Text

Shiffman Chpts. 17 and 18

The *String* is the Thing

- Can use char type for a single character
  - char alpha = 'a';
- Can use a char array to represent text, but String is more powerful

String text = "This is a string of text.";

- Representation is similar to an array (diagram)
- *String* is a class
  - Methods and Data

The *String* is the Thing

- String methods:

  charAt(<int loc>); //returns character at loc

  length(); //returns number of characters

  toUpperCase();

  equals(<string>); //use to compare two strings

Text Display in the Console

- As we've seen, println() can be used to display text in the console
- Useful for debugging
- Not really intended for end user

String Behaves like an Array of Characters

- Must use .charAt(<index>) to access individual characters
  - Replaces [<index>] for an array
- String class provides many methods beyond an array

Accessing Characters in a String

```
String sent1 = "This is a fine looking sentence.";
//retrieving particular characters
char a = sent1.charAt(0);
char b = sent1.charAt(1);
println(a);
println(b);
String fragment = "even better";
//print every character in a string
for(int i =0; i<fragment.length(); i++)
{
  println(fragment.charAt(i));
}
```

Comparing Strings with .equals()

```
String w1 = "a baby";
String w2 = "a baby";
if(w1.equals(w2))
{
  println("Method check works!");
}
else
{
  println("Method check doesn't work.");
}
```

Changing Case with toUpperCase()

```
//NOTE: toUpperCase returns an upper case version of the
//string
//Calling it on its own is not sufficient
w1.toUpperCase();
println(w1);

//must assign the returned value to the variable to change it
w1 = w1.toUpperCase();
println(w1);
```

Manipulating Strings

- indexOf(<searchCharacters>)
  - returns the location of *searchCharacters* in a string
- substring(<beginIndex>, <endIndex>)
  - Returns a substring spanned by the given index values
- +
  - Concatenate two strings

Manipulating Strings

```
String sent1 = "This is a fine looking sentence.";  //get the location of the string "a"
 int locOfA = sent1.indexOf("a");
 //get the first part of the string
 String partA = sent1.substring(0, locOfA+1);
 //get the tail of the string
 String partB = sent1.substring(sent1.indexOf("looking"), sent1.length());
 //concatenate to make a new string
 sent2 = partA + "n " + fragment + " " + partB;
```

Examples

Other useful commands

- split()
  - Split a string into individual words
  - See Shiffman 16.2
- loadStrings()
  - Read text from a file
  - Shiffman 16.3
- saveStrings()
  - Save text to a file

Learning More

- Some information on processing.org
- "String" is actually a Java class
  - More information in the java documentation

Text Display in Main Window

1. Choose a font and add it to the project
   - Tools->Create Font
   - Creates a font file and adds it to your project (i.e. the folder that holds your sketch)
2. Declare font object
   - PFont f;
3. Load the font file you created in step 1.
   - f = loadFont("ArialMT-16.vlw");

Text Display in Main Window

4. Specify the current font
   - textFont(f, 36);//36 is an option font size
5. Specify a color
   - fill(0);
6. Display the text
   - text("This is text!", <x>, <y>);

Examples

Good Example of Aliasing

- 100%

- 400%

Alternative with a System Defined Font

- If you are using a font that is contained in your OS, you can do the following:

PFont f = createFont("Georgia", 24, true);