Below is a **clean, simplified, exam-ready, and internally deep** explanation of **"Changing CSS with JavaScript"**, written in the **same format, depth, and teaching style** you asked for.

This version is suitable for:

- 📘 College exams
- 🧠 Viva / interviews
- 💻 Real-world frontend understanding

🎨 Changing CSS with JavaScript

JavaScript allows us to **dynamically change the CSS styles of HTML elements** at runtime.

This means styles can change **after the page has loaded**, based on user actions, events, or logic.

🧠 Internal Concept (Very Important)

When an HTML page loads:

1. Browser parses HTML
2. Creates a **DOM tree**
3. Each HTML element becomes a **JavaScript object**
4. Every element object contains a `style` **object**

👉 This `style` object holds **all inline CSS properties** for that element.

So internally:

```
js

element.style
```

is an object that directly controls the element's CSS.

🔹 Syntax to Change CSS Using JavaScript

```
js

element.style.property = value;
```

Explanation

- `element` → HTML element from DOM
- `property` → CSS property (written in **camelCase**)
- `value` → CSS value as a string

Example Conversion

CSS	JavaScript
background-color	backgroundColor
font-size	fontSize
border-radius	borderRadius

🔹 Example: Changing Style of an Element

```
html

<div id="square">Changing the color of this Div.</div>
<button onclick="changeColor()">Change Color</button>

<script>
function changeColor() {
  let square = document.getElementById("square");
  square.style.backgroundColor = "red";
}
</script>
```

What Happens Internally?

- JavaScript finds the DOM node
- Updates its `style.backgroundColor`
- Browser repaints only that element

🔹 Changing Style When an Event Triggers

JavaScript is commonly used to **change CSS in response to events** like:

- click
- mouseover
- keypress
- scroll

Example: Change Style on Click

```
html

<div id="square">Click Me</div>

<script>
const square = document.getElementById("square");
```

```
square.addEventListener("click", () => {
  square.style.backgroundColor = "green";
  square.style.fontSize = "25px";
});
</script>
```

📌 **Key Point**

- Event → JavaScript → CSS update
- This is how interactive UIs work

🔹 Dynamically Changing CSS Using Variables

Styles can be changed **dynamically using variables**, not just fixed values.

Example: Changing Color Using Radio Buttons

```
html

<p id="square">Select a color</p>

Yellow <input type="radio" name="color" id="yellow">
Green <input type="radio" name="color" id="green">
Red <input type="radio" name="color" id="red">

<script>
let square = document.getElementById("square");
let colors = document.getElementsByName("color");

for (let i = 0; i < colors.length; i++) {
  colors[i].addEventListener("change", function () {
    square.style.backgroundColor = this.id;
  });
}
</script>
```

Internal Logic

- `this.id` returns selected color
- Assigned dynamically to `backgroundColor`

🔹 Inline Style vs CSS Class (Important Concept)

Inline Style (via JavaScript)

```
js
```

```
element.style.color = "red";
```

CSS Class (Better Practice)

```
js

element.classList.add("active");
```

📌 Best Practice

- Use `style` for **small, dynamic changes**
- Use `classList` for **large or reusable styles**

🔹 Why JavaScript Style Changes Are Powerful

JavaScript-based CSS changes enable:

- Animations
- Theme switching (dark / light mode)
- Responsive UI
- Real-time feedback
- Interactive dashboards

⚠️ Important Rules & Mistakes (Exam + Viva)

❌ CSS property names with hyphens
✔️ Use camelCase

❌ Numbers without units
✔️ `"20px"` instead of `20`

❌ Expecting `style` to read external CSS
✔️ `style` reads only inline styles

🔹 Commonly Used `style` Properties (Must Know)

Property	Use
color	Text color
backgroundColor	Background color
fontSize	Text size
display	Show / hide elements

Property	Use
width / height	Element size
border	Border styling
opacity	Transparency
visibility	Visible / hidden

🧪 Example: Show / Hide Element

```js
element.style.display = "none";   // hide
element.style.display = "block";  // show
```

📌 One-Line Exam Answer

> **JavaScript can dynamically change CSS by modifying the `style` object of DOM elements.**

✅ Final Summary

- Every HTML element has a `style` object
- JavaScript modifies inline CSS
- CSS properties use camelCase
- Events + styles = interactivity
- Prefer classes for complex styling

If you want, next we can cover:

- 🔥 `classList` vs `style`
- 🔥 Dark mode implementation
- 🔥 CSS animations using JS
- 🔥 DOM + CSS interview MCQs
- 🔥 Mini project: Theme switcher

Just say **next** 👍