

```

1: /*-----
2:  Displaying live YTD stock prices using Yahoo as data source.
3:
4:  Author: Michael Goldwasser
5:
6:  This project is derivative of an original example accompanying
7:  the book:
8:
9:  Processing: Creative Coding and Generative Art in Processing 2
10: By Ira Greenberg, Dianna Xu, and Deepak Kumar
11: Friends of Ed (An APress Company), 2013
12: ISBN-13 978-1430244646
13: Please refer to the associated README for a full disclaimer.
14:
15: Copyright (c) 2013, Friends of Ed (An APress Company)
16: All rights reserved.
17: -----*/
18:
19: String symbol = "GOOG"; // initial choice
20:
21: class Entry {
22:     float price; // we'll use adjusted close
23:     int volume;
24:     int month;
25:     int date;
26:     int year;
27: };
28:
29: Entry[] quotes;
30: float minPrice, maxPrice;
31: float increment = 10; // how can this be better tailored?
32:
33: float X1, Y1, X2, Y2;
34:
35: PFont legendFont = createFont("SansSerif", 20);
36:
37: void setup() {
38:     // drawing setup
39:     size(800, 600);
40:
41:     X1 = 50;
42:     Y1 = 50;
43:     X2 = width - 50;
44:     Y2 = height - 50;
45:     textFont(legendFont);
46:
47:     refresh();
48: } // setup()
49:
50: // used to (re)load stop information and to render the graph
51: void refresh() {
52:     symbol = symbol.toUpperCase();
53:     loadData();
54:     drawGraph();
55:     symbol = ""; // reset; can use keyboard to enter a new symbol
56: }
57:
58: void loadData() {
59:     // Load the historical data file...
60:     String url = "http://ichart.yahoo.com/table.csv?s=";
61:     url += symbol;
62:     url += "&a=0&b=1&c=1900"; // start date
63:     url += "&d=" + (month()-1);
64:     url += "&e=" + day();
65:     url += "&f=" + year();
66:     url += "&ignore=.csv";
67:     String[] lines = loadStrings(url);
68:     if (lines == null) {
69:         println("Unable to load information about symbol: " + symbol);
70:         quotes = null;
71:     } else {

```

```

72:     lines = reverse(lines); // Yahoo returns newest first
73:
74:     // How long is the dataset
75:     quotes = new Entry[lines.length-1]; // subtract one to ignore header line
76:
77:     // Parse the needed data
78:     for (int i=0; i<quotes.length; i++) {
79:         // Frst split each line using commas as separator
80:         String[] pieces = split(lines[i], ",");
81:
82:         // get the closing price of stock, and month
83:         quotes[i] = new Entry();
84:         quotes[i].price = float(pieces[6]);
85:         quotes[i].volume = int(pieces[5]);
86:         String date = pieces[0];
87:         String[] temp = split(date, '-');
88:         quotes[i].year = int(temp[0]);
89:         quotes[i].month = int(temp[1]);
90:         quotes[i].date = int(temp[2]);
91:     }
92:     println("Data Loaded: "+quotes.length+" entries.");
93:
94:     // determine min and max stock price for the year
95:     minPrice = maxPrice = quotes[0].price;
96:     for (int k=1; k < quotes.length; k++) {
97:         minPrice = min(minPrice, quotes[k].price);
98:         maxPrice = max(maxPrice, quotes[k].price);
99:     }
100:     println("Min: "+minPrice);
101:     println("Max: "+maxPrice);
102: }
103: }
104:
105: void drawGraph() {
106:     background(0);
107:
108:     // draw plot bounding box
109:     rectMode(CORNERS);
110:     noStroke();
111:     fill(255);
112:     rect(X1, Y1, X2, Y2);
113:
114:     // draw graph of stock prices
115:     if (quotes != null) {
116:         stroke(0);
117:         strokeWeight(1);
118:         beginShape();
119:         for (int i=0; i < quotes.length; i++) {
120:             float x = map(i, 0, quotes.length-1, X1, X2);
121:             float y = map(quotes[i].price, minPrice, maxPrice, Y2, Y1);
122:             vertex(x, y);
123:         }
124:         endShape();
125:
126:
127:         // draw legend
128:
129:         // Load the full name of the stock for caption
130:         String url = "http://finance.yahoo.com/d/quotes.csv?f=n&s=";
131:         url += symbol;
132:         String full = loadStrings(url)[0];
133:         full = full.substring(1, full.length()-1); // remove quotation marks
134:         String caption = "(" + symbol + ") " + full;
135:
136:         // title
137:         fill(255);
138:         textSize(18);
139:         textAlign(LEFT);
140:         text(caption, X1, Y1 - 10);
141:         textSize(10);
142:         textAlign(RIGHT, BOTTOM);

```

```

143:     text("Source: Yahoo! Finance (finance.yahoo.com)", width-10, height-10);
144:
145:     // axis labels
146:     drawXLabels();
147:     drawYLabels();
148: }
149: }
150:
151: void drawYLabels () {
152:     fill(255);
153:     textSize(10);
154:     textAlign(RIGHT);
155:     for (float i=minPrice; i <= maxPrice; i += increment) {
156:         float y = map(i, minPrice, maxPrice, Y2, Y1);
157:         text(nf(i, 0, 1), X1-10, y);
158:         stroke(255);
159:         line(X1-5, y, X1, y);
160:         stroke(0);
161:         line(X1, y, X2, y);
162:     }
163:     textSize(18);
164:     text("$", X1-40, height/2);
165: } // drawYLabels()
166:
167: void drawXLabels() {
168:     fill(255);
169:     stroke(0);
170:     textSize(10);
171:     textAlign(CENTER);
172:
173:     boolean multiyear = (quotes[0].year != quotes[quotes.length-1].year);
174:
175:     String prev = "";
176:     for (int i=0; i<quotes.length; i++) {
177:         String label = getLabel(quotes[i], multiyear);
178:         if (!label.equals(prev)) {
179:             prev = label;
180:             float x = map(i, 0, quotes.length, X1, X2);
181:             text(label, x, Y2+10);
182:             strokeWeight(0.3);
183:             line(x, Y2, x, Y1);
184:         }
185:     }
186:     textSize(18);
187:     textAlign(CENTER, TOP);
188:     text("Trading Date", width/2, Y2+10);
189: } // drawXLabels()
190:
191: String getLabel(Entry e, boolean multiyear) {
192:     if (multiyear) {
193:         return "" + e.year;
194:     } else {
195:         return "" + e.month;
196:     }
197: }
198:
199: // this is required to allow user interactions
200: void draw() {
201: }
202:
203: void mouseClicked() {
204:     save("Chart.jpg");
205: }
206:
207: void keyPressed() {
208:     if (key == ENTER || key == RETURN) {
209:         refresh(); // load and graphd data for current symbol
210:     } else if ((key >= 'a' && key <= 'z') || (key >= 'A' && key <= 'Z')) {
211:         symbol += key;
212:     }
213: }

```