Below is a **simplified, well-structured, and internally explained version** of your content on **JavaScript DOM Document Object**, written in the **same depth and format style** you've been following, but **cleaner, exam-oriented, and easier to remember**.

📘 JavaScript DOM Document Object

The **DOM `document` object** is the **root object of the HTML DOM**.
It represents the **entire web page** loaded inside the browser.

When a web page loads:

- The browser creates a **DOM tree**
- The **document object** becomes the **entry point** to access that tree
- All HTML elements become **children of `document`**

👉 **Every DOM operation in JavaScript starts with the `document` object.**

🔹 What does the Document Object do?

The `document` object allows JavaScript to:

- Access HTML elements
- Modify page structure
- Change content
- Update styles
- Attach events
- Create or remove elements

In short:
HTML builds structure → DOM represents it → JavaScript controls it using `document`.

🌍 Relationship Between `window` and `document`

The `document` object is a **property of the `window` object**.

```js
window.document   // valid
document          // also valid (window is implicit)
```

📌 Since `window` is the global object, you usually write just `document`.

🧠 DOM Hierarchy (Conceptual)

```css
```

```
window
 └── document
      └── html
           ├── head
           └── body
```

So, when you write:

```js
document.getElementById("box")
```

You are asking the **document** to find an element inside this tree.

🔹 JavaScript DOM Document Methods

These methods belong directly to the **document object**.

📋 Commonly Used Document Methods

1️⃣ `document.getElementById()`

Returns **one element** with the given ID.

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

✔️ Most frequently used

✔️ Fast

✔️ Returns `null` if not found

2️⃣ `document.getElementsByClassName()`

Returns a **live HTMLCollection** of elements.

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

3️⃣ `document.getElementsByTagName()`

Selects elements by tag name.

```js
```

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

4️⃣ `document.getElementsByName()`

Returns elements having a specific `name` attribute.

5️⃣ `document.createElement()`

Creates a new HTML element (not added to DOM yet).

```js
document.createElement("div");
```

6️⃣ `document.createTextNode()`

Creates text that can be attached to elements.

7️⃣ `document.createComment()`

Creates an HTML comment node.

8️⃣ `document.createDocumentFragment()`

Creates a lightweight container in memory

✔️ Improves performance when inserting many elements

9️⃣ `document.addEventListener()`

Adds an event to the document.

```js
document.addEventListener("click", handler);
```

🔟 `document.write()` / `document.writeln()`

Writes directly to the page (⚠️ replaces entire page).

📌 **Deprecated in modern apps**, but important for exams.

1️⃣1️⃣ `document.open()` / `document.close()`

Used with `document.write()` to control output stream.

1️⃣2️⃣ `document.normalize()` / `normalizeDocument()`

- Removes empty text nodes
- Merges adjacent text nodes

1️⃣3️⃣ `document.adoptNode()`

Moves a node from another document into the current one.

🔹 JavaScript DOM Document Properties

Properties give **information about the document** or allow modification.

📄 Basic Document Properties

Property	Description
`document.URL`	Full page URL
`document.title`	Page title
`document.domain`	Domain name
`document.baseURI`	Base URI
`document.lastModified`	Last modification time

🧱 Structural Properties

Property	Meaning
`document.documentElement`	`<html>` element
`document.head`	`<head>`
`document.body`	`<body>`
`document.doctype`	Document type

📦 Collection Properties

Property	Returns
`document.forms`	All forms
`document.images`	All images
`document.links`	All `<a>` with href
`document.scripts`	All `<script>`
`document.embeds`	Embedded elements

✏️ Editing & State Properties

Property	Purpose
`document.designMode`	Editable document
`document.hasFocus()`	Focus check
`document.cookie`	Read/write cookies
`document.charset`	Character encoding
`document.characterSet`	Encoding

⚠️ Deprecated / Browser-Specific Properties

- `domConfig` (deprecated)
- `documentMode` (IE only)
- `applets` (deprecated)
- `strictErrorChecking` (rarely used)

📌 Mention in exams, avoid in projects.

🔹 Special Document Properties (With Examples)

✅ `document.childElementCount`

Returns **number of direct child elements of document**.

📌 A document usually has **only one child** → `<html>`

```js
document.childElementCount;
```

Why output is 1?

```css
document
 └── html   ← only direct child
```

✅ `document.links`

Returns a **collection of all anchor tags (`<a>`) with href**.

```js
```

```
document.links;
```

✔️ Can be looped using `for...of`

✅ `document.title`

Gets or sets the page title.

```js
document.title = "New Title";
```

🧠 Internal Understanding (Exam Tip)

- `document` is an **object**
- It owns **methods + properties**
- All elements are **descendants of document**
- JavaScript cannot touch HTML **without DOM**
- DOM = **Bridge between HTML & JavaScript**

📌 Final One-Line Summary (Very Important)

> **The DOM `document` object represents the entire web page and acts as the root through which JavaScript accesses and manipulates all HTML elements.**

If you want next, I can give you:

- 🔹 **document vs window (deep internal difference)**
- 🔹 **DOM traversal diagram**
- 🔹 **Most important document methods for exams**
- 🔹 **MCQs + viva questions**
- 🔹 **Real-world DOM mini examples**

Just say **next** 👍