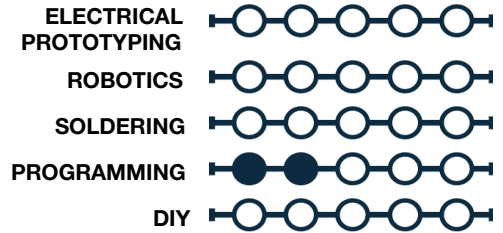


Fonts and Text

Aa
BB
Cc



PROCESSING
SKILL REQUIREMENTS
DIFFICULTY 1-5



MATERIALS LIST

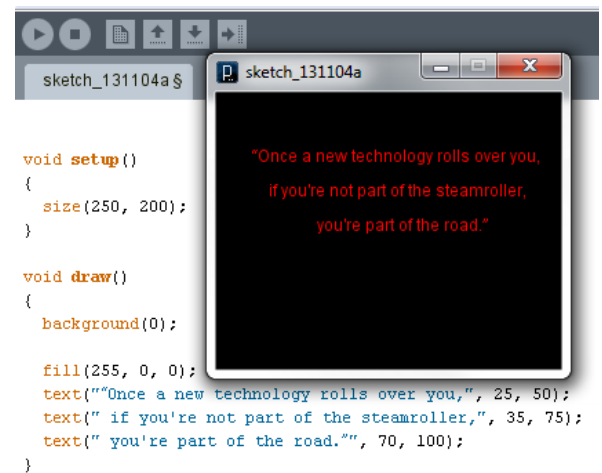
- Computer
- Processing

STEP 1: Text

The `text()` function allows you to add text to your Processing sketches. The basic syntax for text is `text("string", X, Y);` where "string" is the text you want to display.

There are a few useful modifiers for the basic `text()` function before we get into fonts. Here's their syntax:
`textSize(#);` sets the text size in pixels, and
`textAlign(ALIGN X, ALIGN Y);` sets the alignment of the X and Y, using the built-in alignments of `LEFT`, `CENTER` and `RIGHT` for X alignment; and `TOP`, `BOTTOM` and `CENTERLINE` for Y.

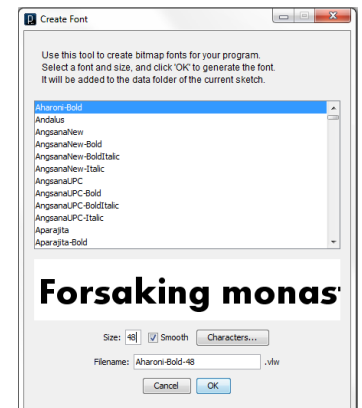
For an example use of the `text()` function, see the sample images. We will be using a quote from Stuart Brandt in this HotSheet: *"Once a new technology rolls over you, if you're not part of the steamroller, you're part of the road."*



STEP 2: The Create Font... Tool

Regular text is fine and dandy, but how do we change the font? We have the **Create Font...** tool for that; you can find it under the Tools pull-down menu. When you select the tool, you will see a window that looks like this.

Similar to any font selector tool you find in a word processing application, you can select the font, but this tool allows you to select italics, bold, etc., in the same selection. You can then select the font size. Once you have selected those options you can then re-name your font and click OK. I usually name this something like "myFont." This creates a font that is placed in your data folder for you to call out in your sketch, similar to how you would an image.



STEP 3: Adding a Font to your Sketch

Once you have created a font file, you'll add it to your sketch in a similar way as adding an image. We use the `PFont` object class and name the font before our `setup()` code. Since we are only using one font, we'll just call it 'font'. Within our `setup()` we assign the font to the object name using the `loadFont()` function. Within the `loadFont()` function, we name the font file we created in quotes, make a note that all Processing fonts end in the file format of `.vlw`. See the code in step 4 for an example.

STEP 4: Setting a Font

Now that we have added the font to our sketch we can use it! Set the font for the text before the `text()` that you want to use, just like any other modifiers. To set the font you use the `textFont()` function and add the name of your font within the function. If you wanted multiple fonts within a sketch, you would add each one and then use the `textFont()` function each time you change the font.

```
sketch_131104a $
PFont font;

void setup()
{
  size(250, 200);
  font= loadFont("myFont.vlw");
}

void draw()
{
  background(0);

  fill(255, 0, 0);
  textFont(font);
  text("Once a new technology rolls over you,", 25, 50);
  text(" if you're not part of the steamroller,", 35, 75);
  text(" you're part of the road.", 70, 100);
}
```

STEP 5: A Simple Example

One of the best uses for text is to be able to visualize data. To create an example of this, we'll build a simple bar graph sketch, graph the `mouseX` and `mouseY` values and display their values in text.

STEP 6: The Rectangles

We are using the same sketch as before, so all of our font information has already been created. We are now going to create two rectangles; you can see the code and how they are grouped with their modifiers in the example to the right. The width of the rectangles are `mouseX` and `mouseY` respectively. When you run the sketch, the rectangles will grow or shrink depending on where your mouse is on the screen. We set each rectangle as a different fill color.

STEP 7: The Text

We can visually show these values using the text function. Instead of a string of text in quotes I placed the data I wanted to display in the first parameter, and then gave an X and Y value that centered the text over a bar graph. I finally added the font at the top of the sketch, since I wanted both readouts to use the same font.

```
sketch_131104a $
PFont font;


void setup()
{
  size(250, 200);
  font= loadFont("myFont.vlw");
}

void draw()
{
  background(0);

  textFont(font);

  fill(255,0,0);
  rect(0,50,mouseX,50);
  fill(255);
  text(mouseX,50,75);

  fill(0,255,0);
  rect(0,150,mouseY,50);
  fill(255);
  text(mouseY,50,175);
}
```



STEP 8: Share your work!

If you would like to share your work - which you do - you can sign up at openprocessing.org. Open Processing allows you to share your code and view the drawing online, and allows others to fork or borrow your code to manipulate and change it on their own, while still keeping your sketch intact for others to check out.



TAKING IT FURTHER

- Try using math and variables to make interaction more intricate.
- Make a snowman run around in your holiday image.