

Hello, World DOMination: Inserting Elements in JS File and Script Block

Let's practice adding new elements to our page. We'll create a second element, a `div` with an id of `lickable-frog`, and append it to the `body` like we did the first time. Update the Javascript function to append a second element to the page.

Javascript

```
const addElements = () => {  
  // create a new div element  
  const newElement = document.createElement("h1");  
  
  // set the h1's id  
  newElement.setAttribute("id", "sleeping-giant");  
  
  // and give it some content  
  const newContent = document.createTextNode("Jell-O, Burled!");  
  
  // add the text node to the newly created div  
  newElement.appendChild(newContent);  
  
  // add the newly created element and its content into the DOM  
  document.body.appendChild(newElement);  
  
  // append a second element to the DOM after the first one  
  const lastElement = document.createElement("div");  
  lastElement.setAttribute("id", "lickable-frog");  
  document.body.appendChild(lastElement);  
};  
// run script when page is loaded  
window.onload = addElements;
```

Notice that our function is now called `addElements`, plural, because we're appending two elements to the `body`. Save your Javascript file and refresh the

HTML file in the browser. When you inspect the page, you should now see two elements in the `body`, the `h1` and the `div` we added via Javascript.

Referencing a JS File vs. Using a Script Block

In our test example above, we referenced an external JS file, which contained our function to add new elements to the DOM. Typically, we would keep Javascript in a separate file, but we could also write a script block directly in our HTML file. Let's try it. First, we'll delete the script source so that we have an empty script block.

HTML

```
<!DOCTYPE html>  
<html>  
  <head>  
    <script type="text/javascript">  
      //Javascript goes here!  
    </script>  
  </head>  
  <body></body>  
</html>
```

Inside of our script block, we'll:

- create a `ul` element with no id
- create an `li` element with the id `dreamy-eyes`
- add the `li` as a child to the `ul` element
- add the `ul` element as the first child of the `body` element.

HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Cool Website</title>
    <script type="text/javascript">
      const addListElement = () => {
        const listElement = document.createElement("ul");
        const listItem = document.createElement("li");
        listItem.setAttribute("id", "dreamy-eyes");
        listElement.appendChild(listItem);
        document.body.prepend(listElement);
      };
      window.onload = addListElement;
    </script>
  </head>
  <body></body>
</html>
```

Refresh the HTML in your browser, inspect the page, and notice the `ul` and `li` elements that were created in the script block.