11.2.4 Storyboarding

Dana has grown more familiar with JavaScript syntax and her basic code is gaining momentum. She's ready to start putting it to use!

Dana's goal is to create an interactive webpage that allows readers to parse the data around UFO sightings. So, she essentially needs to build two things: the webpage that will allow users to view the data (HTML) and a dynamic table that will present it (JavaScript).

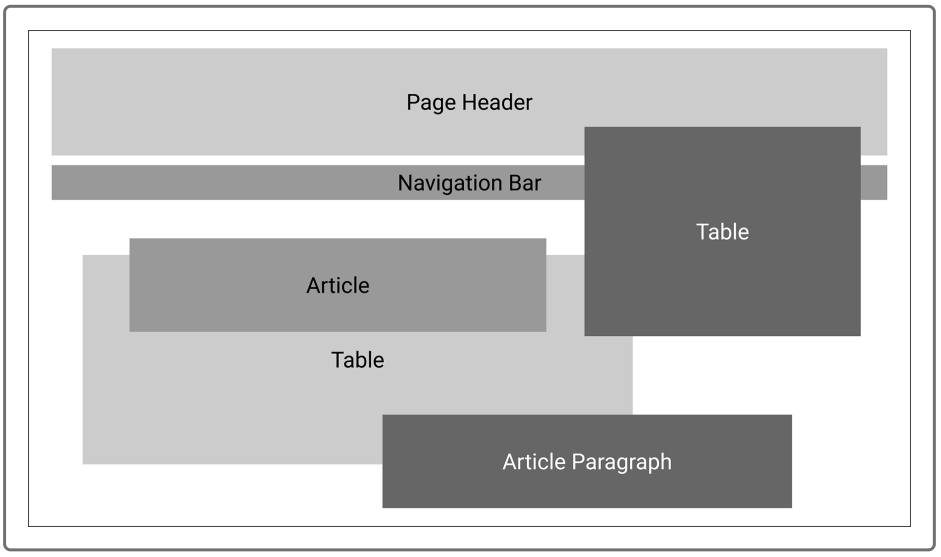
Dana wants to storyboard her website to have an idea of what her readers will see when they view the final product. Storyboarding is incredibly useful in determining the layout of a webpage, so it's important to complete this step early in order to save time later. It's like building a house. You need to know how it's all going to fit together before you start building!

Once the template has been created, Dana can begin to code the JavaScript portion by first importing the data and then referencing it with a variable.

Typically, developers build HTML and JavaScript elements somewhat simultaneously because they complement each other. For example, the JavaScript table will be referenced within the HTML code, and different HTML components will be referenced within the JavaScript code. Because these files are so closely linked, Dana will switch between building the JavaScript table (within the `app.js` file) and the HTML page (within an `index.html` file).

Dana also has a solid idea of how she wants her webpage to look, but it's easy to get lost in the details of building a webpage without a visual reference. A visual reference such as a storyboard will help Dana outline all of the elements she wants included, such as the article title, a

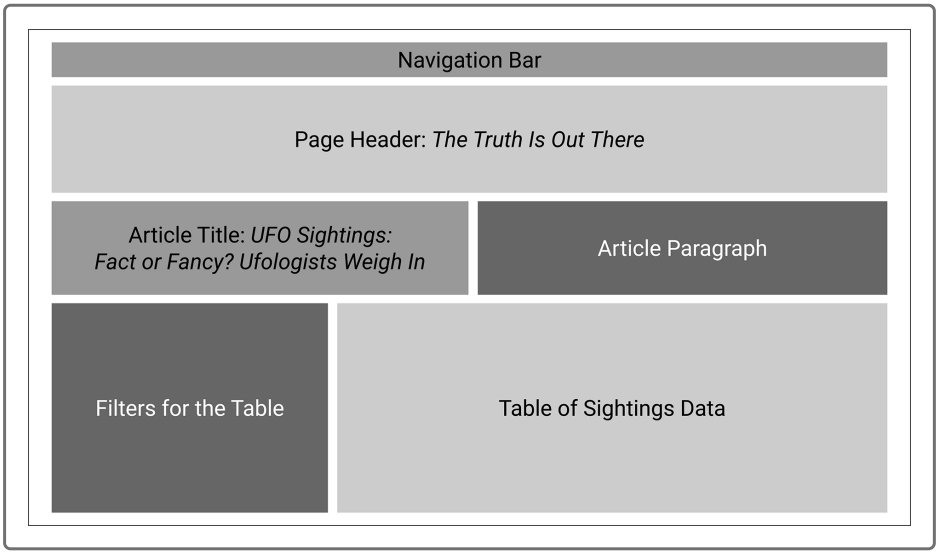
summary, and the table itself. Then, when she begins creating JavaScript code to include the table, she'll know exactly which HTML components she'll be connecting to her table. Dana already knows she'll have several individual components on the webpage, shown below:



Now she just needs to figure out how to assemble them. This is where a storyboard comes in.

Create a Storyboard

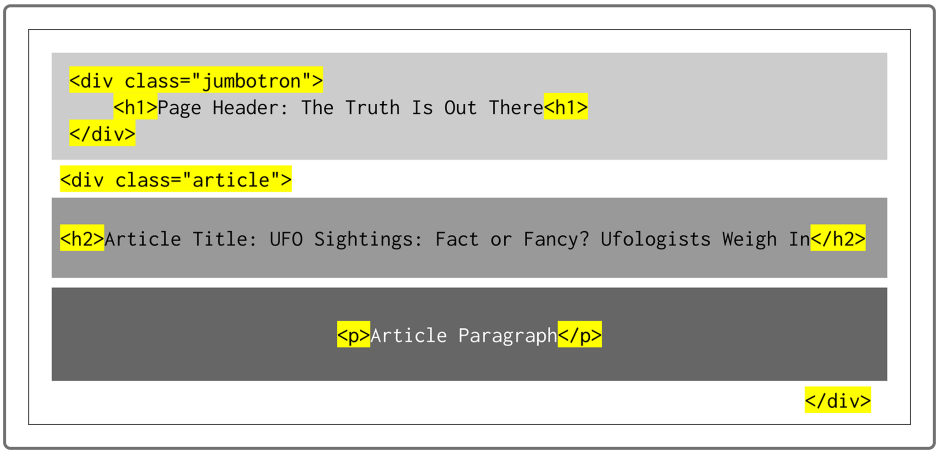
A **storyboard** serves as a kind of blueprint for your site and helps with the transition from idea to finished product. Think of it as a map of the webpage.



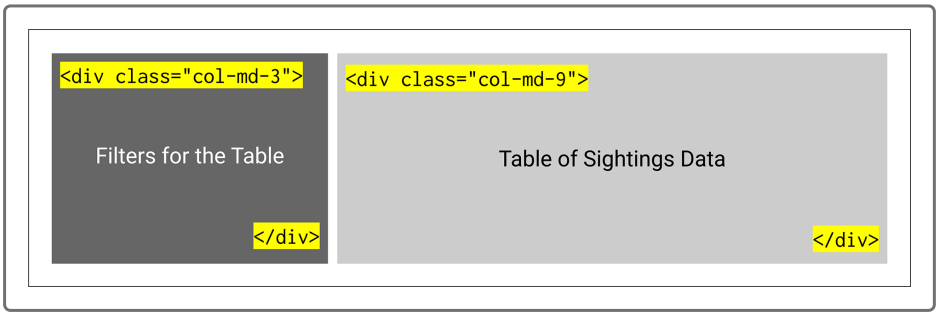
This step is key for a couple of reasons.

First, knowing how we want the webpage to look before building it will save us time later. Second, it helps us make sure we've captured everything we want displayed. Sometimes, seeing the map of the website helps us ensure that all the elements we want displayed are included.

We already know what components we want to use, such as a Jumbotron for the header, and the grid system for the filters and table. See the following image:



We also have an idea of how many columns we want each table component to use.



Now that a storyboard is in place, we can really get going! Let's align our code.

Align the Code

When we align our code, we're putting our plans into action, such as when we start transitioning our storyboard into a webpage. We'll start by

building our components. The first one will be the table we generate with JavaScript. Open your `app.js` file with VS Code. The first thing we're going to do is import the data. This won't look like an import from Python. For starters, the double backslash (//) is how you comment your code in JavaScript.

In your code editor, type the following to declare a variable, `tableData`, using `const`.

```
// import the data from data.js
const tableData = data;
```

Next, we need to point our data to our HTML page. Specifically, we need to tell JavaScript what type of element the data will be displayed in. We already know that the data will be displayed in a table, so in our code editor we'll reference the `tbody` HTML tag using D3.

IMPORTANT

> D3 is a JavaScript library that produces sophisticated and highly dynamic graphics in an HTML webpage. It is often used by data professionals to create dashboards, or a collection of visual data (such as graphs and maps), for presentation.

Return to your code editor and type the following:

```
// Reference the HTML table using d3
var tbody = d3.select("tbody");
```

With this code, we:

1. Declare a variable, `tbody`
2. Use `d3.select` to tell JavaScript to look for the `<tbody>` tags in the HTML

Although we aren't building the HTML right now—we'll do this after we put together the code—we already know that the data will fit into that tag because it's a standard table tag that is used often in HTML, with or without JavaScript enhancements.

© 2020 - 2022 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.