

INLS 672

Web Development 2

JavaScript Application Programming Interfaces (APIs)



UNC
SCHOOL OF INFORMATION
AND LIBRARY SCIENCE

Joan Boone
jpboone@email.unc.edu

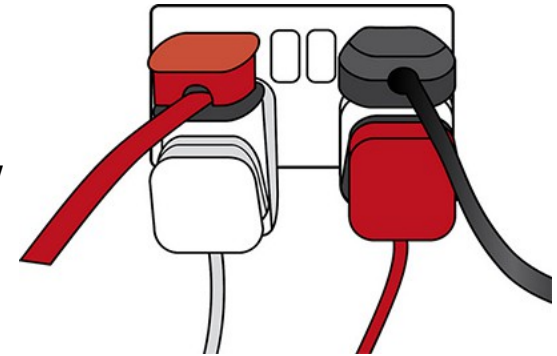
Part 1: API Overview

Part 2: Fetch API Example

What are APIs ?

Application Programming Interfaces are collections of code and tools that are available in all programming languages, and that allow developers to create complex functionality more easily.

- Provide building blocks of code that simplify the interface to complex functions by abstracting away the underlying details
- Provide common, frequently needed functions that allow developers to reuse existing code



JavaScript (client-side) has many APIs that fall into 2 categories

- Browser APIs that are built into your browser, e.g., DOM API
- Third party APIs – not built into your browser, and you need to access the code and documentation from the vendor's website.

What can APIs do?

Just about anything you need them to do: MDN [Web APIs](#)

Most common categories of browser APIs include

- DOM APIs for manipulating HTML and CSS, handling events
- Fetching data from the server using Fetch API (or XMLHttpRequest), and returning data in a standard format such as JSON or XML
- Drawing and manipulating graphics (Canvas, WebGL)
- Video and Audio APIs for customizing user controls
- Device APIs to access features such as GPS, camera, orientation, vibration
- Client-side storage APIs that enable offline app use, and saving state of apps between page loads.

Many third-party APIs: Twitter, Facebook, YouTube, Google, ...

Using DOM APIs

The theme-changer example uses the DOM API in several ways:

- Adds an event listener to the lighthouse image element
- Changes HTML content by toggling the lighthouse image
- Changes CSS styles by toggling the color properties of the custom variables between light and dark



Cape Hatteras Lighthouse
The Cape Hatteras Lighthouse is located on Hatteras Island in the Outer Banks in the town of Hatteras, North Carolina and is part of the Cape Hatteras National Seashore.
The Cape Hatteras Lighthouse is the tallest lighthouse in the United States. From ground to peak, the lighthouse is 202.5 feet tall, with 146 steps from the entrance to the lantern room. (Source: www.nps.gov/hahat)



Cape Hatteras Lighthouse
The Cape Hatteras Lighthouse is located on Hatteras Island in the Outer Banks in the town of Hatteras, North Carolina and is part of the Cape Hatteras National Seashore.
The Cape Hatteras Lighthouse is the tallest lighthouse in the United States. From ground to peak, the lighthouse is 202.5 feet tall, with 146 steps from the entrance to the lantern room. (Source: www.nps.gov/hahat)

theme-change.html

```
// When an image is clicked, toggle theme
let lighthouse = document.querySelector('img');
let body = document.querySelector('body');
lighthouse.addEventListener('click', changeTheme);
```

```
function changeTheme() {
  // If the 'light' image is currently displayed, change theme to 'dark'
  if (lighthouse.src.match("../HLH-Daytime.jpg")) {
    lighthouse.src = "../HLH-Nighttime.jpg";
    document.documentElement.style.setProperty('--bg-color', 'darkslateblue');
    document.documentElement.style.setProperty('--text-color', 'white'); }
  else { // Change 'dark' to 'light' theme
    lighthouse.src = "../HLH-Daytime.jpg";
    document.documentElement.style.setProperty('--bg-color', 'white');
    document.documentElement.style.setProperty('--text-color', '#5dade2'); }
}
```



Video courtesy of [Big Buck Bunny](#).

Using Video API to Customize User Interface Controls

```
var myVideo = document.querySelector("#video1");

var playPauseButton = document.querySelector("#playpause");
playPauseButton.addEventListener('click', playPause);

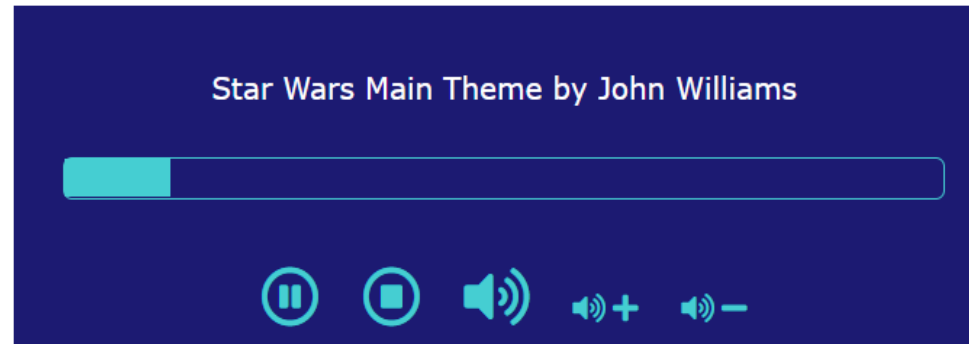
var stopButton = document.querySelector("#stop");
stopButton.addEventListener('click', stop);

function playPause() {
    if (myVideo.paused) {
        myVideo.play();
        playPauseButton.className = "far fa-pause-circle fa-2x";
    }
    else {
        myVideo.pause();
        playPauseButton.className = "far fa-play-circle fa-2x";
    }
}

function stop() {
    myVideo.pause();
    myVideo.currentTime = 0;
    playPauseButton.className = "far fa-play-circle fa-2x";
}
```

JavaScript

Using Audio API to Customize User Interface Controls



```
var myAudio = document.querySelector("#audio1");
. . .
var muteButton = document.querySelector("#mute");
muteButton.addEventListener('click', muteOnOff);
var volumeUpButton = document.querySelector("#volume-up");
volumeUpButton.addEventListener('click', volumeUp);
var volumeDownButton = document.querySelector("#volume-down");
volumeDownButton.addEventListener('click', volumeDown);
. . .
function muteOnOff() {
    if (myAudio.muted) {
        myAudio.muted=false;
        muteButton.className = "fas fa-volume-up fa-2x";    }
    else {
        myAudio.muted=true;
        muteButton.className = "fas fa-volume-off fa-2x";    }    }

function volumeUp() {
    if (myAudio.volume < 1.0)
        myAudio.volume = myAudio.volume + 0.1;    }

function volumeDown() {
    if (myAudio.volume > 0.0)
        myAudio.volume = myAudio.volume - 0.1;    }
```

audio-star-wars-theme-custom-ui.html

Fetch API

What is it?

A JavaScript interface that fetches data from a server and returns it to the browser in a standard format such as JSON or XML.

Why use it?

- Allows websites and applications to retrieve individual data items to update sections of a webpage *without having to re-load an entire page*. This technique greatly improves performance and website behavior.
- Requests are *asynchronous* and also improve performance by allowing the browser to keep working while waiting for a response. Synchronous requests stop the execution of any code until a response is received.
- *Practical application*: there is much open data available for applications to extract information and insights. The Fetch API simplifies access to this data so that it can be programmatically analyzed, and made more understandable with web interfaces. Some examples include:
 - Re-purpose with HTML+CSS to present data in a usable and actionable way.
 - Use 3rd party APIs and libraries to visualize data

Fetching data...a few comments

And some sources of confusion :)

A basic problem solved by Fetch API is that when you make a network request, it may take awhile for the results to be returned, so your code must 'pause' until the results are successfully returned before proceeding.

- The Fetch API is the most recent approach for retrieving data from a server asynchronously. It is a simpler interface than XMLHttpRequest (XHR), but XHR is still widely used, so you will see many examples that use it (and you can use it if it makes more sense to you).
- There are multiple ways to use the Fetch API
 - Frequently examples and tutorials will use the 'promises' approach which is a JavaScript feature for performing asynchronous operations. The `fetch()` returns a promise, which resolves to the response sent back from the server — then you use a `then()` function to run some follow-up code after the promise resolves. Multiple `then()` functions can be chained to handle various conditions. As you might guess, this begins to get verbose and complicated.
 - ▶ ▪ The `async/await` approach (to me) is simpler and easier to understand, and it is the one I use in the examples. You can use the promises approach, but if you are new to fetching data, I would recommend `async/await`.

Part 1: API Overview

Part 2: Fetch API Example

Using Fetch API to re-purpose JSON as HTML+CSS

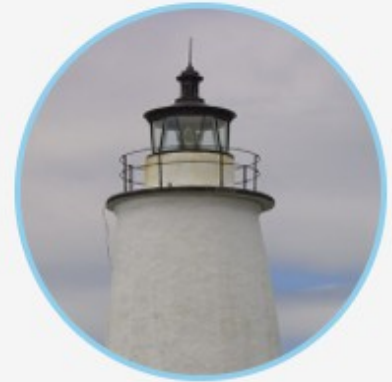
2 steps:

- 1) Fetch the data
- 2) Render the page



Cape Hatteras Lighthouse
Buxton, NC

The Cape Hatteras Lighthouse is located on Hatteras Island in the Outer Banks in the town of Buxton, North Carolina, and is part of the Cape Hatteras National Seashore.



Ocracoke Lighthouse
Ocracoke Island, NC

Ocracoke Light is the oldest operating light station in North Carolina and the second oldest lighthouse still standing in the state. The lighthouse was automated in 1955.

```
[{  
  "name": "Cape Hatteras Lighthouse",  
  "location": "Buxton, NC",  
  "image": "https://assets.codepen.io/108450/hatteras.jpg",  
  "description": "The Cape Hatteras Lighthouse is located on Hatteras Island ...."  
},  
{  
  "name": "Ocracoke Lighthouse",  
  "location": "Ocracoke Island, NC",  
  "image": "https://assets.codepen.io/108450/ocracoke.jpg",  
  "description": "Ocracoke Light is the oldest operating light station in ..."  
}]
```

lighthouses.json

Using Fetch API to re-purpose JSON as HTML+CSS

lighthouse.html



Cape Hatteras Lighthouse

Buxton, NC

The Cape Hatteras Lighthouse is located on Hatteras Island in the Outer Banks in the town of Buxton, North Carolina, and is part of the Cape Hatteras National Seashore.



Ocracoke Lighthouse

Ocracoke Island, NC

Ocracoke Light is the oldest operating light station in North Carolina and the second oldest lighthouse still standing in the state. The lighthouse was automated in 1955.

```
<!DOCTYPE html>
<html lang="en">
<head>
  ...
  <title>Lighthouses Fetch API Demo</title>
  <link rel="stylesheet" href="style.css">
</head>

<body>
  <div class="container">

    </div>
    <script src="lighthouses.js"></script>
</body>

</html>
```

When the browser loads the page, it creates the DOM, and then instructs the JavaScript interpreter to run the script.



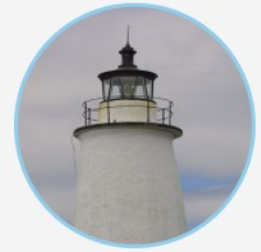
Using Fetch API to re-purpose JSON as HTML+CSS

lighthouses.js



Cape Hatteras Lighthouse
Buxton, NC

The Cape Hatteras Lighthouse is located on Hatteras Island in the Outer Banks in the town of Buxton, North Carolina, and is part of the Cape Hatteras National Seashore.



Ocracoke Lighthouse
Ocracoke Island, NC

Ocracoke Light is the oldest operating light station in North Carolina and the second oldest lighthouse still standing in the state. The lighthouse was automated in 1955.

```
async function getLighthouses() {
  let url = 'https://assets.codepen.io/108450/lighthouses.json';
  try {
    let response = await fetch(url);
    return await response.json();
  } catch (error) {
    console.log(error);
  }
}

// After getting the lighthouse data (a JSON object containing a list with 2 items).
async function renderLighthouses() {
  let lighthouses = await getLighthouses();
  let html = '';

  . . .


  let container = document.querySelector('.container');
  container.innerHTML = html;
}

renderLighthouses();
```

Step 1: Fetch the data

- Define an **async** function to get the lighthouse data. **async** functions know that there is a possibility that its code will need to wait for a response to be returned by the server – it won't happen immediately!
- The **await** keyword causes the JavaScript runtime to pause your code on this line, not allowing further code to execute in the meantime until the function call has returned its result — very useful if subsequent code relies on that result!
- **try/catch** clause because things can go wrong :(

```
async function getLighthouses() {  
  let url = 'https://assets.codepen.io/108450/lighthouses.json';  
  try {  
    let response = await fetch(url); // fetch the data  
    return await response.json();    // return as JSON  
  } catch (error) {  
    console.log(error);  
  }  
}
```



Once the **response** is available, create a **json** object from it, and return to the calling function

Step 2: Render the page

- Gets the lighthouse information that has been converted to a JSON object – a list with 2 items
- Loop through the list, extract the data, and insert into HTML template string.

```
[{
  "name": "Cape Hatteras Lighthouse",
  "location": "Buxton, NC",
  "image": "...hatteras.jpg",
  "description": "The Cape Hatteras ..."
},
{
  "name": "Ocracoke Lighthouse",
  "location": "Ocracoke Island, NC",
  "image": "...ocracoke.jpg",
  "description": "Ocracoke Light is ..."
}
]
```

`lighthouses.json`

```
async function renderLighthouses() {
  let lighthouses = await getLighthouses();
  let html = '';
  lighthouses.forEach(light => {
    let htmlSegment = `<div class="lighthouse">
      
      <h2>${light.name}</h2>
      <h3>${light.location}</h3>
      <div>${light.description}</div>
    </div>`;
    html += htmlSegment;
  });
  let container = document.querySelector('.container');
  container.innerHTML = html;
}
```

Loop through the list and for each item:

- Assign the current list item to the `light` variable
- Extract the data from the `light` object and substitute into the HTML template.

JavaScript template strings are delimited by the backtick (`). They contain placeholder variables that are replaced with values from the `light` object

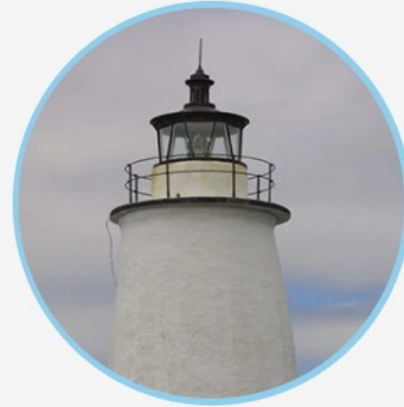
Practice: Add a website link



Cape Hatteras Lighthouse

Buxton, NC

The Cape Hatteras Lighthouse is located on Hatteras Island in the Outer Banks in the town of Buxton, North Carolina, and is part of the Cape Hatteras National Seashore.



Ocracoke Lighthouse

Ocracoke Island, NC

Ocracoke Light is the oldest operating light station in North Carolina and the second oldest lighthouse still standing in the state. The lighthouse was automated in 1955.

- `lighthouses.json` at the CodePen URL also contains a website for each lighthouse.
- Add this value, `{light.website}` to the HTML template so that the lighthouse name links to the website as illustrated in the screenshot.