



# SNS COLLEGE OF ENGINEERING



An Autonomous Institution

## Coimbatore-107

### 19TS601-FULL STACK DEVELOPMENT

UNIT-1

JAVASCRIPT AND BASICS OF MERN STACK

### Modules - DOM tree



# Modules

- JavaScript modules allow you to break up your code into separate files.
- This makes it easier to maintain a code-base.
- Modules are imported from external files with the import statement.
- Modules also rely on `type="module"` in the `<script>` tag.



# Example

```
<!DOCTYPE html>
<html>
<body>
<h1>JavaScript Modules</h1>
<p id="demo"></p>
<script type="module">
import message from "./message.js";
document.getElementById("demo").innerHTML=
message();
</script>
</body>
</html>
```



- OUTPUT

JavaScript Modules

Jesse is 40 years old.

`message.js`

```
const message = () => {  
  const name = "Jesse";  
  const age = 40;  
  return name + ' is ' + age + 'years old.';  
};  
export default message;
```



# Export

- Modules with functions or variables can be stored in any external file.
- There are two types of exports:
  - Named Exports and
  - Default Exports.

## Named Exports

- Let us create a file named person.js, and fill it with the things we want to export.
- You can create named exports two ways.
- In-line individually, or all at once at the bottom.



## In-line individually:

`person.js`

```
export const name = "Jesse";  
export const age = 40;
```

## All at once at the bottom:

`person.js`

```
const name = "Jesse";  
const age = 40;  
export {name, age};
```



# Default Exports

- Let us create another file, named message.js, and use it for demonstrating default export.
- You can only have one default export in a file.

message.js

```
const message = () => {  
  const name = "Jesse";  
  const age = 40;  
  return name + ' is ' + age + 'years old.';  
};  
export default message;
```



# Import

- You can import modules into a file in two ways, based on if they are named exports or default exports.
- Named exports are constructed using curly braces. Default exports are not.

## Import from named exports

import named exports from the file person.js:

```
import { name, age } from "./person.js";
```





## Import from default exports

import a default export from the file message.js:

```
import message from "./message.js";
```

### NOTE:

- Modules only work with the HTTP(s) protocol.
- A web-page opened via the file:// protocol cannot use import / export.



# What is the DOM?

- The DOM is a W3C (World Wide Web Consortium) standard.
- The DOM defines a standard for accessing documents
- *"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."*



# What is the DOM?

- The W3C DOM standard is separated into 3 different parts:
- **Core DOM** - standard model for all document types
- **XML DOM** - standard model for XML documents
- **HTML DOM** - standard model for HTML documents



# What is Html DOM?

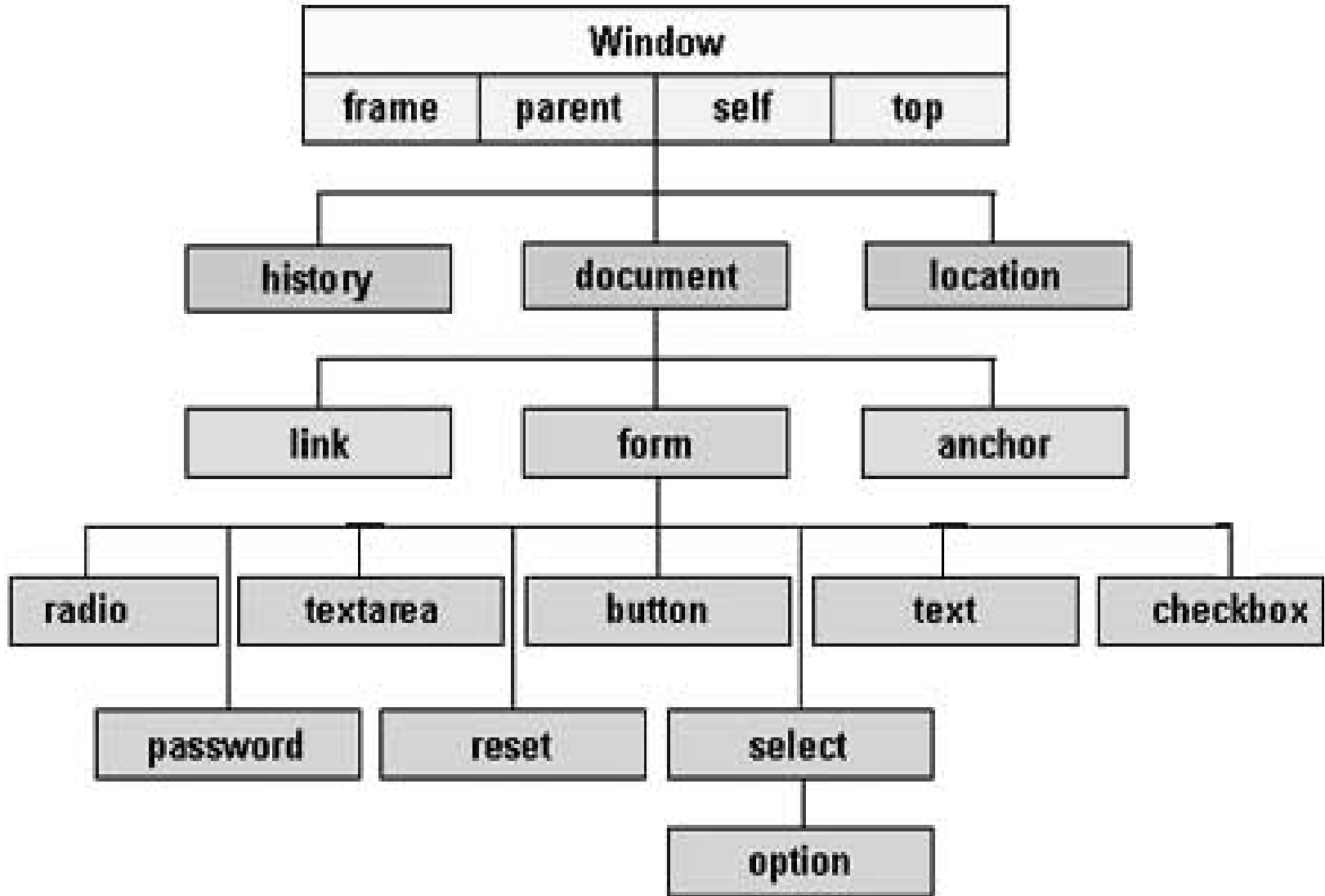
- The HTML DOM is a standard for how to get, change, add, or delete HTML elements.
- The HTML DOM is a standard object model and programming interface for HTML. It defines:
  - The HTML elements as objects
  - The properties of all HTML elements
  - The methods to access all HTML elements
  - The events for all HTML elements



- Every web page resides inside a browser window which can be considered as an object.
- A Document object represents the HTML document that is displayed in that window.
- The Document object has various properties that refer to other objects which allow access to and modification of document content.
- The way a document content is accessed and modified is called the **Document Object Model**, or **DOM**.
- The Objects are organized in a hierarchy. This hierarchical structure applies to the organization of objects in a Web document.



- **Window object** – Top of the hierarchy. It is the outmost element of the object hierarchy.
- **Document object** – Each HTML document that gets loaded into a window becomes a document object. The document contains the contents of the page.
- **Form object** – Everything enclosed in the `<form>...</form>` tags sets the form object.
- **Form control elements** – The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.





# DOM Tree

- The DOM is often referred to as the DOM tree, and consists of a tree of objects called nodes.
- The backbone of an HTML document is tags.
- According to the Document Object Model (DOM), every HTML tag is an object.
- Nested tags are “children” of the enclosing one. The text inside a tag is an object as well.





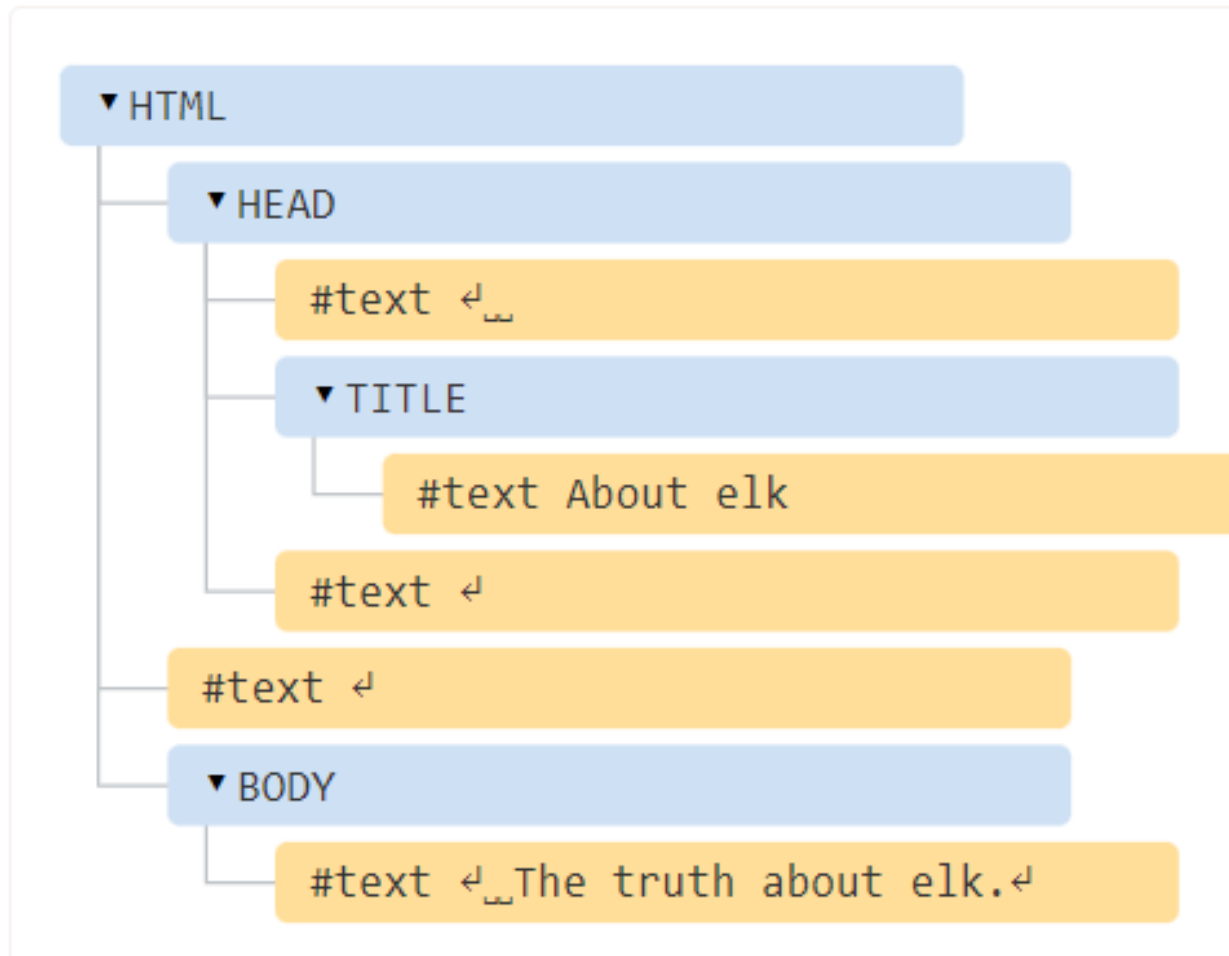
# DOM Tree

```
<!DOCTYPE HTML>
<html>
<head>
  <title>About elk</title>
</head>
<body>
  The truth about elk.
</body>
</html>
```



# DOM Tree

The DOM represents HTML as a tree structure of tags.





- All objects are accessible using JavaScript, and we can use them to modify the page.

```
document.body.style.background = 'red';
```

```
// make the background red
```

```
setTimeout(() => document.body.style.background = "", 3000); // return back
```

- For example, `document.body` is the object representing the `<body>` tag.
- Running this code will make the `<body>` red for 3 seconds:

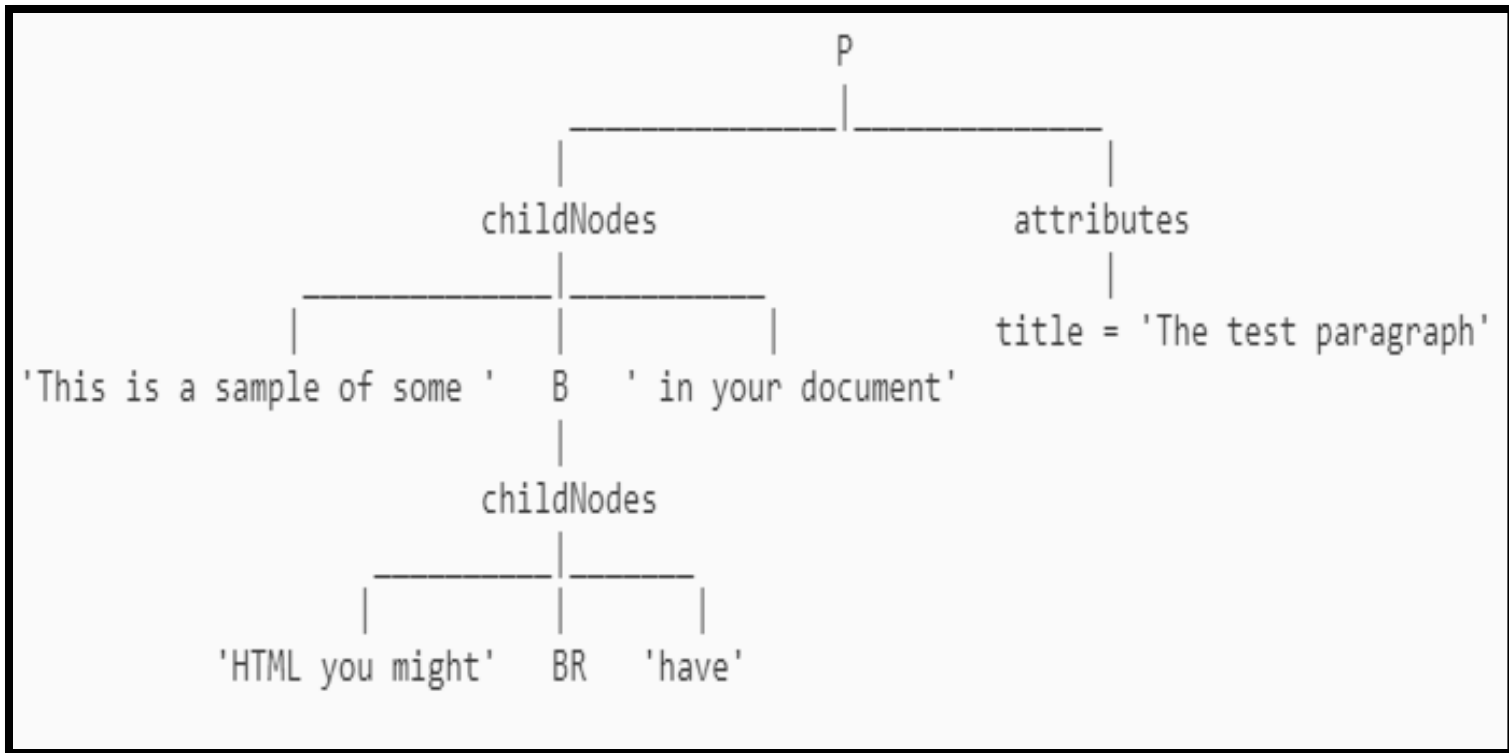


# Example

- `<p title="The test paragraph">This is a sample of some HTML you might  
have</b>` in your document`</p>`
- **OUTPUT**
- This is a sample of some **HTML you might have** in your document



- The DOM tree views this (simplified) as follows:





- The 'P', 'B' and 'BR' nodes are element nodes
- childNodes and attributes are collections
- the title='The test paragraph' pair is an attribute node
- the text strings are text nodes.



# Thank You