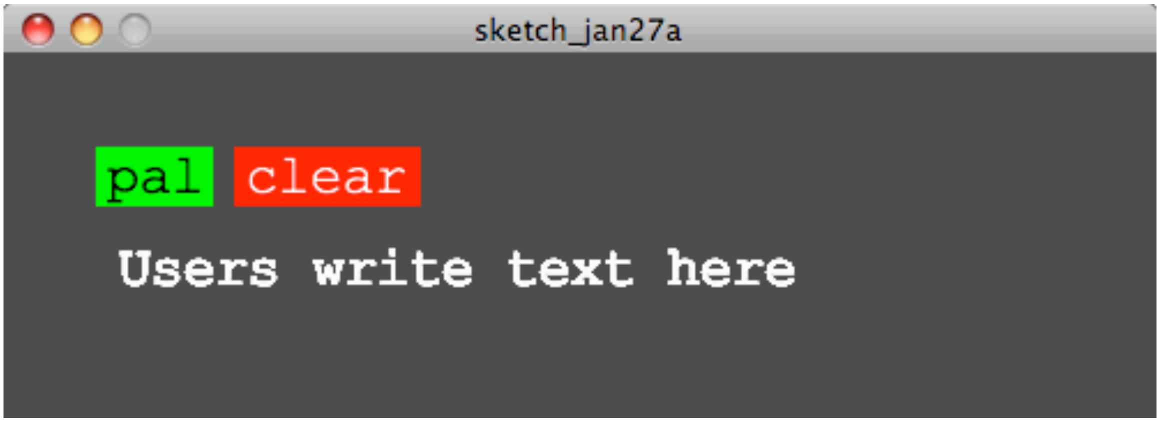
Homework 12: Check For Palindromes

Goal: To write a simple Processing program that takes text input and displays it dynamically on the screen.

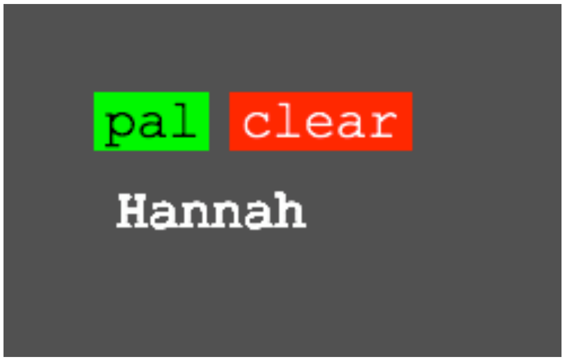
The App

In this assignment you will create a simple application, as shown below. There are two buttons, and as the user types text, it appears in a line below the buttons. The font is called CourierNewPSMT-24.vlw and is recommended because it looks like an old school typewriter.

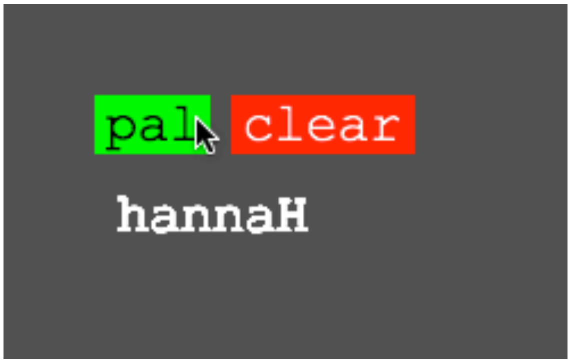


Users write text here

After typing some text, the user can click the pal button and the same text is shown in reverse. Obviously, clicking it again shows the text in its original form. For example, the following sequence shows the action.



Hannah



hannaH

Clicking the clear button deletes the text, and sets up to accept more text. That’s it.

Palindromes

Palindromes are words like *kayak* and *radar* or phrases like *Madam, I’m Adam* that read the same going forward as backward. (We ignore spaces and punctuation.) They are unimportant, but they give us a good excuse to build an app that reverses text.

Assignment: Implement the application described above for showing text in reverse. The following are suggestions to make this task easier:

a) Set up your page, loading the font as described in class.

b) Draw the buttons and write the text on them.

c) READ the "Strings & Arrays" resource – you can't do this assignment without knowing that material. [Remember, we have no book, so the Resource pages are how we teach specific facts you need to know.]

d) Declare two string array variables, say `datain` and `nothing`, as in

```
String[] datain = {""}; //Start array out with 1 element
String[] nothing = {""};//Array has only 1 element
```

Notice the use of the capital letter on `String`, the square brackets indicating that it is an array and the initialization of a single string element, which is the empty string containing no letters. You will need one more declaration of a string,

```
String reform = "";
```

e) Develop the `keyPressed( )` block in which you append to the `datain` array each new character that is typed. Because you are adding data of type characters to an array whose elements are type `String`, you must convert the key character to a string, as in `str(key)`. You will use the function `append( )`, described in the references.

f) The text is printed out using the usual `text( )` function, but it needs a string, not a string array as its first argument. So, before printing you will need a `for`-loop to sweep through the array, and concatenate (it's like appending, but for strings) each item in the array using the + sign, as in

```
reform = reform + datain[i]; //Put ith item at end of string
```

When the loop is finished, print text using text(reform, <*x pos*>, <*y pos*>).

g) Finally, you will develop the `mousePressed()` block. This block takes care of mouse clicks. You need to figure out two cases, (a) when the mouse clicks on the "pal" button, and (b) when the mouse clicks on the "clear" button. This is straightforward by looking to see if `mouseX` and `mouseY` are within the area where you drew the buttons; if not, there is no effect. In case (a), simply reverse the `datain` array and display it as usual; use the `reverse( )` function listed in the references. In case (b), set `datain` to `nothing`, and display it as usual.

Wrap Up

You have written a simple program that reverses a string of letters to see if it makes sense to you, i.e. if it is a palindrome. You converted between characters and strings, and between strings and an array of strings. These data manipulations help us to use data in different ways.

Turn In

Rename your program `<yourname>.pde` and turn it in to the class drop box.