

DOM in JS

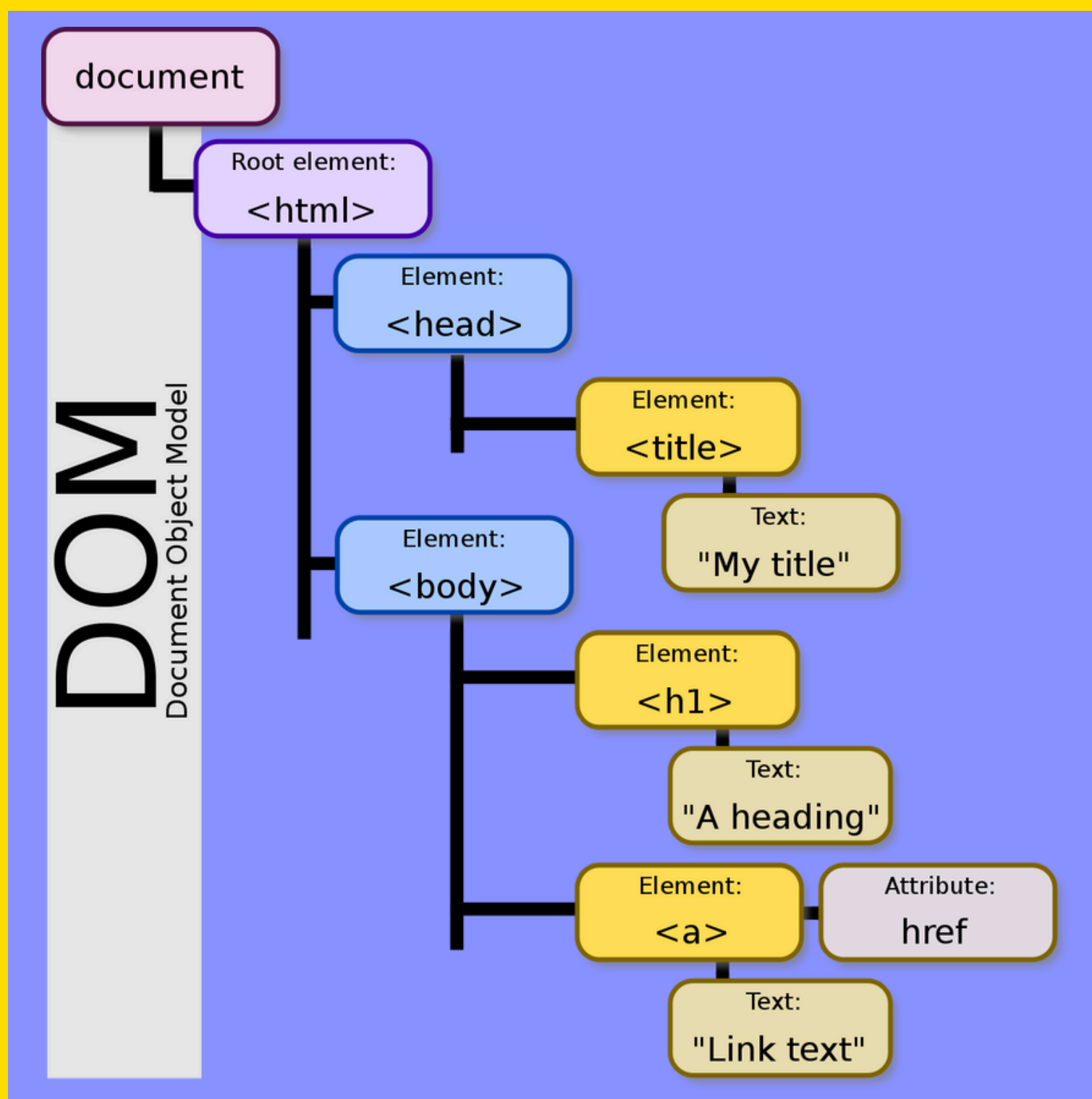


www.scribbler.live



Document Object Model

- Think of the DOM as a tree-like structure that represents the elements of a web page. It provides a structured way to access, modify, and interact with the content and structure of HTML documents.



Application of DOM

- Imagine you have a web page with various elements like headings, paragraphs, buttons, and images. The DOM represents each of these elements as nodes in a tree structure.
- You can use JavaScript to traverse this tree, find specific elements, and change their content, styles, or even add new elements dynamically.



Methods to select a HTML Element

`getElementById`

`getElementsByClassName`

`getElementsByTagName`

`querySelector`

`querySelectorAll`



getElementById

```
JS  
var elementById = document.getElementById("list");
```

```
HTML  
  
<html>  
<body>  
<h1>The Document Object</h1>  
<p>An unordered list:</p>  
<ul>  
  <li id="list">Java</li>  
  <li>JavaScript</li>  
  <li>C#</li>  
</ul>
```

- Returns the first matching element

getElementsByClassName

```
JS  
var elementsByClass = document.getElementsByClassName("language");
```

```
HTML  
  
<html>  
<body>  
<h1>The Document Object</h1>  
<p>An unordered list:</p>  
<ul>  
  <li class="language">Java</li>  
  <li class="language">JavaScript</li>  
  <li class="language">C#</li>  
</ul>
```

- Returns a collection of elements which can be accessed using [index]

getElementsByTagName querySelectorAll

- Returns a collection of elements which can be accessed using [index] similar to class name.

querySelector

- Returns the first matching element similar to Id.



Get ready!

**We're excited to announce
a contest coming soon.**



Stay tuned for more details!

Keep Exploring Javascript with us!

Share this with a friend who needs it and
make sure to practice these in scribbler.



Scribbler.live

Free and Open Interface to
experiment JavaScript