HOW TO VISUALIZE DATA *with* JAVASCRIPT

WITH ELISABETH ROBSON

Austin • Annapolis • Seattle

wickedlysmart.com

Contact: wickedlysmart.com

Weather and Climate

Generally speaking, both weather and climate are about things like temperature, humidity, pressure, high and low temperatures, and so on. Weather is about the measurements of those values now, while climate is about the measurements of those values over long periods of time.

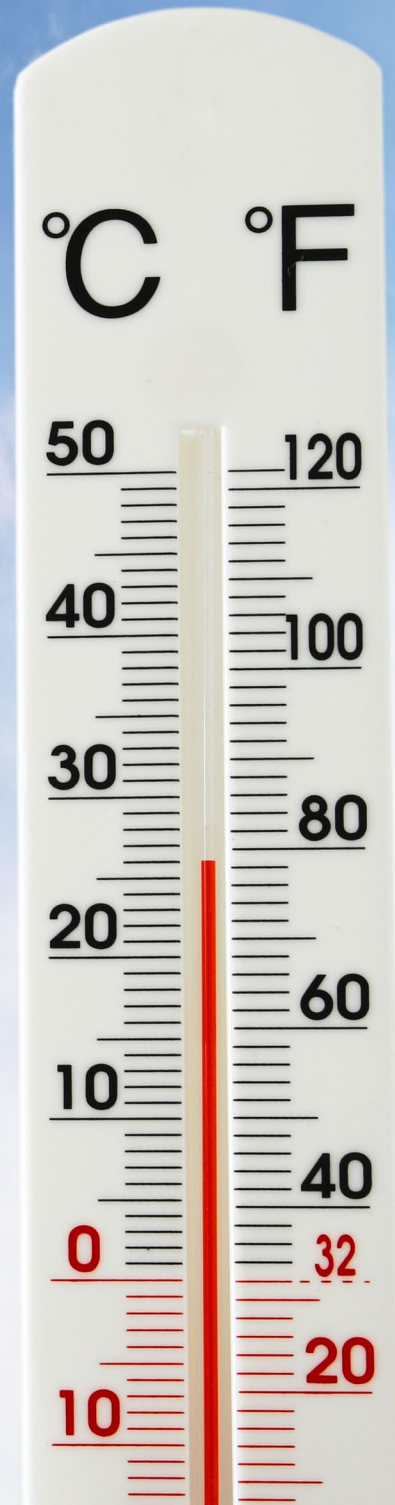
NOAA, the National Oceanic and Atmospheric Association, defines weather as *"The state of the atmosphere with respect to wind, temperature, cloudiness, moisture, pressure, etc. Weather refers to these conditions at a given point in time (e.g., today's high temperature)."*

And they define climate as *"The long-term pattern of weather conditions at a location."* By "long-term" NOAA typically means thirty years or more; what we need is a long enough period of time to establish the normal range of conditions at a given location.

You can probably see that sometimes there's some overlap in weather and climate. After all, long-term patterns of weather conditions can help inform what kind of weather to expect at your next summer solstice party.

Weather: short-term state of conditions in the atmosphere

Climate: long-term patterns of weather conditions



What is Climate Change?

Climate change is a change in global or regional climate patterns.

Climate scientists study weather conditions over the long-term to find patterns, see if those patterns are changing, and project how the patterns might change in the future.

For instance, we can look at a regional history of high temperatures in the summer to estimate how hot it will be next summer. If the pattern of high temperatures changes—perhaps unusually hot summer nights are becoming more common—this trend indicates the summer temperature in that region might be increasing.

Not too long ago (on Earth's timescale anyway—a mere 55 million years ago), the planet was having a *long* heat wave. It was so warm for so long, there were palm trees growing in the Arctic. Climate scientists want to know: what caused this heat wave? Why did the heat wave end? Will there be another one? Are humans having an impact on how soon that might happen?

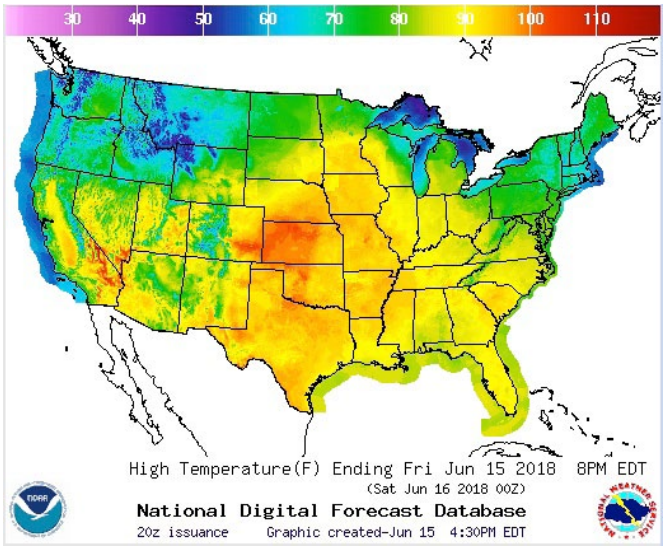
Climate scientists strive to answer that question by looking at data from the past (ice cores, lake sediments, leaf fossils, and more) and they project future trends based on past and current observations.

How do data visualizations help?

Data visualizations are representations of data in pictorial or graphical format. Sometimes we draw a simple line graph, like this graph of global surface temperature; sometimes we use maps, like the high temperature map for the United States for June 15, 2018. Other types of visualization include bar graphs, scatter plots, pie charts, bubble charts, heatmaps, treemaps, cloropleth maps, sankey diagrams, and more. Even art can visualize data.

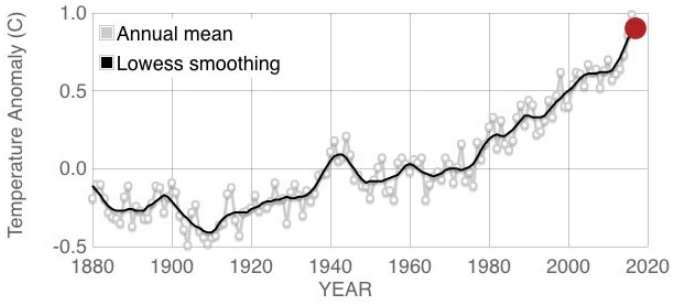
Visualizations help because they convey a lot of information quickly, and can expose patterns that aren't always visible in raw data sets, which are typically tables of numbers. If you want to know which area of the United States is warmest on June 15, 2018 using the raw data, you'd have to do a painstaking review of the recorded high temperatures and compare them all. However, if you look at a high temperature map visualization you can quickly see that the warmest temperatures are in the southwest region (California and Nevada), and in the midwest (Kansas, Oklahoma, and Colorado).

High temperature map



Global surface temperature

Data source: NASA's Goddard Institute for Space Studies (GISS). Credit: NASA/GISS

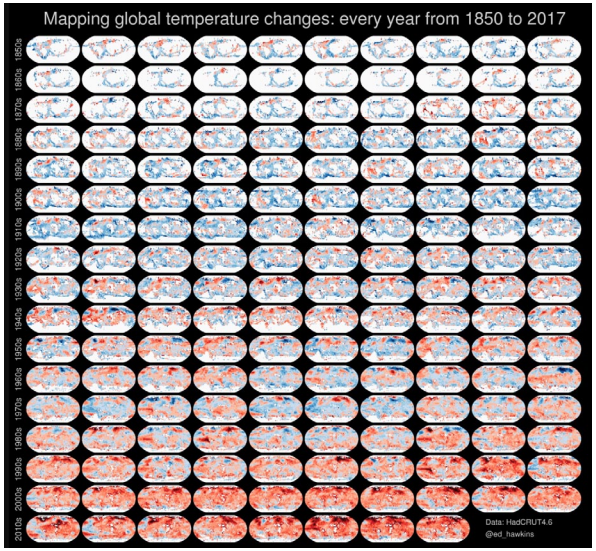


Do the Work: Explore Visualizations of Climate Change Data

Each of the visualizations below depicts the change in annual global temperatures from 1850 or 1880 to 2017. While the visualizations depict essentially the same data, each does so in a different way, using a different visualization technique. Review the visualizations below and consider the following questions:

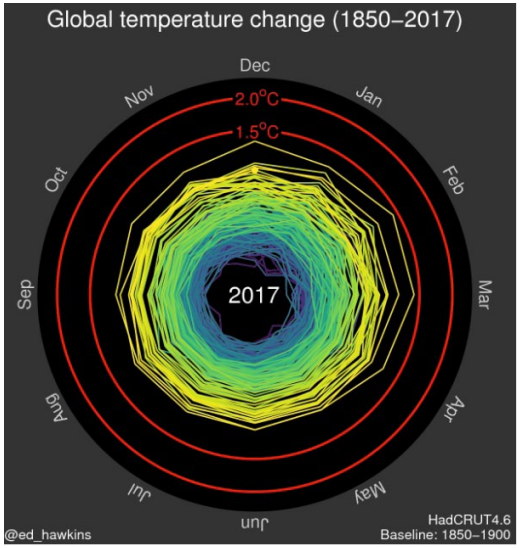
	Circle your answer			
Which visualization catches your eye first?	A	B	C	D
Which visualization best conveys the information that the global temperature is warming?	A	B	C	D
Which makes the most effective use of color?	A	B	C	D

A:



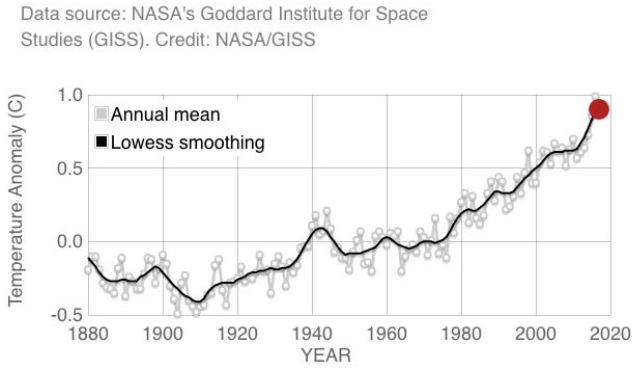
Small multiples map chart by Ed Hawkins

B:



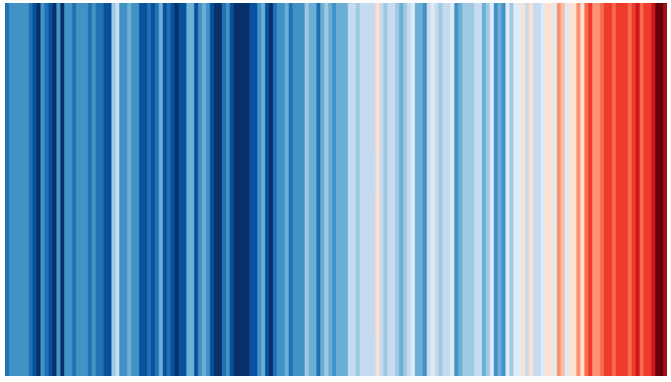
Spiral animation by Ed Hawkins

C:



Line graph by NASA

D:



Color band chart by Ed Hawkins

Do the Work: Explore the Data

We're going to build a visualization of monthly temperature anomaly data from NOAA.

First, let's understand what a temperature anomaly is, and then we'll take a look at the data. Temperature anomalies are a useful measure in climate science because they tell you how much above or below normal a temperature is. Let's say the normal high temperature for December averages around 38 degrees (F) where you live. If you get up tomorrow and it's 35 degrees, then you've got a temperature anomaly of -3 degrees. If, however, you get up tomorrow and it's 45 degrees, then you've got a temperature anomaly of +7 degrees.

With that in mind, let's look at the temperature data from NOAA, which you can find online at https://www.ncdc.noaa.gov/cag/global/time-series/globe/land_ocean/p12/12/1880-2017.csv

If, for some reason, that file isn't available, click here to access our copy of the data.

The measurements are in degrees Celsius.

```
Global Land and Ocean Temperature Anomalies
Units: Degrees Celsius
Base Period: 1901-2000
Missing: -999.0000
Year,Value
188001,-0.00
188002,-0.12
188003,-0.14
188004,-0.05
188005,-0.07
188006,-0.18
...
201707,0.83
201708,0.81
201709,0.78
201710,0.73
201711,0.76
201712,0.80
```

The base period tells us how "normal" is measured. We'll look at this in more depth in a moment...

The temperatures start in 1880, in January (01). The value (-0.00) is the temperature anomaly for the entire month. More on this in a moment, too...

We've got data up through December (12), 2017.

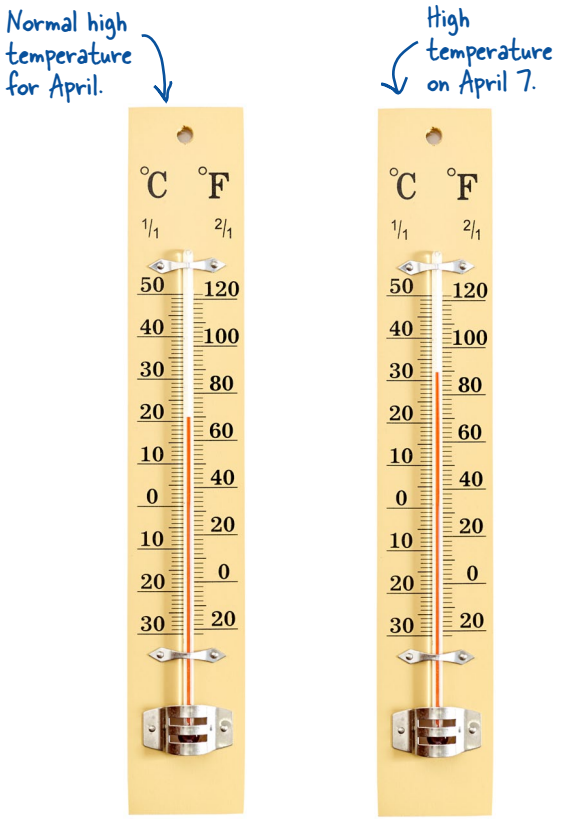
201712,0.80

Year Month

Temperature anomaly (scientists use Celsius, so this is 0.80C).

Practice with temperature anomalies

If the normal April high temperature in your town is 70 degrees Fahrenheit, what is the temperature anomaly for April 7 if the high temperature that day is 89 F?



Your answer:

Questions about the data

Before we go any further we should, like any good scientist would, ask a few questions:

What does this data represent?

What does "base period" mean?

And who's measuring these temperatures?

The temperature data

The data is a combination of land and sea surface temperatures, averaged over the entire globe. Some areas have better temperature data than others so the average temperature might be computed directly using measurements, or computed indirectly using sparse data from the area and estimated from other nearby regions. All these regional averages are combined and compared to "normal", and the difference is the anomaly. Do that every month since 1880, and you get the data in the file you downloaded from NOAA.

The base period

The base period tells us how to measure "normal" so we know how to calculate the anomaly. The file header tells us that NOAA used the average temperatures of each month from 1901-2000 to compute "normal".

So, to compute the normal temperature for January, we would take the average temperature for January 1900, January 1901, January 1902... January 2000 and compute the average of all those averages! That is then the normal temperature for January. Do the same thing for every month and you have the normal for each month.

Why pick those years as the base period? Those years are familiar to most people alive today, and that range establishes a fairly long term average for comparison purposes. All we need is some sensible definition of "normal" to see long term trends.

Who's measuring these temperatures?

That's a great question. Measuring temperatures accurately is an incredibly complex science. Back in the day, sailors used to measure ocean temperatures by dipping a bucket in the ocean and taking the temperature of the water in the bucket. Needless to say, we have much more sophisticated temperature gathering tools these days, including high-tech buoys that are constantly keeping track of ocean temperatures all over the globe, and weather stations in all corners of every country.

Analyze the data

Make sure you've downloaded the JavaScript data file gta_data.js:

https://www.ncdc.noaa.gov/cag/global/time-series/globe/land_ocean/p12/12/1880-2017.csv

Once you have, open it up and take a look. Pick out all the temperature anomalies for January, for the years below. Write them in to the spaces below, and see if you notice a trend:

January 1880:	_____	January 1960:	_____
January 1900:	_____	January 1980:	_____
January 1920:	_____	January 2000:	_____
January 1940:	_____	January 2017:	_____

Do the Work: Create the visualization

If you want a fun break before you begin coding the visualization using HTML, CSS, and JavaScript, you can create the visualization by hand by doing this exercise.

All you need is a set of coloring pencils in a range of colors that match (or, close enough) the color legend on the chart.

Use your reds and oranges for the hottest months (indicated with 1, 2, 3 and so on), your blues for the coolest months (indicated with 8, 9, 10) and your yellows and greens for the months in between (4-7).

This visualization was inspired by Ed Hawkins , Zachary Labe, and Brian Foo.

Ed Hawkins (@edhawkins) is professor of climate science at the University of Reading, and principal research science at NCAS, and he is known for creating compelling visualizations of climate data.

Zachary Labe (@ZLabe) is climate scientist studying the Arctic who regularly creates visualizations for the data he's studying.

Brian Foo (@beefoo) is an artist and computer scientist, and is the author of The Climate Change Coloring Book.

Create the visualization by hand

Color the months, where 1 is hottest (the largest positive temperature anomaly) and 10 is coolest (the smallest positive temperature anomaly). Use the colors in the color legend, so use purple for cooler values (months with a 10) and red for hotter values (months with a 1). See if you notice any trends.

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1990			8									
1991												
1992												
1993												
1994												
1995		8										
1996												
1997												9
1998		4		9	9	10	5	6				
1999		10										
2000												
2001											10	
2002	7	5	5									
2003	8								10	5		6
2004		6	10								5	
2005	9		9	8	8	9	9		8	7	7	8
2006								10	9	9		5
2007	3	9		7	10							
2008			7							10		
2009	10					8	10	5	6			10
2010	6	7	4	3	5	5	4	9			3	
2011				10								
2012				6	7	6	7	8	5	6	8	
2013					6	7	8	7	7	8	2	7
2014	5		6	4	4	4	6	4	3	2	9	2
2015	4	3	3	5	2	2	3	2	1	1	1	1
2016	1	1	1	1	1	1	1	1	2	3	6	4
2017	2	2	2	2	3	3	2	3	4	4	4	3

Do the Work: Experiment with Color Tools

Color is an important part of almost every visualization, and is an effective way to represent quantities (like the ranking of a top 10 hottest month).

To help choose a good color scheme for your visualizations, we recommend that you explore the many online tools for selecting colors schemes. One of our favorites is Color Brewer.

For the top 10 hottest months visualization, we chose a sequential color scheme from blue (for cooler) to red (for hotter). We used Color Brewer to pick colors from a multi-hued blue scheme, and a multi-hued red scheme and joined them together to create the colors for the visualization.

Choosing good colors for a visualization is both an art and a science. When you are ready to begin creating visualizations on your own, we recommend reading more about selecting colors and color schemes for visualizations. One short guide you can begin with is Use of Color in Data Visualization by Robert Simmon, from NASA.

Hotter Cooler

Sequential color scheme: a sequential color scheme is a scale that represents a larger value at one end and a smaller value at the other.

We associate reds and oranges with hotter temperatures, and blues and purples with cooler temperatures, so this scale works well when we're representing the top 10 hottest months, that range from 1 (the hottest) to 10 (the coolest).

Diverging color scheme: a sequential color scheme that is used for data that diverge from an average or midpoint.

Color Discovery Tools

colorbrewer2.org
canva.com
design-seeds.com
color.adobe.com
coolors.co
colormind.io
colourlovers.com
mycolor.space

"Color selection in data visualization is not merely an aesthetic choice, it is a crucial tool to convey quantitative information."

— Robert Simmon, NASA

Build the Top Ten Hottest Months in code

Here are the prerequisites you need to complete this project:

1. You know how to create a basic web page using HTML and CSS, and you know how to use a plain text editor or IDE to write the code.

2. You have some programming experience with the JavaScript programming language.

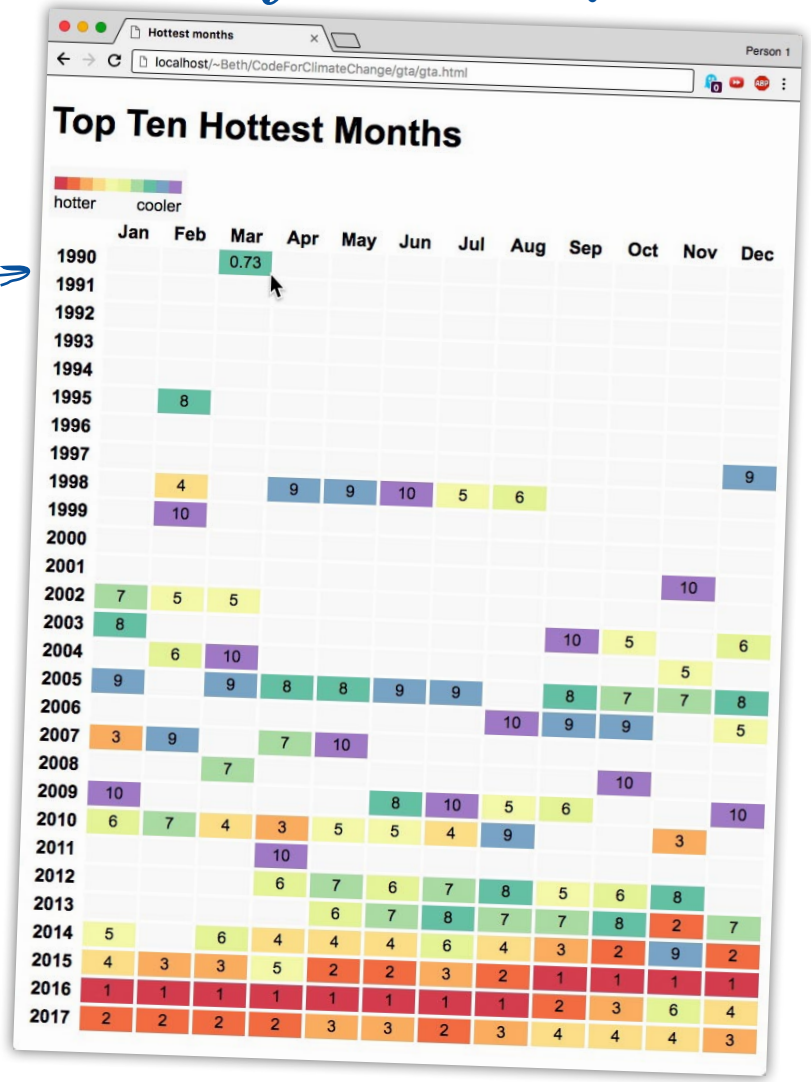
3. You have used a JavaScript library, like jQuery or Underscore, and are comfortable downloading and Inking to library code.

If you have all these prerequisites, then follow along with the videos in the online course, *How to Visualize Data with JavaScript*.

We've included all the steps to complete the visualization in this workbook, however, the online course videos include much more detail about the code and how it works.

Here's what the final visualization will look like in your browser... almost exactly the same as the coloring exercise you did a few pages back.

We'll add an interactive feature to this visualization so you can see the temperature anomaly value when you mouse over the month.



Step 1: Get the data into an array

Earlier, you got a look at the data we'll be using for this visualization when we analyzed it to understand what it meant. The data file we looked at had monthly data points for the years 1880-2017. For this coding exercise, we need only the data points for the top ten hottest months for those years.

We've processed that original data file and pulled out the top ten hottest Januarys, the top ten hottest Februarys, and so on, and saved that data into a new file.

We changed the format of the file a bit too, so that each data point is represented by a JavaScript object. All of the top 10 hottest months for each month is in an array, and the data for all the months is also in an array, so we have a two dimensional array. We're storing that array in the variable `data`, and we've put the array in a separate file, "gta_data.js".

We'll link to this file in the next step when we create the HTML.

If you want to, you can store and access this data in JSON format (see the file "gta.json"). We take you through how to load the data from this JSON file in the online course, in lesson *JavaScript to Load the Data*.

This is a two dimensional JavaScript array, containing a JavaScript object for each data point.

Now we have all the hottest Januarys grouped together, in order from hottest to coolest.

Each data point for a month is stored in an object with the year, month, and temperature anomaly value.

The ten data points for each month are stored in array...

...And those arrays are themselves stored in an array. This format makes it easy to read in the data and access the year, month, and value data points that we need for the visualization.

We'll read and process this data using JavaScript, and use it to make the visualization.

```
let data = [[
  { "year": 2016,  "month": 1,  "value": 1.06 },
  { "year": 2017,  "month": 1,  "value": 0.91 },
  { "year": 2007,  "month": 1,  "value": 0.89 },
  { "year": 2015,  "month": 1,  "value": 0.82 },
  { "year": 2014,  "month": 1,  "value": 0.7 },
  { "year": 2010,  "month": 1,  "value": 0.7 },
  { "year": 2002,  "month": 1,  "value": 0.7 },
  { "year": 2003,  "month": 1,  "value": 0.69 },
  { "year": 2005,  "month": 1,  "value": 0.62 },
  { "year": 2009,  "month": 1,  "value": 0.6 }
],[
  { "year": 2016,  "month": 2,  "value": 1.21 },
  { "year": 2017,  "month": 2,  "value": 0.97 },
  { "year": 2015,  "month": 2,  "value": 0.89 },
  { "year": 1998,  "month": 2,  "value": 0.86 },
  { "year": 2002,  "month": 2,  "value": 0.78 },
  { "year": 2004,  "month": 2,  "value": 0.72 },
  { "year": 2010,  "month": 2,  "value": 0.71 },
  { "year": 1995,  "month": 2,  "value": 0.69 },
  { "year": 2007,  "month": 2,  "value": 0.67 },
  { "year": 1999,  "month": 2,  "value": 0.67 }
],[ ...
[
  { "year": 2015,  "month": 12,  "value": 1.13 },
  { "year": 2014,  "month": 12,  "value": 0.84 },
  { "year": 2016,  "month": 12,  "value": 0.8 },
  { "year": 2006,  "month": 12,  "value": 0.77 },
  { "year": 2003,  "month": 12,  "value": 0.74 },
  { "year": 2013,  "month": 12,  "value": 0.71 },
  { "year": 2005,  "month": 12,  "value": 0.63 },
  { "year": 1997,  "month": 12,  "value": 0.63 },
  { "year": 2009,  "month": 12,  "value": 0.62 },
  { "year": 1999,  "month": 12,  "value": 0.58 }
]];
```

Download the data and all the rest of the code from: URL HERE. Make sure this data file is saved as gta.json in the same folder as where you'll put your HTML and CSS.

Step 2: Write the HTML

The next step is to write the HTML. You should already know that HTML is how we create and organize content in a web page. The HTML for this application is pretty simple; we'll be doing the heavy lifting in the JavaScript.

A couple of things to note about the HTML: in the `<head>`, we are linking to the jQuery library (which provides some nice features to create and manipulate HTML and CSS from JavaScript) as well as the JavaScript we'll be writing for the application. Also notice that we're using a list (`<ul>`) to represent the color legend at the top (go back a couple of pages to see what that looks like) and a `<table>` to store the year/month/temperature table of data. We'll be generating the rest of the color legend list and the data table from JavaScript. Once you've typed in the HTML, save the file as "gta.html".

```
<!doctype html>
<html>
<head>
  <title>Hottest months</title>
  <meta charset="utf-8">
  <script src="lib/jquery-3.3.1.min.js"></script>
  <script src="gta.js"></script>
  <script src="gta_data.js"></script>
  <style>...</style>
</head>
<body>
  <h1>Top Ten Hottest Months</h1>
  <div id="legend">
        <ul id="colorList"></ul>
        <div class="left">hotter</div> <div class="right">cooler</div>
  </div>
  <table id="dataTable">
    <tr><th></th><th>Jan</th><th>Feb</th><th>Mar</th><th>Apr</th><th>May</th>
    <th>Jun</th><th>Jul</th><th>Aug</th><th>Sep</th><th>Oct</th><th>Nov</th>
    <th>Dec</th></tr>
  </table>
</body>
</html>
```

Here's your standard HTML page stuff.

Make sure you've downloaded the jQuery library and put the file in the "lib" folder.

We're linking to jQuery and to our own JavaScript in gta.js. We'll put the data in an array in gta_data.js, and link to that too.

We'll get to the style later.

Here's the structure for the color legend. We'll build this out further with JavaScript and of course CSS.

And here's the table where we'll put all the data points. Right now, we've just got table headings for each month, and we'll build the rest with JavaScript.

Step 3: Write the JavaScript

We've got our data, we've got our HTML and our CSS, so now it's time to write the JavaScript to add the behavior to the page. This code does four things:

1. Calls createGraphic() with the data from the array in gta_data.js.
2. Builds the table of data points with years on the left column of the table, and a table cell for each value in the data.
3. Creates the color legend.
4. Adds the value of the temperature anomaly and the appropriate color for each data point to the table.

Remember, the table headings with the months are already in the HTML.

Here's the code. Remember, we're using jQuery so you'll see jQuery syntax mixed in with the plain JavaScript.

After the browser loads the page (along with the CSS and code), this function is called. The function calls createGraphic, passing in the data (which is an array– take another look at the data a couple of pages back if you need a reminder).

```
window.onload = function() {
  createGraphic(data);
}
function createGraphic(data) {
  let colors = // hottest to coolest
               ["#D53E4F", "#F46D43", "#FDAE61", "#FEE08B", "#F6FAAA",
                "#E6F598", "#ABDDA4", "#66C2A5", "#7BA5C7", "#A17BC7"],
  $dataTable = $("table#dataTable");
  // build the table
  for (let i = 0; i < 28; i++) {
    let cell = "<th>" + (1990+i) + "</th>";
    for (let j = 1; j <= 12; j++) {
      cell = cell + "<td id=\"" + (1990+i) + "_" + j + "\"> </td>";
    }
    let row = "<tr>" + cell + "</tr>";
    $dataTable.append(row);
  }
```

An array of colors we'll use to make the color legend.

Here's where we're building the table. We're showing 28 years worth of data (1990–2017) and each year has 12 months.

We build up a whole row of cells (<td> elements).

We're assigning a unique id to each <td> element so we can easily access it later when we add value and color to the data points. Each id will look like year_month, where year starts at 1990 and goes to 2017.

Here's where we add the row to the table.

Step 3, continued: Write the JavaScript

```
// make the legend
for (let k = 0; k < colors.length; k++) {
  $("ul#colorList").append("<li style=\"background-color: " + colors[k] + "\"> </li>");
}
// 0 is 1990, 27 is 2017
// i is index = color to choose
data.forEach(function(monthData) {
  monthData.forEach(function(yearData, i) {
    let id = yearData.year + "_" + yearData.month;

    let color = colors[i];
    $td = $("table#dataTable td#" + id);
    $td.text(i + 1);
    $td.on("mouseover", function(e) {
      $(e.target).text(yearData.value);
    });
    $td.on("mouseout", function(e) {
      $(e.target).text(i+1);
    });
    $td.css("background-color", color);
  });
});
}
```

For each color in the color array (previous page), we add a list item (<li>) with a color from the array.

Now it's time to add the values from the data to the table. We loop over the main array in the data, getting each item...

Remember that each item is an array of objects for a given month.

So, we loop through the objects for that month, and grab the year and month values from each object and use that to make the id we'll use to get the right <td> element from the table. The id will look like 1990_3 (for March, 1990).

This part's a bit tricky; we know the data for each month is organized from hottest to coolest, and our colors array is also from hottest to coolest color, so we can use the index of the data point in the array (i) as the index into the color array to get the appropriate color for this table cell.

We get the table cell from the table using its id, and then add the top 10 ranking as the text for the element. The ranking is i+1 (corresponding to the position in the array)

Now we can grab the <td> element from the table with the id we constructed above. Once we have that element, we add a mouseover handler to show the temperature anomaly value for that month, and a mouseout handler to hide that value again.

And finally, we add the appropriate color to the <td> element.

Whew! That's it.

Step 4: Create the style with CSS

If you load the HTML page into the browser now, you'll notice we still need to do some work to make the visualization look good. We've created the structure of the page HTML, and we've added the data and the interactive behavior with JavaScript, so now we'll use CSS to make the page look a certain way.

We won't go into too much detail on the CSS here (but we do in the video *Add the Color Legend*).

Add this CSS in the "gta.html" file in between the `<style>` tags.

And don't forget you can download all the code.

```
body {
    font-family: Arial, Helvetica, sans-serif;
}
div#legend {
    position: relative;
    background-color: rgb(250, 250, 250);
    display: inline-block;
    padding: 5px 5px 20px 5px;
    font-size: .8em;
}
div#legend div.left {
    position: absolute;
    bottom: 5px; left: 5px;
}
div#legend div.right {
    position: absolute;
    bottom: 5px; right: 5px;
}
ul#colorList {
    padding: 0px;
    margin: 5px 0px;
}
ul#colorList li {
    list-style-type: none;
    display: inline-block;
    width: 14px;
    height: 14px;
}
```

Your basic font info for the page.

These next five rules are for constructing the color legend, which, remember, looks like this:

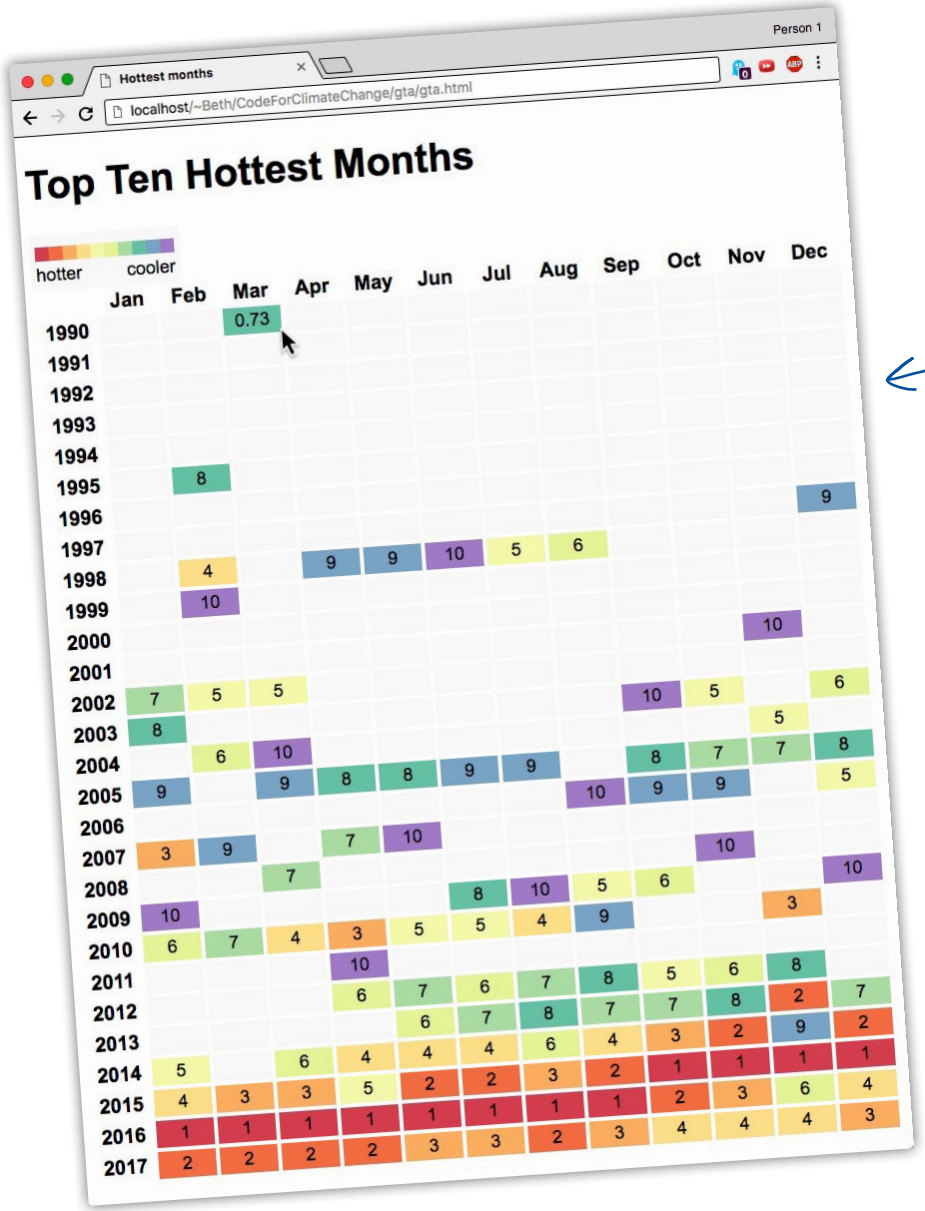
We're absolutely positioning the Hotter and Cooler labels underneath the color legend. Look back at the HTML to see where we defined the classes for these.

And here's where we style the list of colors.

We're using a <li> element for each color in the legend. By setting the display to inline-block we make sure the colors are right next to each other, and then we can set a width and height for each <li> so it's a square. We'll be creating these <li> elements and assigning colors to them in code.

```
table#dataTable {
    width: 100%;
    table-layout: fixed;
}
table#dataTable td {
    background-color: rgb(250, 250, 250);
    padding: 1px;
    border: 1px solid white;
    font-size: .5em;
    text-align: center;
    height: 25px;
}
```

These last two rules are for the table where we'll be putting the data. Initially each table cell (<td> element) will be gray, but we'll be filling in the ones that represent the top ten hottest months with colors in code.



Step 5: Test your code

To run this code, save your HTML in "gta.html", your CSS in "gta.css", and your JavaScript in "gta.js". Make sure you've downloaded the "gta_data.js" file, and got all your files are in the right places and and then load the HTML file "gta.html" into your browser. If you've done everything right you should see...

Congratulations! You just built your first climate data visualization in code! Try mousing over the data points in the table to see the temperature anomaly for those months. Compare the values from back in the '90s and early 2000s to the values in the teens, and notice how clearly you can see the increase in temperature anomalies just by seeing how the colors are changing over time.

The climate data we downloaded from NOAA is data from the entire globe, and over a long period of time—138 years. We took that data, analyzed it, and visualized the top ten hottest months since 1880, so we can see how the climate is changing over time (remember, *climate* is a pattern of long-term weather conditions, and *climate change* is the change in that pattern).

Climate is what you expect;
weather is what you get.

Climate Change is how what you expect is changing over time.

wickedlysmart

Austin • Annapolis • Seattle

wickedlysmart.com