ALLSTARCODE

# Code Along: More DOM

**Starting scenario:**

Fork and clone this repo moreDomCodeAlong and have students do the same so they can follow along.

**Agenda:**

1. Visibility: hidden;
2. Display: none;
3. Remove element
4. Add element
5. Change element

**Guide:**

1. Visibility: hidden;
    a. View the site in the browser.
    b. Explain we'll first cover different ways to remove elements, then we'll go over using the DOM.
    c. Select one of the images and change it's "visibility" attribute to "hidden"
    d. Ask: What observations can you make about how "visibility:hidden" works? Elicit or provide: It hides the element, but the element still takes up space and affects the positioning of other elements.
    e. Revert the change.
2. Display: none;
    a. Select one of the images and change it's "display" attribute to "none"
    b. Ask: What observations can you make about how "display:none" works? Elicit or provide: The element no longer takes up space and does not affect the positioning of other elements.
    c. Ask: Does the image still exist in the document? Elicit or provide: Yes.
        i. *NOTE: you can inspect the HTML using Dev Tools and still see the element exists.*
    d. Revert the change.
3. Remove element
    a. In the "main.js" file, have students dictate selecting the first image using its id - "first" - and storing it in a variable named "img"
    b. Explain we can also remove elements from the DOM completely using .remove(). Add the code and refresh the page.

i.
```
const img = document.getElementById("first");
img.remove();
```

c. Refresh the page.
d. Ask: Does the image still exist in the document? Elicit: No.
e. Explain that although the node was removed it still exists in memory, just not attached to the DOM!
    i. In the console, check on the “img” variable.
    ii. Explain we could modify the node or append it elsewhere later if we wanted.

4. Add element
    a. Explain we can create a new element with “*document.createElement()*”
    b. Add the code to the “main.js” file to create a new “img” element and store it in a variable called “newImg”.
    i.
```
const img = document.getElementById("first");
img.remove();

const newImg = document.createElement("img");
```

    c. Refresh the page.
    d. Ask: Where’s the new image node we created? Elicit or provide: It only exists in memory.
    e. Show students the “newImg” variable in the console.
    f. Explain it only exists in memory and we’ll need to attach it to the DOM in order to see it
    g. Explain we can attach one node to another, so let’s grab the “.main-content” div using .querySelector() and store it in a variable called “mainContainer”
    i.
```
const img = document.getElementById("first");
img.remove();

const newImg = document.createElement("img");
const mainContainer = document.querySelector(".main-container");
```

    h. Ask: How can we attach a node onto another? Elicit or provide: by using .appendChild()
    i. Add the code to append “newImg” to the .main-container
    i.
```
const img = document.getElementById("first");
img.remove();

const newImg = document.createElement("img");
const mainContainer = document.querySelector(".main-container");

mainContainer.appendChild(newImg);
```

    j. Refresh the page.
    k. Ask: Is our new image node in the document? Elicit: Yes, but there’s not actual image!

5. Change element
    a. Explain we can alter the images .src by setting the path to the image as a string. In this case we'll use the "cry.png" image in the "assets" folder
        i.
```
const img = document.getElementById("first");
img.remove();

const newImg = document.createElement("img");
const mainContainer = document.querySelector(".main-container");

mainContainer.appendChild(newImg);
newImg.src = "assets/cry.png";
```
            1. *NOTE: the pathing reference is from the document's perspective and not the .js file.*