

Document object model – DOM

You'll learn how the browser builds HTML documents.

Materials and Resources

Material	Time
JavaScript Tutorial For Beginners #32 - What is the DOM in JavaScript?	4:31
JavaScript Tutorial For Beginners #33 - Traversing the DOM	7:43
JavaScript: Browser - querySelector(), querySelector	5:06
JavaScript Tutorial For Beginners #34 - Changing Page Content	5:38
JavaScript Tutorial For Beginners #35 - Changing Element Attributes	6:02
JavaScript Tutorial For Beginners #36 - Changing CSS Styles	4:49
JavaScript Tutorial For Beginners #37 - Adding Elements to the DOM	7:43
JavaScript Tutorial For Beginners #38 - Removing Elements from the DOM	4:12

Material

Time

[JavaScript Tutorial For Beginners #40 - The onClick Event](#)

8:04

<https://www.youtube.com/watch?v=XQEfWd1lh4Q>

Reading

Material

[What is the DOM?](#)

<https://css-tricks.com/dom/>

[Introduction to the DOM](#)

https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction

[Document Object Model \(DOM\)](#)

https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model

Digging deeper

Read more about the DOM in the Eloquent Javascript book's relevant chapter:

http://eloquentjavascript.net/13_dom.html

https://eloquentjavascript.net/14_dom.html

Material Review

- What is the Window object?
- What is the window.document object?
- What is the DOM?
- Is the DOM part of the JavaScript?
- What is the difference between Node and Element?
- How can you access elements?
 - By ID?

- By class?
- By tag?
- By any css selector?
- What is the difference between `querySelector` and `querySelectorAll`?
- What's the difference between `querySelectorAll('.class')` and `getElementsByClassName('class')`
- How can you create a new Element?
- How can you append an Element to the DOM?
- How can you move an element in the DOM?
- How can you remove an Element from the DOM?
- How can you change the text content of an Element?
- Whats the difference between `innerText`, `textContent` and `innerHTML`?
- How can you set/get an attribute?
- How can you add custom attributes to an Element?
- How can you modify an Element's style?
- How can you add a new `class` to an Element?
- What is the difference between the `className` and `classList` property?
- What does the `classList.toggle` used for?
- What is the `onClick` property used for?

Workshop

Including JavaScript in an HTML page

Within the `<script>` tag you can run JavaScript code or if you specify the `src` attribute you can load a `.js` file.

```
<script>
  console.log('Hello World!');
</script>

<script src="something.js"></script>
```

Selecting elements

```
const king = document.getElementById('b325');
const lamplighter = document.querySelector('.b329');
const asteroids = document.querySelectorAll('div.asteroid');

for (let i = 0; i < asteroids.length; i++) {
  console.log(asteroids[i]);
}
```

Example

Open the developer tools in your browser to see the results of `console.Log`
You can find the above JavaScript code in HTML format:

- [accessing-elements.html](#)

```
<!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>DOM: Accessing Elements</title>
</head>
<body>
  <div id="b325" class="asteroid">The King</div>
  <div class="asteroid b326">The Conceited Man</div>
  <div class="asteroid b328">The Businessman</div>
  <div class="asteroid b329">The Lamplighter</div>

  <script>
    'use strict';

    const king = document.getElementById('b325');
    console.log('king', king);
    const lamplighter = document.querySelector('.b329');
    console.log('lamplighter', lamplighter);

    const asteroids = document.querySelectorAll('div.asteroid');
    for (let i = 0; i < asteroids.length; i++) {
      console.log(asteroids[i]);
    }
  </script>
</body>
</html>
```

Exercises

- [querySelector-1.html](#)

```
<!DOCTYPE html>

<html lang="en">
```

```

<head>
  <meta charset="UTF-8">
  <title>Workshop: QuerySelector</title>
</head>
<body>
  <section class="container">
    <div id="b325" class="asteroid">The King</div>
    <div class="asteroid b326">The Conceited Man</div>
  </section>
  <p class="asteroid big">The Businessman</p>
  <div class="asteroid b329 big">The Lamplighter</div>

  <!-- You can work in the html or in a separate js file. Like:
  <script>
    1. store the element that says 'The King' in the 'king' variable.
       console.log it.
    2. store 'The Businessman' and 'The Lamplighter'
       in the 'businessLamp' variable.
       console.log each of them.
    3. store 'The King' and 'The Conceited Man'
       in the 'conceitedKing' variable.
       alert them one by one.
    4. store 'The King', 'The Conceited Man' and 'The Lamplighter'
       in the 'noBusiness' variable.
       console.log each of them.
  </script>
  OR
  <script src="queryselector-1.js"></script> -->
</body>
</html>

```

An element's contents

```

const asteroid = document.querySelector('.asteroid');

console.log(asteroid.innerHTML);
console.log(asteroid.textContent);
asteroid.innerHTML = 'This is your <strong>new content!</strong>';

```

Example

You can find the above JavaScript code in HTML format:

- [element-contents.html](#)

```

<!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>DOM: Element Contents</title>
</head>
<body>
  <p class="container">This was my <strong>original content</strong>.</p>
  <script>
    'use strict';

    const container = document.querySelector('.container');
    console.log('innerHTML:', container.innerHTML);
    console.log('textContent:', container.textContent);
    container.innerHTML = 'This is your <strong>new content!</strong>';
  </script>
</body>
</html>

```

Footer

© 2022 GitHub, Inc.

Footer navigation

- [Terms](#)

Exercises: innerHTML, textContent

- [content-1.html](#)

```

<!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Workshop: innerHTML, textContent</title>
</head>
<body>
  <p class="apple">apple</p>
  <p class="balloon">banana</p>
  <p class="cat">car</p>
  <p class="animals">goat rabbit <strong>cat</strong> dog</p>

  <!-- You can work in the html or in a separate js file. Like:
  <script>
    1) Append the content of the last paragraph to the other paragraphs.

```

```

2) Do the same again, but you should keep the cat strong.
</script>
OR
<script src="content-1.js"></script> -->
</body>
</html>

```

- [content-2.html](#)

```

<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Workshop: innerHTML, textContent</title>
  </head>
  <body>
    <ul>
      <li>placeholder</li>
      <li>placeholder</li>
      <li>placeholder</li>
      <li>placeholder</li>
    </ul>

    <!-- You can work in the html or in a separate js file. Like:
    <script>
      1) replace the list items' content with items from this list:
      ['apple', 'banana', 'cat', 'dog']
      2) change the <ul> element's background color to 'limegreen'
         - use css class to change the color instead of the style property
    </script>
    OR
    <script src="content-2.js"></script> -->
  </body>
</html>

```

Element attributes

```

const littlePrince = document.querySelector('img');
console.log(littlePrince.getAttribute('src'));

```

```
littlePrince.setAttribute('src', 'http://deji.chez.com/dessins/pp1.gif');
```

Example

You can find the above JavaScript code in HTML format:

- [element-attributes.html](#)

```
<!DOCTYPE html>

<html lang="en">

  <head>
    <meta charset="UTF-8">
    <title>DOM: Element Attributes</title>
  </head>

  <body>
    
    <script>
      'use strict';

      const littlePrince = document.querySelector('img');
      console.log('original image:', littlePrince.getAttribute('src'));
      littlePrince.setAttribute('src',
'http://deji.chez.com/dessins/pp1.gif');
    </script>
  </body>

</html>
```

Exercises

- [attribute-1.html](#)

```
<!DO
CTYP
```



```
E
html
>
```

```
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Workshop: Attributes</title>
  </head>
  <body>

    
    <a href="http://www.themostamazingwebsiteontheinternet.com/">The Best Website
Ever</a>
    <button>Click me!</button>
    <button class="this-one">Click me!</button>

    <!-- You can work in the html or in a separate js file. Like:
    <script>
      Write the image's url to the console.
      Replace the image with a picture of your favorite animal.
      Make the link point to the Green Fox Academy website.
      Disable the second button.
      Replace its text with 'Don't click me!'.
    </script>
    OR
    <script src="attribute-1.js"></script> -->

  </body>
</html>
```

Classes

```
const asteroid = document.querySelector('div');

console.log(asteroid.classList.value);

console.log('asteroid?', asteroid.classList.contains('asteroid'));
```

```
console.log('planet?', asteroid.classList.contains('planet'));

asteroid.classList.add('new-class');
asteroid.classList.remove('asteroid');
console.log('still asteroid?', asteroid.classList.contains('asteroid'));
```

Example

The `.classList` property of a DOM node contains useful methods to analyze its classes.

You can find the above JavaScript code in HTML format:

- [classlist.html](#)

```
<!DOCTYPE html>

    <html lang="en">

        <head>
            <meta charset="UTF-8">
            <title>DOM: ClassList</title>
        </head>

        <body>
            <div class="asteroid b325">an asteroid</div>

            <script>
                'use strict';

                const asteroid = document.querySelector('div');
                console.log(asteroid.classList.value);
                console.log('asteroid?', asteroid.classList.contains('asteroid'));
                console.log('planet?', asteroid.classList.contains('planet'));
                asteroid.classList.add('new-class');
                asteroid.classList.remove('asteroid');
                console.log('still asteroid?',
asteroid.classList.contains('asteroid'));
            </script>
        </body>

    </html>
```

Exercises

- [classlist-1.html](#)

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Workshop: ClassList</title>
    <style>
      .red {
        color: red;
      }
      .balloon {
        background-color: lime;
        border-radius: 100%;
      }
      .pear {
        background: lime;
        padding: 2rem;
        text-decoration: uppercase;
      }
    </style>
  </head>
  <body>
    <p class="apple">apple</p>
    <p class="balloon">balloon</p>
    <p class="cat red-dot">cat</p>
    <p>dolphin</p>

    <!-- You can work in the html or in a separate js file. Like:
    <script>
      If the fourth p has a 'dolphin' class, replace apple's content with
      'pear'
      If the first p has an 'apple' class, replace cat's content with
      'dog'
      Make apple red by adding a .red class
      Make balloon less balloon-like (change its shape)
    </script>
    OR
    <script src="classlist-1.js"></script> -->
  </body>
</html>
```

Creating, inserting & removing elements

```
const asteroidList = document.querySelector('ul.asteroids');
const newAsteroid = document.createElement('li');

newAsteroid.id = 'b555';
newAsteroid.textContent = 'The Green Fox';
asteroidList.appendChild(newAsteroid);

const businessAsteroid = document.querySelector('.b328');
asteroidList.removeChild(businessAsteroid);
```

Example

You can find the above JavaScript code in HTML format:

- [create-insert-remove.html](#)

```
<!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>DOM: Create, Insert, Remove</title>
</head>
<body>
  <ul class="asteroids">
    <li class="asteroid">The King</li>
    <li class="asteroid b326">The Conceited Man</li>
    <li class="asteroid b328">The Businessman</li>
    <li class="asteroid b329">The Lamplighter</li>
  </ul>

  <script>
    'use strict';

    const asteroidList = document.querySelector('ul.asteroids');

    const newAsteroid = document.createElement('li');
    newAsteroid.id = 'b555';
    newAsteroid.textContent = 'The Green Fox';
    asteroidList.appendChild(newAsteroid);
```

```

        const businessAsteroid = document.querySelector('.b328');
        asteroidList.removeChild(businessAsteroid);
    </script>
</body>
</html>

```

Exercises

- [createelement-1.html](#)

```

<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Workshop: CreateElement</title>
  </head>
  <body>
    <ul class="asteroids">
      <li id="b325" class="asteroid">The King</li>
      <li class="asteroid b326">The Conceited Man</li>
    </ul>
    <div class="container"></div>

    <!-- You can work in the html or in a separate js file. Like:
    <script>
      Add an item that says 'The Green Fox' to the asteroid list.
      Add an item that says 'The Lamplighter' to the asteroid list.
      Add a heading saying 'I can add elements to the DOM!' to the
.container.
      Add an image, any image, to the container.
    </script>
    OR
    <script src="createelement-1.js"></script> -->
  </body>
</html>

```

- [createelement-2.html](#)

```
<!DOCTYPE html>
```

```
  <html lang="en">
    <head>
      <meta charset="UTF-8">
      <title>Workshop: CreateElement</title>
    </head>
    <body>
      <ul class="asteroids">
        <li>The King</li>
      </ul>
```

```
    <!-- You can work in the html or in a separate js file. Like:
```

```
    <script>
```

```
      Remove the king from the list.
```

```
      Fill the list based on the following list of objects.
```

```
      Only add the asteroids to the list.
```

```
      Each list item should have its category as a class and its content
    as text content. -->
```

```
    const planetData = [
      {
        category: 'inhabited',
        content: 'Foxes',
        asteroid: true,
      },
      {
        category: 'inhabited',
        content: 'Whales and Rabbits',
        asteroid: true,
      },
      {
        category: 'uninhabited',
        content: 'Baobabs and Roses',
        asteroid: true,
      },
      {
        category: 'inhabited',
        content: 'Giant monsters',
        asteroid: false,
      },
      {
        category: 'inhabited',
```

```

        content: 'Sheep',
        asteroid: true,
    },
];
<!-- </script>
OR
<script src="createelement-2.js"></script> -->
</body>
</html>

```

- [createelement-3.html](#)

```

<!DOCTYPE html>

<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>Workshop: CreateElement</title>
</head>

<body>
  <main></main>

  <!-- You can work in the html or in a separate js file. Like:
  <script>
    const kids = [
      { 'petName': 'Wattled crane', 'owner': 'Bryant' },
      { 'petName': 'Devil, tasmanian', 'owner': 'Alejandro' },
      { 'petName': 'Mynah, common', 'owner': 'Nelie' },
      { 'petName': 'Dolphin, common', 'owner': 'Mariele' },
      { 'petName': 'Gray duiker', 'owner': 'Maddalena' },
      { 'petName': 'Crab (unidentified)', 'owner': 'Lucine' },
      { 'petName': 'Ant (unidentified)', 'owner': 'Lorianna' },
      { 'petName': 'Bison, american', 'owner': 'Tommie' },
      { 'petName': 'Yellow mongoose', 'owner': 'Vivyan' },
      { 'petName': 'Carpet snake', 'owner': 'Veda' },
      { 'petName': 'Lesser mouse lemur', 'owner': 'Isidor' },
    ];
    1) Create an <article> element for each kid
    2) Create a <h3> and a <p> element for each article and append them
  </script>

```

```

    as a child to the <article>
        - The H3 should contain the owner's name
        - The p should contain the pet's name
    3) Add the article to the pets main.
</script>
OR
<script src="createelement-3.js"></script> -->
</body>

</html>

```

- [moon-and-stars.md](#)

onClick

```

const button = document.querySelector('button');

button.onclick = () => {
  alert('Green Fox!');
};

```

Example

You can find the above JavaScript code in HTML format:

- [onclick.html](#)

```

<!DOCTYPE html>

<html lang="en">

  <head>
    <meta charset="UTF-8">
    <title>Workshop: onclick</title>
    <style>
      body {
        margin: 0;
      }

      div {
        height: 100vh;

```



```

    }

    .party {
        background:
url(https://media.giphy.com/media/4c4jQPdVv381G/giphy.gif);
    }
</style>
</head>

<body>
    <div>
        <button>Party</button>
    </div>

    <!-- You can work in the html or in a separate js file. Like:
    <script>
        Turn the party (i.e. the whole page background) on and off by
clicking the button.
    </script>
    OR
    <script src="events-1.js"></script> -->
</body>

</html>

```

Exercises

- [onclick.html](#)

```

<!DOCTYPE html>

<html lang="en">

    <head>
        <meta charset="UTF-8">
        <title>Workshop: onclick</title>
        <style>
            body {

```

```

        margin: 0;
    }

    div {
        height: 100vh;
    }

    .party {
        background:
url(https://media.giphy.com/media/4c4jQPdVv381G/giphy.gif);
    }
</style>
</head>

<body>
    <div>
        <button>Party</button>
    </div>

    <!-- You can work in the html or in a separate js file. Like:
    <script>
        Turn the party (i.e. the whole page background) on and off by
clicking the button.
    </script>
    OR
    <script src="events-1.js"></script> -->
</body>

</html>

```

- [list-manager.html](#) Build up the structure.

```

<!DOCTYPE html>

<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

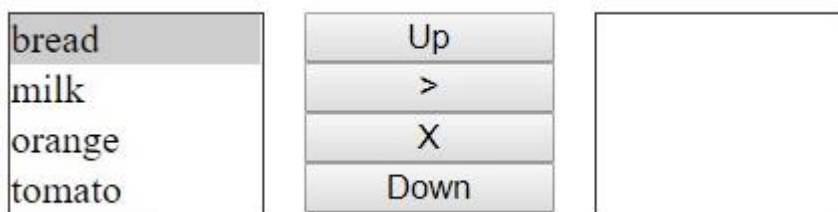
```

```

<link rel="stylesheet" type="text/css" href="main.css">
<title>List Manager</title>
</head>
<body>
  <!-- Do not modify this file -->
  <!--
    1, Build up the necessary structure with JavaScript (try to use
    semantically correct elements)
    2, Achive the same design by css
    3, Try not to use css classes
    4, Add some functionality to the buttons
      - If "Up" is clicked the selection should move up by one
      - If "Down" is clicked the selection should move down by one
      - If "X" is clicked the selected item should be removed and the
first item should be selected
      - If ">" is clicked the selected item should be moved to the
right side and the first item on the left side should be selected
    5, Check all the edge cases, no error should be printed to the
console
  -->
  <script type="text/javascript" src="main.js"></script>
</body>

```

Build up this structure:



SVG Elements

The SVG image format is a vector format based on XML and it can be embedded to the DOM.

```
<svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <rect x="10" y="10" height="100" width="180"
  style="stroke:#ff0000; stroke-width: 10; fill: #0dc453"/>
</svg>
```

Example

You can find the above JavaScript code in HTML format:

- [svg-1.html](#)

```
<!DOCTYPE html>

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <style>
    svg {
      border: 1px solid black;
    }
  </style>
</head>
<body>
<h1>YAY, SVG!</h1>
<svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <rect x="10" y="10" height="100" width="180" style="stroke:#ff0000;
stroke-width: 10; fill: #0dc453"/>
</svg>
<ul>
  <li>The black border is coming from CSS</li>
  <li>The red border is coming from SVG</li>
</ul>
</body>
</html>
```

Exercises

- [svg-1.html](#)

```

<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <style>
      svg {
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <h1>YAY, SVG!</h1>
    <svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
      <rect x="10" y="10" height="100" width="180" id="very-rectangle"
style="stroke:#ff0000; stroke-width: 10; fill: #0dc453"/>
    </svg>
    <ul>
      <li>The black border is coming from CSS</li>
      <li>The red border is coming from SVG</li>
    </ul>

    <!-- You can work in the html or in a separate js file. Like:
    <script>
      1) Select the rectangle by the ID: "very-rectangle"
      2) Change the rectangle's position to x:50, y:50
      3) Change its fill to tomato
    </script>
    OR
    <script src="svg-1.js"></script> -->

  </body>
</html>

```