

# DOM

When the browser renders a HTML page, it creates an object, names it document and sends it to the console. We can use the document variable and access the page contents. Create a HTML page and open it in the browser.

```
<html>
<head>
  <title>Intro to the DOM</title>
</head>
<body>
  <h1>What I have learned at Coding Dojo</h1>
  <ul id="topics">
    <li class="subtopics">HTML</li>
    <li class="subtopics">CSS</li>
    <li class="subtopics">JavaScript</li>
    <li class="subtopics">Git</li>
  </ul>
</body>
</html>
```

Now that we have the HTML in the browser, let's console.log(document) in the console and see what it has.

```
> console.log(document);
VM109:1
#document
  <html>
    <head>
      <title>Intro to the DOM</title>
    </head>
    <body>
      <h1>What I have learned at Coding Dojo
      </h1>
      <ul id="topics">
        <li class="subtopics">HTML</li>
        <li class="subtopics">CSS</li>
        <li class="subtopics">JavaScript</li>
        <li class="subtopics">Git</li>
      </ul>
    </body>
  </html>
← undefined
```

We can also only access the body using console.log(document.body).

```
> console.log(document.body);  
VM262:1  
▼ <body>  
  <h1>What I have learned at Coding Dojo</h1>  
  ▼ <ul id="topics">  
    <li class="subtopics">HTML</li>  
    <li class="subtopics">CSS</li>  
    <li class="subtopics">JavaScript</li>  
    <li class="subtopics">Git</li>  
  </ul>  
</body>  
← undefined
```

If we use `console.dir(document)` to view the document object, we see a totally different set of tools we can use. The below is an abbreviated list, which contains properties, methods and events that the browser keeps track of. We can use many of these properties to interact with the browser and make our content dynamic.

```
> console.dir(document);  
VM363:1  
▼ #document  
  ▶ location: Location {href: "file:///Users/pk_...  
  ▶ implementation: DOMImplementation {}  
    URL: "file:///Users/pkrull/Documents/coding_...  
    documentURI: "file:///Users/pkrull/Document_...  
    compatMode: "BackCompat"  
    characterSet: "windows-1252"  
    charset: "windows-1252"  
    inputEncoding: "windows-1252"  
    contentType: "text/html"  
    doctype: null  
  ▶ documentElement: html  
    xmlEncoding: null  
    xmlVersion: null  
    xmlStandalone: false  
    domain: ""  
    referrer: ""  
    cookie: ""  
    lastModified: "02/14/2020 14:20:27"  
    readyState: "complete"  
    title: "Intro to the DOM"  
    dir: ""  
  ▶ body: body  
  ▶ head: head  
  ▶ images: HTMLCollection []  
  ▶ embeds: HTMLCollection []  
  ▶ plugins: HTMLCollection []  
  ▶ links: HTMLCollection []  
  ▶ forms: HTMLCollection []  
  ▶ scripts: HTMLCollection []  
    currentScript: null  
  ▶ defaultView: Window {parent: Window, opener_...  
    designMode: "off"  
    onreadystatechange: null  
  ▶ anchors: HTMLCollection []  
  ▶ applets: HTMLCollection []  
    fgColor: ""  
    linkColor: ""  
    vlinkColor: ""  
    alinkColor: ""  
    bgColor: ""  
  ▶ all: HTMLAllCollection(10) [html, head, _...  
  ▶ scrollingElement: body  
    onpointerlockchange: null  
    onpointerlockerror: null  
    hidden: false  
    visibilityState: "visible"
```

In this video, we will show you how we can tie everything we have learned so far, and make page interactive.