Lesson Thursday

Introduction to Programming (/introduction-to-programming) / Git, HTML and CSS (/introduction-to-programming/git-html-and-css) / Downloading and Installing Bootstrap

Text Cheat sheet

In this lesson, we'll walk through how to download and install Bootstrap into a project. We'll also discuss Bootstrap versions and how they impact how we download and find information about Bootstrap. There are multiple ways to download Bootstrap. We'll address the two simplest approaches here:

- Downloading the Bootstrap source code and then adding the files directly to a project.
- Using a content delivery network (CDN) to link to Bootstrap online.

In future courses, we'll explore additional ways to integrate Bootstrap and other frameworks into a project. In the very next lesson, we'll create an example project and learn how to use Bootstrap.

Versioning

You likely are already familiar with the concept of versioning from using computers and phones: new operating systems (software) are released regularly, and so are new versions of phones (hardware)! Well, many libraries, frameworks, and toolkits follow the same process of releasing software in versions to separate a distinct state of the software from later states. The higher the version number, the newer it is. One goal of versioning is to enable existing clients using an older software version to continue using it, even as newer versions are released.

The difference between version 1 (the earlier version) and version 2 (the later version) of a piece of software could be anything: the names of the tools, what tools are offered, how the tools should be used or configured, and more. Because of this, it's very important for us to pay attention to the version number of the software we're using so that we can use it in the right way, and find the right documentation online.

Websites choose to upgrade to newer versions of frameworks (or other tools) for a few reasons: when there are fixes for security issues, when there are performance improvements, to use new features, or when the tool has met its "end of life". When a version of software is at its **end of life**, it means no new features are being added to it and no developer is maintaining it, including making security fixes. It is usually possible to use software that's at its end of life, but it's not recommended.

While there's a lot of good reasons to update to the latest version number, making a version upgrade can be costly in terms of time and money. Often when the version number of one tool is upgraded, often other tools also need to be upgraded to a compatible version number. When there are **breaking changes** between one version and the latest version, that means that your code will break when you upgrade to the latest version unless you update your code to match the new standards. Because of all of these reasons, many websites don't upgrade to the latest version, and wait until there's a compelling reason to upgrade.

The same is true in our curriculum. We often don't use the latest version number of the technology we teach, just the same as most tech companies. As of late 2022, we're using version 5.2 of Bootstrap, but by the time you are reading this, Bootstrap may be at a later version. It's up to you if you want to try using a later version in your own projects. However, you should always be aware that there could be breaking changes (like completely different tools) between the version we use in the LearnHowToProgram.com curriculum and a later version Bootstrap.

Downloading Bootstrap

The first way we can add Bootstrap to a project is to download all of the necessary files and add them to our project. Then we'll link to them in our HTML just like we link to our CSS file.

Downloading Files from Bootstrap's Website

We are currently using Bootstrap version 5.2, which can be downloaded from the Bootstrap 5.2 download page (https://getbootstrap.com/docs/5.2/getting-started/download/). Scroll down to the section titled "Compiled CSS and JS" and hit the download button.

This will download a `.zip` file. Unzip the file using your computers extractor tool and then look inside of the folder. You should see something similar to this:

```
bootstrap/
├── css/
│   ├── bootstrap-grid.css
│   ├── bootstrap-grid.css.map
│   ├── bootstrap-grid.min.css
│   ├── bootstrap-grid.min.css.map
│   ├── bootstrap-grid.rtl.css
│   ├── bootstrap-grid.rtl.css.map
│   ├── bootstrap-grid.rtl.min.css
│   ├── bootstrap-grid.rtl.min.css.map
│   ├── bootstrap-reboot.css
│   ├── bootstrap-reboot.css.map
│   ├── bootstrap-reboot.min.css
│   ├── bootstrap-reboot.min.css.map
│   ├── bootstrap-reboot.rtl.css
│   ├── bootstrap-reboot.rtl.css.map
│   ├── bootstrap-reboot.rtl.min.css
│   ├── bootstrap-reboot.rtl.min.css.map
│   ├── bootstrap-utilities.css
│   ├── bootstrap-utilities.css.map
│   ├── bootstrap-utilities.min.css
│   ├── bootstrap-utilities.min.css.map
│   ├── bootstrap-utilities.rtl.css
│   ├── bootstrap-utilities.rtl.css.map
│   ├── bootstrap-utilities.rtl.min.css
│   ├── bootstrap-utilities.rtl.min.css.map
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   ├── bootstrap.rtl.css
│   ├── bootstrap.rtl.css.map
│   ├── bootstrap.rtl.min.css
│   └── bootstrap.rtl.min.css.map
└── js/
    ├── bootstrap.bundle.js
    ├── bootstrap.bundle.js.map
    ├── bootstrap.bundle.min.js
    ├── bootstrap.bundle.min.js.map
    ├── bootstrap.esm.js
    ├── bootstrap.esm.js.map
    ├── bootstrap.esm.min.js
    ├── bootstrap.esm.min.js.map
    ├── bootstrap.js
    ├── bootstrap.js.map
    ├── bootstrap.min.js
    └── bootstrap.min.js.map
```

There are two directories: `css` and `js`.

- The `css` folder contains files related to Bootstrap CSS. We can use either `bootstrap.css` or `bootstrap.min.css` in our projects to access all of Bootstrap's CSS. Here's the difference between the two files:
    - `bootstrap.css` is the larger version of the source code. If we open this file, we'll be able to read all the CSS rules for Bootstrap, which can be very helpful when we are just learning about Bootstrap — and when we want to figure out how to override a Bootstrap rule.
    - `bootstrap.min.css` is a **minified** file, which means that the source code has been compressed so that it's more efficient for machines to read. However, it's not in a human-readable form anymore. This is great for production (code that's deployed online) because it's faster, but it's not so great for development (code that's in the process of being built — which is exactly what we are doing). We recommend using `bootstrap.css` for learning purposes.
- The other files in the `css` directory, like `bootstrap-grid.css`, contains CSS for only a specific Bootstrap feature. This allows developers who only want to use Bootstrap's Grid functionality to include a smaller Bootstrap resource in their projects. We'll stick to including all of Bootstrap in our projects.

- The `js` directory contains JavaScript code necessary for animations, pop-ups, alerts, and other visual effects, as described here (https://getbootstrap.com/docs/5.2/getting-started/javascript/). We won't learn how to use JavaScript until the next course section, however, we will learn how to load

Bootstrap's JavaScript (JS) into our HTML so that we can use Bootstrap components that use JS. Keep reading to learn how to do this.

Adding a Downloaded Copy of Bootstrap's CSS to a Project

To integrate Bootstrap into our project, we simply need to move `bootstrap.css` into the `css` directory of the project we're working on. This would be a good time to practice the `$ mv` command in the terminal, though it's also fine to drag and drop the file.

Once the Bootstrap stylesheet is located in our `css` folder, we need to link to it in our site's `<head>` tags, just as we do with our own stylesheets:

```
<!DOCTYPE html>
<head>
  <link href="css/bootstrap.css" rel ="stylesheet" type="text/css">
  <link href="css/styles.css" rel="stylesheet" type="text/css">
  <title>My Project</title>
</head>
...
```

Note: When you see ellipses (`...`) in a code snippet, it means portions of the file were omitted from the example. This keeps examples brief, making it easier to identify the **new** code that has been added.

As we see here, we can link to multiple stylesheets. However, it's very important that *our* stylesheet is listed **after** the Bootstrap stylesheet. This way, the rules in our stylesheet will override rules that have the same name in the Bootstrap stylesheet. We'll cover this further in a future lesson.

OPTIONAL: Adding a Downloaded Copy of Bootstrap's JavaScript to a Project

We don't need to know JavaScript in order to configure our projects to use Bootstrap's pre-built JavaScript functionality. This will open the door for us to use Bootstrap tools that contain animations, pop-ups, alerts, and other visual effects. An example of a Bootstrap component that uses JavaScript is a dropdown (https://getbootstrap.com/docs/5.2/components/dropdowns/). To see all Bootstrap components that use JavaScript, check out this list (https://getbootstrap.com/docs/5.2/getting-started/introduction/#js-components).

To start, create a `js/` subdirectory in your project. Then move the downloaded `bootstrap.bundle.js` file into the `js/` subdirectory. We'll use `bootstrap.bundle.js` instead of `bootstrap.js`, because the bundled file contains all of the JavaScript tools (including the JavaScript library called "Popper") to use any Bootstrap component that uses JavaScript.

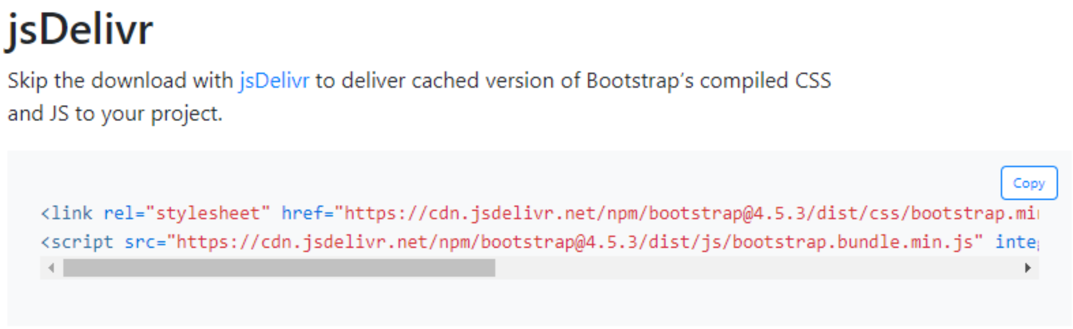
Next, we'll load the `bootstrap.bundle.js` into our HTML by using the HTML `<script>` tag. `<script>` is short for JavaScript, and it is used to embed JavaScript into our HTML file. .**The script tag must be placed at the bottom of our HTML `<body>` tag** so that it is loaded last:

```
<!DOCTYPE html>
<head>
  <link href="css/bootstrap.css" rel ="stylesheet" type="te
xt/css">
  <link href="css/styles.css" rel="stylesheet" type="text/c
ss">
  <title>My Project</title>
</head>
<body>
  ...
  <script src="js/bootstrap.bundle.min.js"></script>
</body>
...
```

Similar to `<img>` tags, the `<script>` tag has a `src` attribute that tells our program where to find the JavaScript file that we want to load into our HTML.

Linking to a CDN

We can also add Bootstrap into a project via a **content delivery network**, or **CDN**. This is the option under the section titled "CDN via jsDelivr" on Bootstrap's download page (https://getbootstrap.com/docs/5.2/getting-started/download/#cdn-via-jsdelivr). jsDelivr is a content delivery network.



jsDelivr

Skip the download with jsDelivr to deliver cached version of Bootstrap's compiled CSS and JS to your project.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/css/bootstrap.mi
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js" inte
```

A content delivery network is exactly what it sounds like — a network of servers that make content available for users. In this case, Bootstrap is making its stylesheets available online, which means we can just load the online version directly to our project without needing to add any files to our `css` directory.

We can copy/paste the `<link>` directly into the `<head>` tags of our sites. This is similar to the manner we've been linking our stylesheets so far. The only difference is that the `href` property refers to an online location instead of a location in our `css` directory:

```
<!DOCTYPE html>
<head>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65" crossorigin="anonymous">
  <link href="css/styles.css" rel="stylesheet" type="text/css">
  <title>My Project</title>
</head>
...
```

What are the `integrity` and `crossorigin` Attributes?

The `integrity` and `crossorigin` attributes are used for Subresource Integrity (SRI) (https://www.w3.org/TR/SRI/) checking. According to srihash.org (https://www.srihash.org/), SRI checking allows browsers to ensure that resources hosted on third-party servers (like Bootstrap's CSS library that we are accessing via CDN) have not been tampered with. Performing SRI checking is a best practice set by W3C, the World Wide Web Consortium (https://www.w3.org/Consortium/), an international organization that sets standards for the web.

Note, you do not need to retain any of the above information. However, it is interesting and you are welcome to read more on your own!

OPTIONAL: Using a CDN to Load Bootstrap's JavaScript in a Project

If we want to load Bootstrap's JavaScript into our project from a CDN, we simply need to add the CDN `<script>` tag from the Bootstrap download page to the bottom of our HTML's `<body>` tag:

```
<!DOCTYPE html>
<head>
  <link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
    rel="stylesheet"
    integrity="sha384-rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2Kadq2F9CUG65"
    crossorigin="anonymous">
    <link href="css/styles.css" rel="stylesheet" type="text/css">
  <title>My Project</title>
</head>
<body>
  ...
  <script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-kenU1KFdBIe4zVF0s0G1M5b4hcpxyD9F7jL+jjXkk+Q2h455rYXK/7HAuoJl+0I4"
    crossorigin="anonymous">
  </script>
</body>
...
```

Notice the new formatting: we've updated both the Bootstrap `<link>` and `<script>` tags to list each attribute on a new line. Doing this is optional, but greatly improves the readability of longer tags.

It's worth mentioning again that we can review all of the Bootstrap components that use JavaScript by checking out this list on the Bootstrap docs (https://getbootstrap.com/docs/5.2/getting-started/introduction/#js-components)!

Final Thoughts

Ultimately, both approaches of loading Bootstrap into our projects work great. CDNs are a great way to go because they are very fast and reliable and don't require the extra step of downloading the files. In the example projects in LearnHowToProgram.com, we'll primarily use CDNs to link to Bootstrap.

The downside to installing Bootstrap with a CDN link is that you'll be unable to see Bootstrap styles in your project without an internet connection, because the project needs to load the online stylesheet to use Bootstrap's CSS.

Now that we know how to add Bootstrap to our own projects, we're ready to create an example project and learn how to use Bootstrap styles.

Previous (/introduction-to-programming/git-html-and-css/introduction-to-bootstrap-and-front-end-frameworks)
Next (/introduction-to-programming/git-html-and-css/bootstrap-basic-elements)

Lesson 59 of 64
Last updated March 24, 2023

disable dark mode

Epicodus (http://www.epicodus.com)

© 2023 Epicodus (http://www.epicodus.com/), Inc.