1. Event Basics

- **Event:** Action in the browser (click, keydown, input, etc.).
- **Event Listener:** Function that runs when an event occurs.

```
element.addEventListener("click", handler);
element.removeEventListener("click", handler);
```

---

2. Event Flow

1. **Capturing phase:** Event travels from `window → document → parent → child`.
2. **Target phase:** Event reaches the actual element.
3. **Bubbling phase:** Event bubbles back up to parent elements.

⚡ Default is **bubbling** (`false` as 3rd arg).
⚡ Use `{ capture: true }` for capturing.

```
document.addEventListener(type, listener, { capture: true });
```

PHASES: CAPTURE → AT TARGET → BUBBLING

---

3. Common Events

- **Mouse:** click, dblclick, mousemove, mouseenter, mouseleave
- **Keyboard:** keydown, keyup, keypress
- **Form:** input, change, submit, focus, blur
- **Window:** load, resize, scroll

---

4. Event Object (`e`)

- `e.target` → Element that triggered the event
- `e.currentTarget` → Element where listener is attached
- `e.preventDefault()` → Stop default behavior (e.g., anchor navigation)
- `e.stopPropagation()` → Stop bubbling/capturing

---

5. Event Delegation

- Attach **one listener** to parent instead of multiple child elements.
- Use `e.target` to check which child was clicked.

```
document.getElementById("list").addEventListener("click", (e) => {
  if (e.target.tagName === "LI") {
    console.log("Clicked:", e.target.textContent);
  }
});
```

---

6. DOM Manipulation

Create Element:

```
let div = document.createElement("div");
div.textContent = "Hello";
```

Insert:

```
parent.appendChild(div);
parent.insertBefore(div, referenceNode);
```

Remove:

```
element.remove();
```

```
parent.removeChild(child);
```

Modify:

```
element.textContent = "New text";
element.style.color = "red";
element.setAttribute("class", "active");
```

---

7. Example Codes

- **CODE1 – Log click anywhere:**

```
document.addEventListener('click', function() {
    console.log("Clicked");
});
```

- **CODE2 – Change background of an element on click:**

```
let content = document.querySelector(".grid");
content.addEventListener('click', function() {
    content.style.background = 'red';
});
```

- **CODE3 – Change body and h1 styles on click:**

```
document.addEventListener('click', function() {
    document.body.style.background = 'black';
    document.body.style.color = 'white';
    document.querySelector('h1').style.color = 'green';
});
```

---

8. Equality in JS

- `==` → Loose equality (`1 == '1'` is true)

- `===` → Strict equality (`1 === '1'` is false)

---

9. Important Notes

- For `removeEventListener`, pass the **same function reference** used in `addEventListener`.
- Anonymous functions cannot be removed directly; use named functions instead.

---

10. Anchor Tag – Prevent Default

```
let links = document.querySelectorAll("a");
let thirdLink = links[2]; // GitHub link
thirdLink.addEventListener("click", function(event) {
    event.preventDefault(); // block navigation
    console.log("Default action prevented for GitHub link!");
});
```

---

11. Avoid Too Many Events – Dynamic Paragraphs

```
let DIV = document.createElement('div');

function para(event) {
    console.log("Clicked on " + event.target.textContent);
}

for (let i = 1; i <= 100; i++) {
    let newElement = document.createElement('p');
    newElement.textContent = 'This is paragraph ' + i;
    newElement.addEventListener('click', para);
    DIV.appendChild(newElement);
}

document.body.appendChild(DIV);
```

---

12. Event Delegation with Spans

```
let element = document.querySelector('#wrapper');

element.addEventListener('click', function(event) {
    // Use nodeName to avoid clicks on parent elements
    if (event.target.nodeName === 'SPAN') {
        console.log("Clicked on span: " + event.target.textContent);
    }
});
```