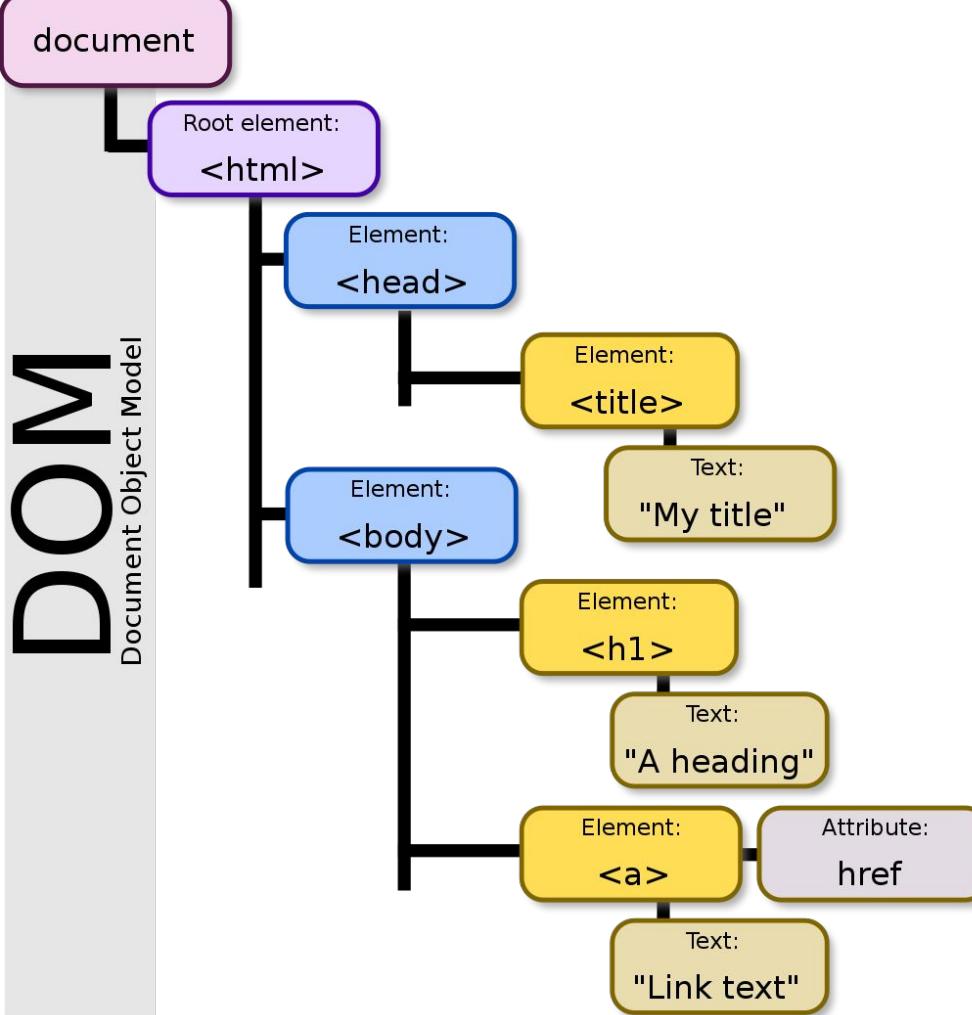


DOM STRUCTURE

Document Object Model Manipulation

DOM

Document Object Model



Document Object Model

The **Document Object Model (DOM)** is a [cross-platform](#) and [language-independent](#) interface that treats an [XML](#) or [HTML](#) document as a [tree structure](#) wherein each [node](#) is an [object](#) representing a part of the document. The DOM represents a document with a logical tree. Each branch of the tree ends in a node, and each node contains objects. [DOM methods allow programmatic access to the tree; with them one can change the structure, style or content of a document.](#) Nodes can have event handlers attached to them. Once an event is triggered, the event handlers get executed.

Manipulating the DOM

The DOM was designed to be independent of any particular programming language, making the structural representation of the document available from a single, consistent API. Though we focus exclusively on JavaScript in this reference documentation, implementations of the DOM can be built for any language:

```
<p>One</p>
<p>Two</p>
<p>Three</p>
```

```
var paragraphs = document.getElementsByTagName("p"); // paragraphs[0] is the
first <p> element // paragraphs[1] is the second <p> element, etc.
alert(paragraphs[0].nodeName);
```

Property Accesors

Property accessors provide access to an object's properties by using the dot notation or the bracket notation:

```
1 var person = {};
2 person['firstname'] = 'Mario';
3 person['lastname'] = 'Rossi';
4
5 console.log(person.firstname);
6 // expected output: "Mario"
7
8 person = { 'firstname': 'John', 'lastname': 'Doe' }
9
10 console.log(person['lastname']);
11 // expected output: "Doe"
12
```

JavaScript HTML DOM Elements

```
document.getElementById(" ");
```

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_getelementbyid

```
document.getElementsByTagName(" ");
```

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_getelementsbytagname2

```
document.getElementsByClassName(" ");
```

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_getelementsbyclassname

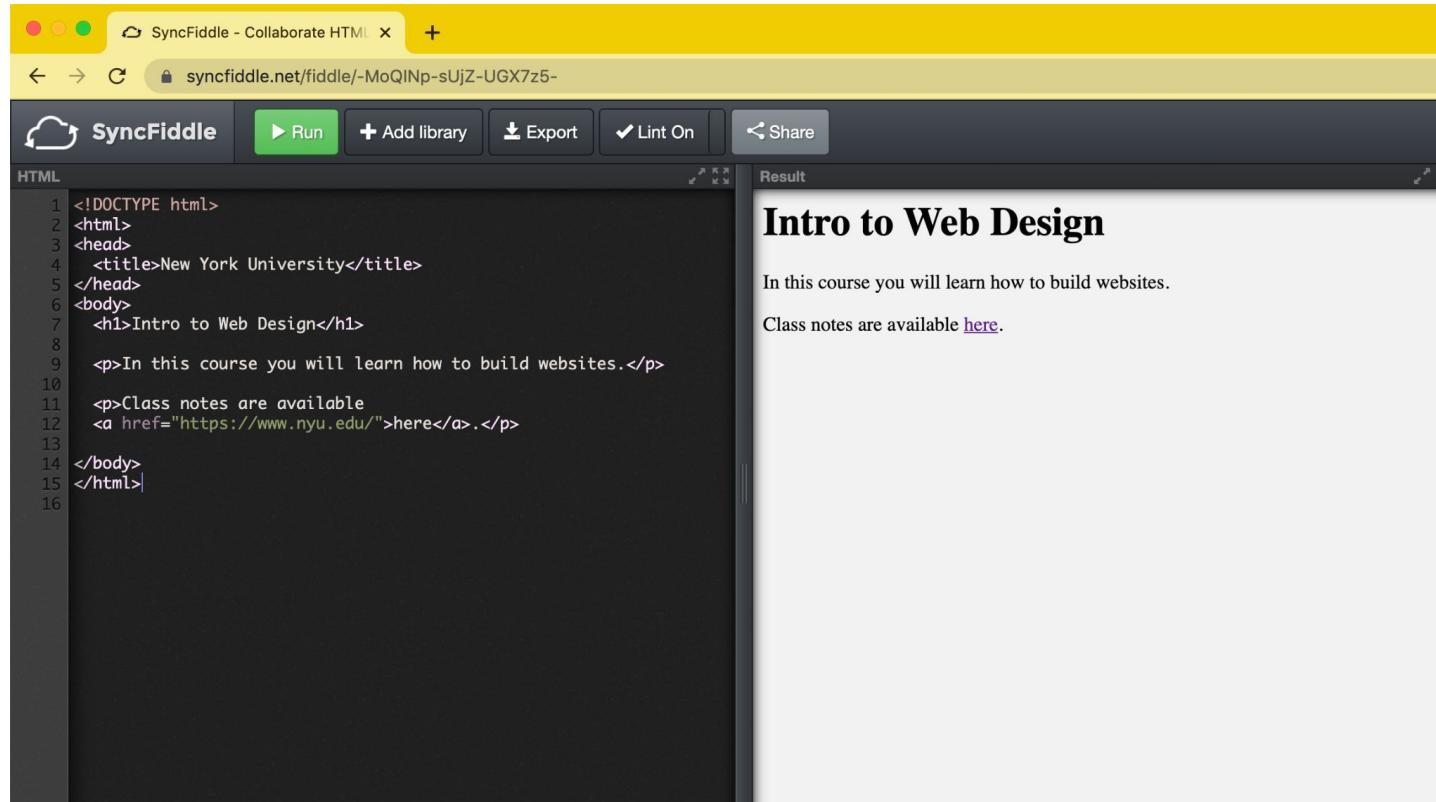
```
document.querySelectorAll(" "); (CSS selector)
```

https://www.w3schools.com/js/tryit.asp?filename=tryjs_dom_queryselectorall

The following HTML objects (and object collections) are also accessible:

- [document.anchors](#)
- [document.body](#)
- [document.documentElement](#)
 - [document.embeds](#)
 - [document.forms](#)
 - [document.head](#)
 - [document.images](#)
 - [document.links](#)
 - [document.scripts](#)
 - [document.title](#)

Let's Practice: <https://syncfiddle.net/fiddle/-MoQINp-sUjZ-UGX7z5->



The screenshot shows the SyncFiddle web application interface. The top navigation bar includes a yellow header with three dots, a back/forward button, a search bar containing the URL, and a plus sign for creating new fiddles. Below the header is a toolbar with a cloud icon labeled "SyncFiddle", a "Run" button, an "Add library" button, an "Export" button, a "Lint On" checkbox, and a "Share" button.

The main area is divided into two sections: "HTML" on the left and "Result" on the right. The "HTML" section contains the following code:

```
<!DOCTYPE html>
<html>
<head>
<title>New York University</title>
</head>
<body>
<h1>Intro to Web Design</h1>
<p>In this course you will learn how to build websites.</p>
<p>Class notes are available
<a href="https://www.nyu.edu/">here</a>.</p>
</body>
</html>
```

The "Result" section displays the rendered HTML output:

Intro to Web Design

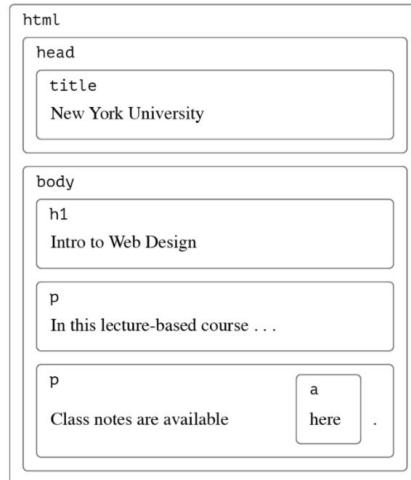
In this course you will learn how to build websites.

Class notes are available [here](https://www.nyu.edu/).

The DOM

Introduction to Web Design

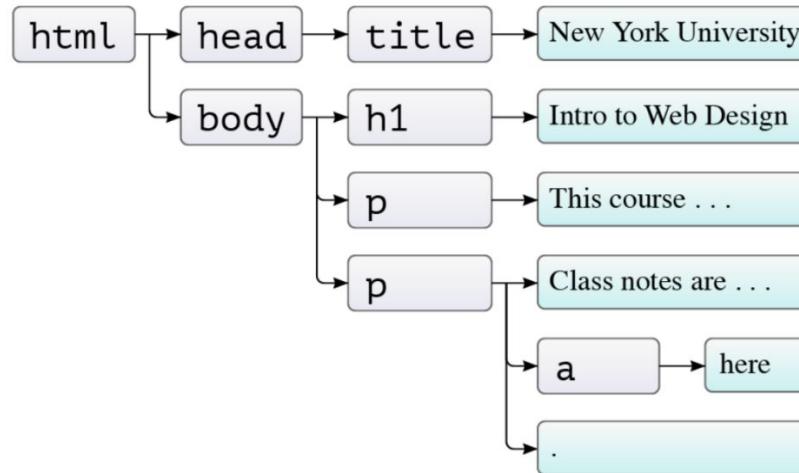
Document Object Model



Adapted from Eloquent JavaScript: The Document Object Model

Introduction to Web Design

Document Object Model



Adapted from Eloquent JavaScript: The Document Object Model

Introduction to Web Design

DOM Queries

Document Object Model

JavaScript methods that find elements in the DOM tree are called “DOM queries.”

DOM queries may return one element, or they may return a “node list.”

Which DOM query you use depends on what you want to do and the scope of browser support required.

Introduction to Web Design

Document Object Model

DOM Queries

Methods that return a single element node:

- `getElementById()`
- `querySelector()`

Introduction to Web Design

DOM Queries

Document Object Model

Methods that return one or more elements as a node list:

- `getElementsByClassName()`
- `getElementsByTagName()`
- `querySelectorAll()`

Let's change the link via the <a> element

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>New York University</title>
5  </head>
6  <body>
7      <h1>Intro to Web Design</h1>
8
9      <p>In this course you will learn how to build websites.</p>
10
11     <p>Class notes are available
12     <a href="notes.html">here</a>.</p>
13
14 <script type="text/javascript">
15
16     let link= document.querySelector('a');
17     link.textContent = "Mozilla Developers Network";
18     link.href = "https://developer.mozilla.org/en-US/"
19
20
21 </script>
22
23 </body>
24 </html>
```

Creating new elements

```
let sect= document.querySelector('section');
let para= document.createElement('p');
para.textContent = "JavaScript is fun!";
sect.appendChild(para);
```

Generate some CSS styling with JS

Use JS to create a variable selecting the ‘para.style’ element.

Change the following:

- color='red'
- backgroundColor='yellow'
- padding='10px'
- width ='250px'
- textAlign='center'

When you are done, use the element inspector in your browser to see how the code is rendered.