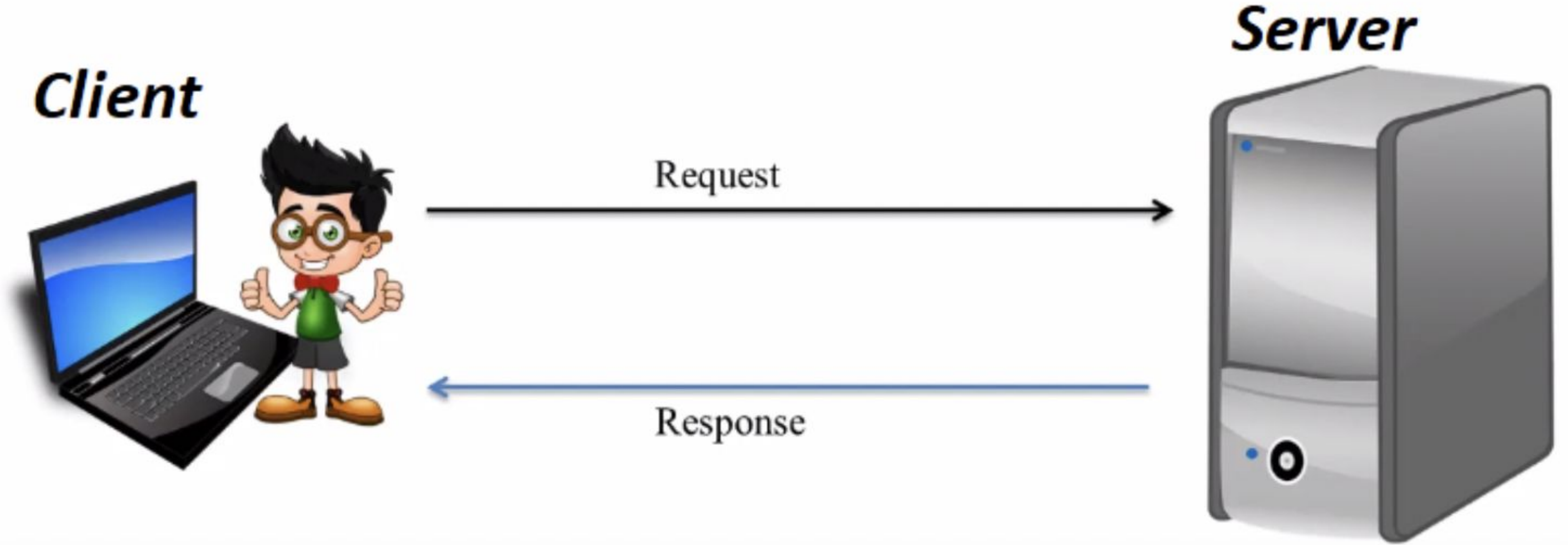
Client Server Architecture

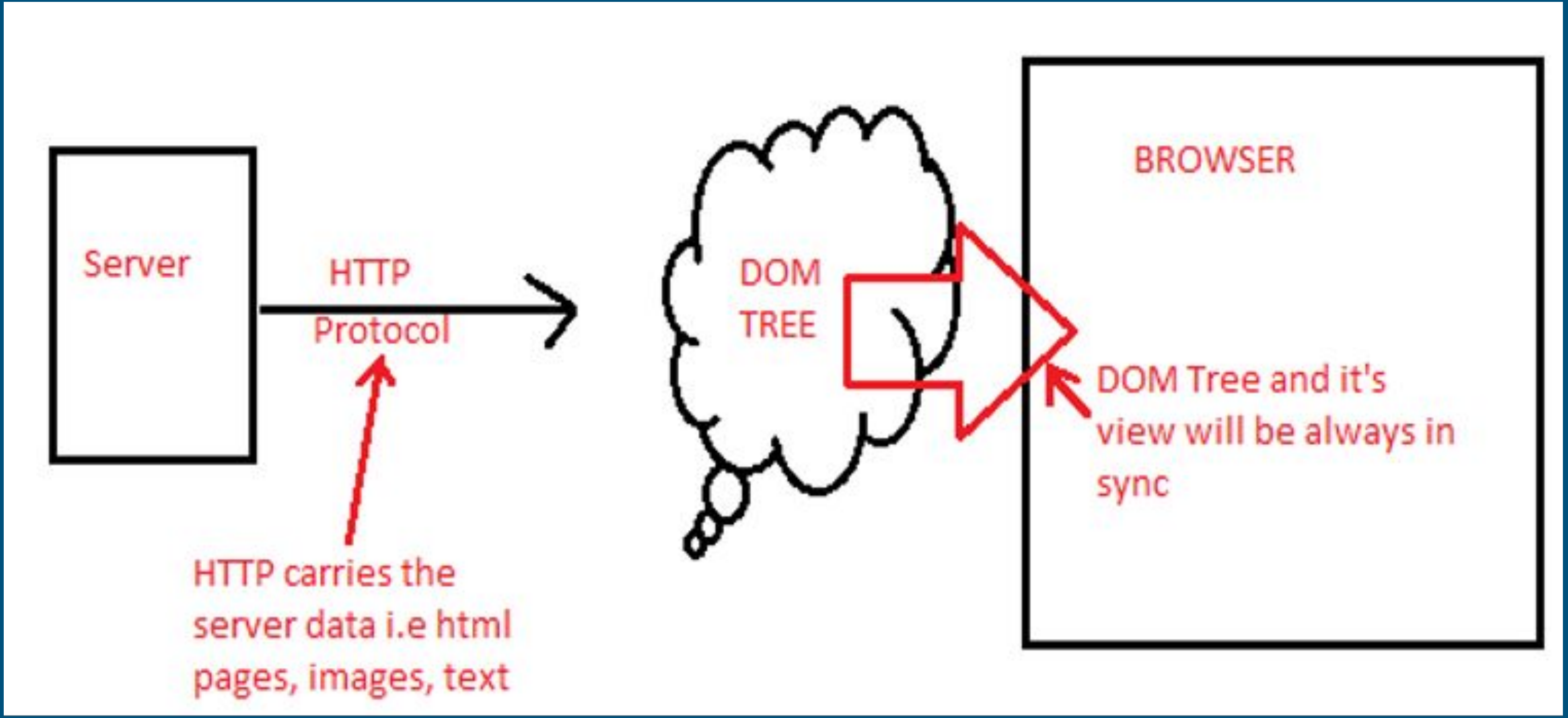
Client: A client is a piece of computer hardware or software that accesses a service which made available by Server. Ex. Browser

Server: A server is a Computer program or a device that provides functionality for other programs or a device called client.



DOM - Document Object Model

- The DOM defines a standard for accessing documents
- In the DOM all HTML elements are defined as objects



HTTP carries the server data i.e html pages, images, text

DOM Tree and it's view will be always in sync

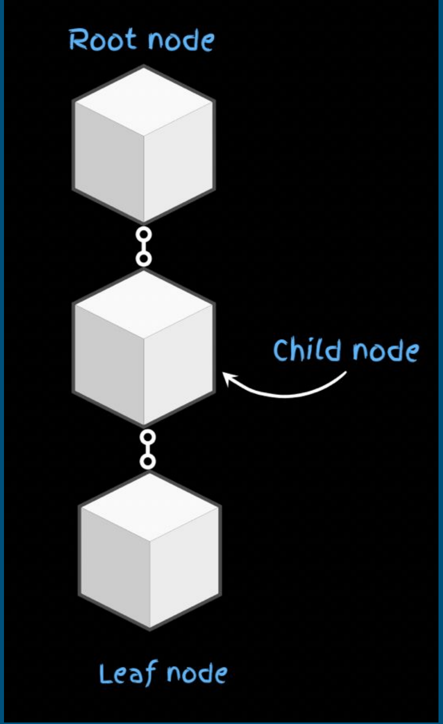
DOM - Document Object Model

- When a web page is loaded, the browser creates a Document Object Model of the page.
- The DOM defines a standard for accessing documents

- The Document Object Model represents each of these web pages in a tree-like structure to make it easier to access and manage the Elements.
- The DOM is the tree of nodes corresponding to HTML elements on a page

DOM Defines

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The events for all HTML elements



window object

- Window is the main container or global object. Any operation related to entire browser window can be a part of window object

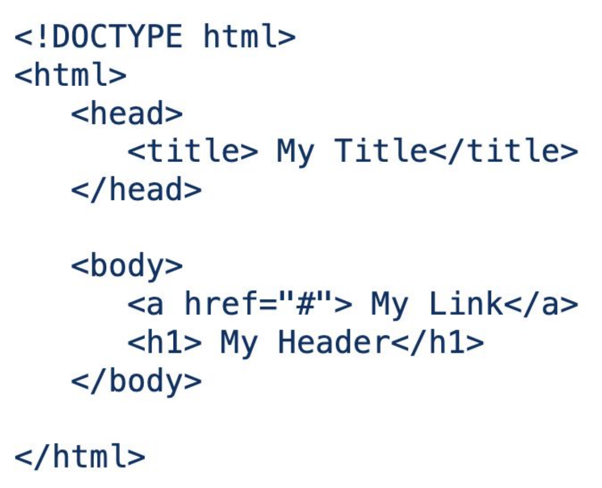
- Window has methods, properties and objects. Ex. setTimeout() or setInterval().
Where as document is the object of the Window
And It also has a screen object with properties describing the physical display.

document

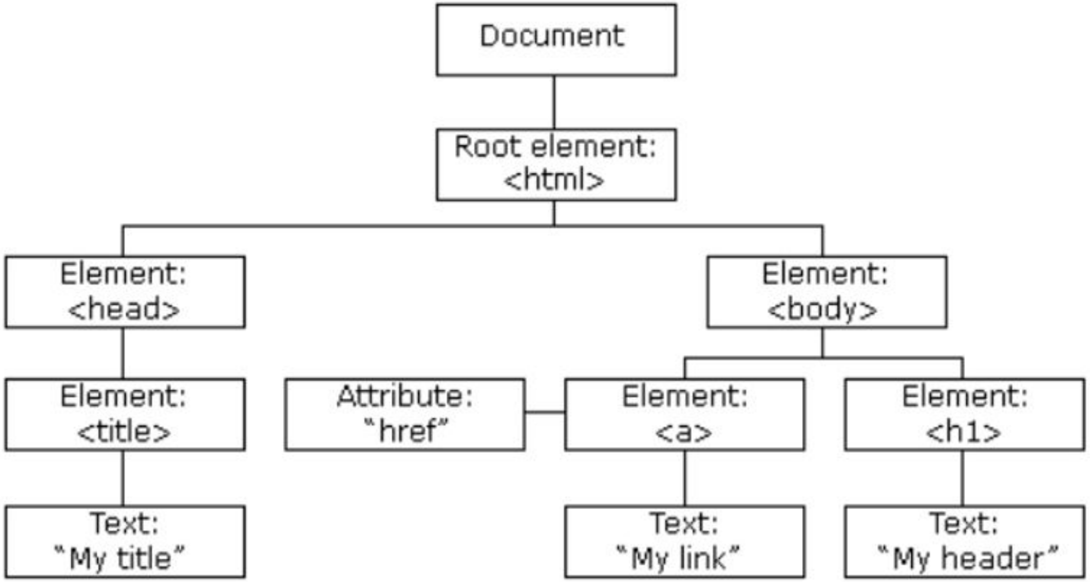
- The DOM is the child of window object
- Document is used to render the HTML elements
- Inspect in browser and type on console window, we can see document is the child object window

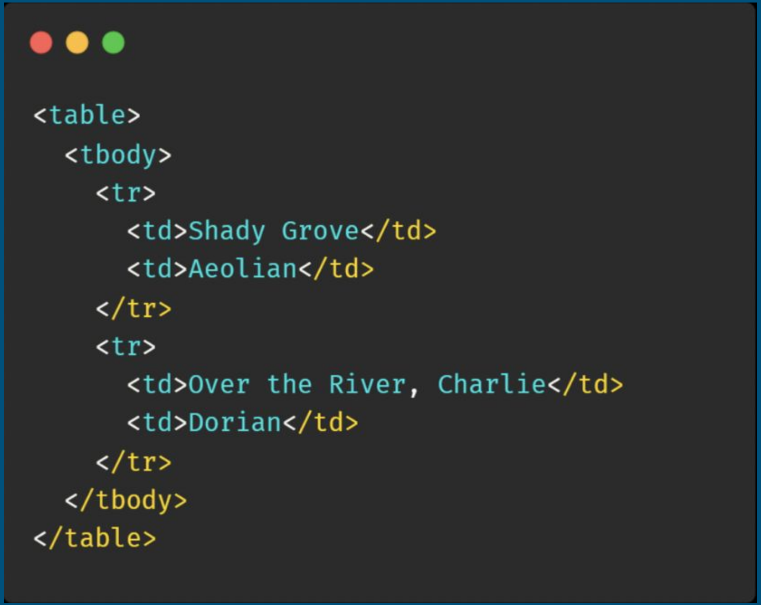
If we want to see the location of our page that we are in → window.location or window.location.href
console.dir(document) → to see the complete document object

HTML code it’s DOM tree representation



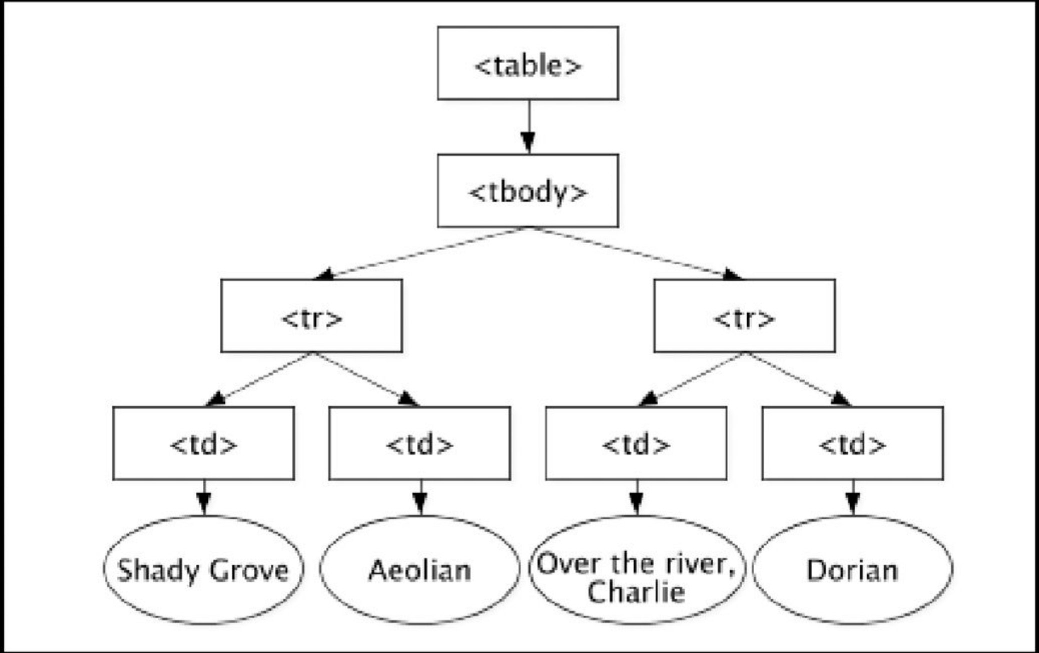
```
<!DOCTYPE html>
<html>
   <head>
      <title> My Title</title>
   </head>

   <body>
      <a href="#"> My Link</a>
      <h1> My Header</h1>
   </body>

</html>
```





```
<table>
  <tbody>
    <tr>
      <td>Shady Grove</td>
      <td>Aeolian</td>
    </tr>
    <tr>
      <td>Over the River, Charlie</td>
      <td>Dorian</td>
    </tr>
  </tbody>
</table>
```

The Document Object Model of this code can be created like this:



With the object model, JavaScript gets all the power it needs to create dynamic HTML

- JavaScript can create new HTML events in the page
- JavaScript can change all the HTML elements in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can change all the HTML attributes in the page

List of common APIs in web and XML page scripting using the DOM.

- document.getElementById(‘element_id’)
- document.querySelector(selector)
- document.querySelectorAll(name)
- document.createElement(name)
- parentNode.appendChild(node)
- element.innerHTML
- element.style.left
- element.setAttribute()
- element.getAttribute()
- element.addEventListener()
- window.content

querySelector( ), getElementById( )

- Return single element
- Different ways of querying element (By CSS selector, by ID)
- Direct reference to DOM element is returned

querySelectorAll( ), getElementByTagName( )

- Return collection of elements (array - like objects): NodeList
- Different ways of querying element (By CSS selector, by Tag Name, by CSS class)
- querySelectorAll() returns a non-live NodeList, getXByY( ) returns a live NodeList

What is a Node in the context of DOM?

Every item in the DOM tree is called a node, There are two types of node

1. An Element Node: Node that has an element
2. Text Node: Node that has text

Child Node: A node which is a child of another node

Parent Node: A node which has one or more child

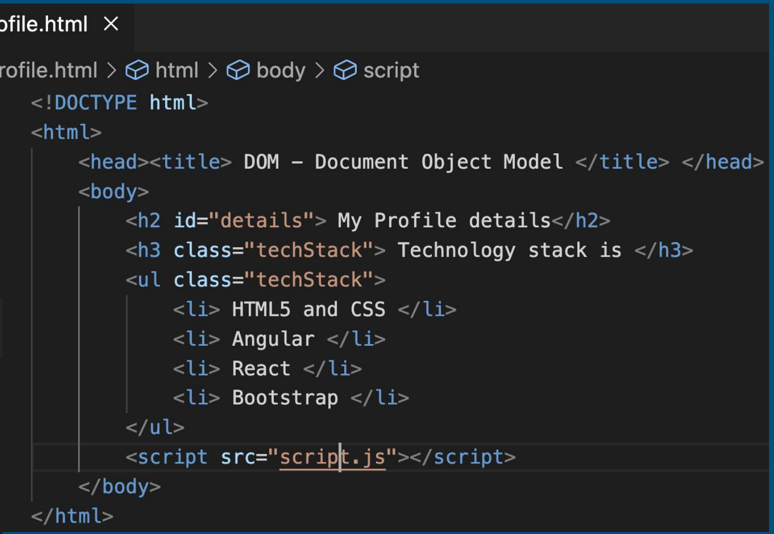
Sibling Node: A node that share the same parent node

Descendant Node: A node which is nested deep in the tree

Finding HTML Element

getElementById('element_id');

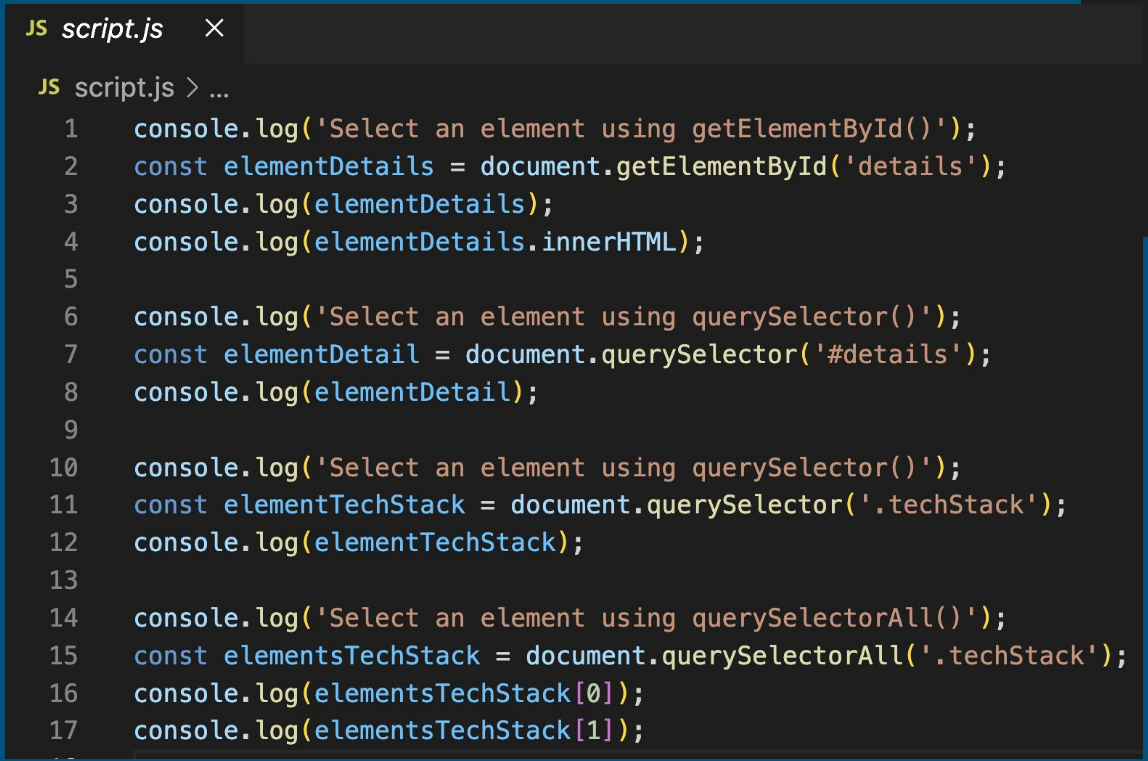
querySelector(selector), querySelectorAll(selector);



profile.html

profile.html > html > body > script

```html
<!DOCTYPE html>
<html>
    <head><title> DOM - Document Object Model </title> </head>
    <body>
        <h2 id="details"> My Profile details</h2>
        <h3 class="techStack"> Technology stack is </h3>
        <ul class="techStack">
            <li> HTML5 and CSS </li>
            <li> Angular </li>
            <li> React </li>
            <li> Bootstrap </li>
        </ul>
        <script src="script.js"></script>
    </body>
</html>
```

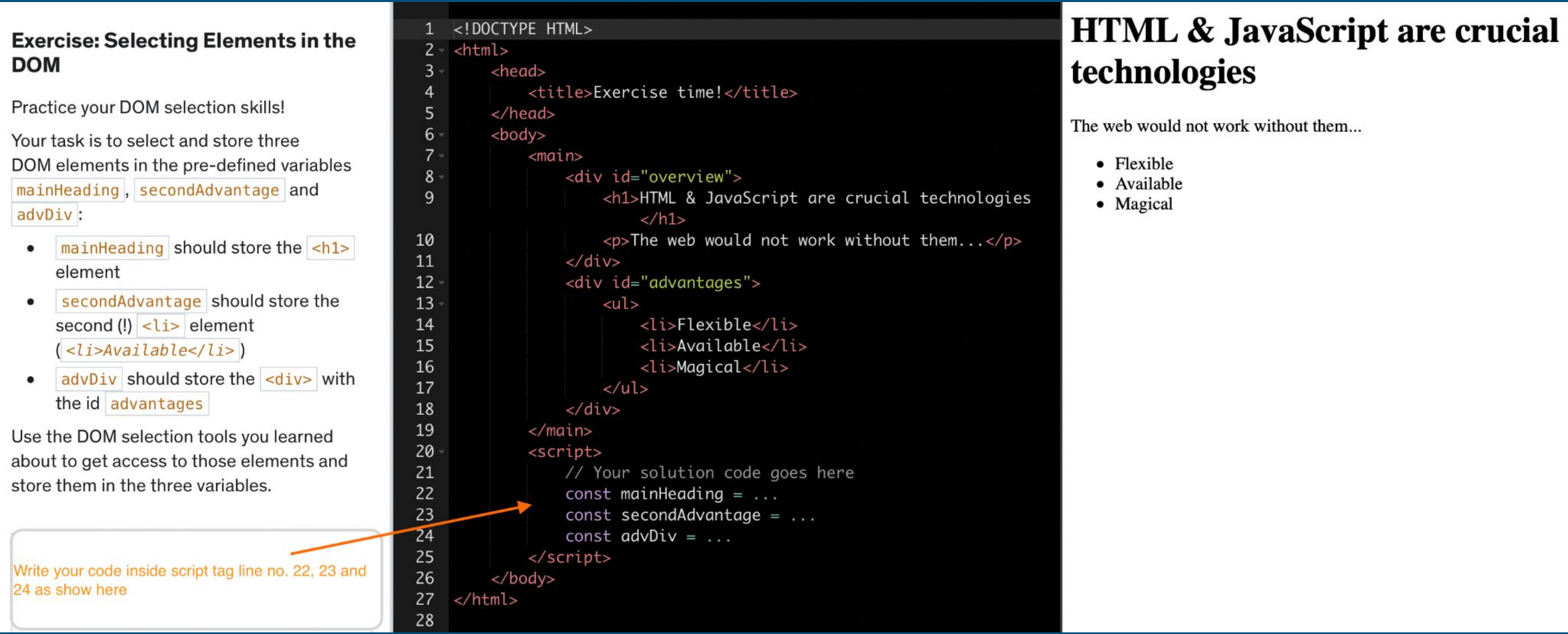


script.js

script.js > ...

```javascript
console.log('Select an element using getElementById()');
const elementDetails = document.getElementById('details');
console.log(elementDetails);
console.log(elementDetails.innerHTML);

console.log('Select an element using querySelector()');
const elementDetail = document.querySelector('#details');
console.log(elementDetail);

console.log('Select an element using querySelector()');
const elementTechStack = document.querySelector('.techStack');
console.log(elementTechStack);

console.log('Select an element using querySelectorAll()');
const elementsTechStack = document.querySelectorAll('.techStack');
console.log(elementsTechStack[0]);
console.log(elementsTechStack[1]);
```

Assignment to select DOM elements



Exercise: Selecting Elements in the DOM

Practice your DOM selection skills!

Your task is to select and store three DOM elements in the pre-defined variables `mainHeading`, `secondAdvantage` and `advDiv`:

- `mainHeading` should store the `<h1>` element
- `secondAdvantage` should store the second (!) `<li>` element (*`<li>Available</li>`*)
- `advDiv` should store the `<div>` with the id `advantages`

Use the DOM selection tools you learned about to get access to those elements and store them in the three variables.

Write your code inside script tag line no. 22, 23 and 24 as show here

```
1  <!DOCTYPE HTML>
2  <html>
3      <head>
4          <title>Exercise time!</title>
5      </head>
6      <body>
7          <main>
8              <div id="overview">
9                  <h1>HTML & JavaScript are crucial technologies
                       </h1>
10                 <p>The web would not work without them...</p>
11             </div>
12             <div id="advantages">
13                 <ul>
14                     <li>Flexible</li>
15                     <li>Available</li>
16                     <li>Magical</li>
17                 </ul>
18             </div>
19         </main>
20         <script>
21             // Your solution code goes here
22             const mainHeading = ...
23             const secondAdvantage = ...
24             const advDiv = ...
25         </script>
26     </body>
27 </html>
28
```

HTML & JavaScript are crucial technologies

The web would not work without them...

- Flexible
- Available
- Magical