

## Lecture 23

18/02/2023

### \* Browser Events

Browser event are called browser

### \* Events

(i) Monitor Events: see all events

(ii) → monitorEvents(document)

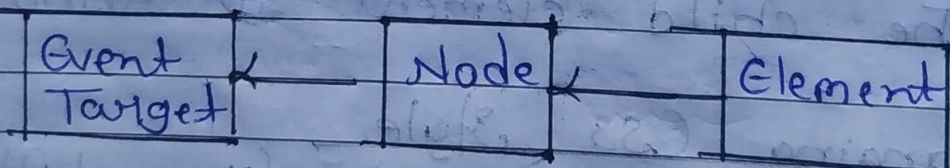
→ unmonitorEvents(document)

→ not see any events

### \* Event Listener

(i) Event Target - <sup>Top level entity</sup>

It is a interface implemented by object that can receive events & may have listeners for them.



(a) addEventListener()

(b) removeEventListener()

(c) dispatchEvent()

### \* By addEventListener() -

(i) Listen to events

(ii) Responds to events

(iii) Hook into events



## \* Pseudocode for apply Event

<event - target>. addEventListener (   
 <event - to - listen - for>, <function   
 - to - run - when - event -   
 happened > );

Needed

- event - target → Component ( document,   
 P, h1, - etc )
- event - type → ( click, dblclick - etc )
- function → define what to do when   
 event happen.

## \* removeEventListener

pass exact same function which is   
 added in addEventListener for   
 remove event.

### Conditions

- (i) Same target
  - (ii) Same type
  - (iii) Same function
- Ex → document.removeEventListener ( type,   
 function );

## \* Phases of an Event

### (i) Capturing Phase

Capture element on which event   
 added.

### (ii) At target Phase

event function run on target item

### (iii) Bubbling Phase

Return event from target to root.



- ★ By default event execute in bubbling phase but if we have to run it on capturing phase give true -

Ex

element.addEventListener('click',

{ type: listener, true })

↑

func()

↑

Click  
dblclick  
etc

What should  
happen after  
Event ~~action~~  
~~event~~ trigger

use capture

(true → capturing phase)

(false → bubbling phase)

## ★ The Event Object

When an event occurs at that time addEventListener function receive event object which have lot of information.

document.addEventListener('click', function(e) {

console.log(e); })

all info  
about event

## ★ The Default Action

all element have some default action we can prevent it by preventDefault()

Suppose → anchor tag

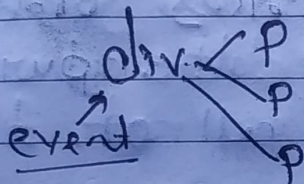
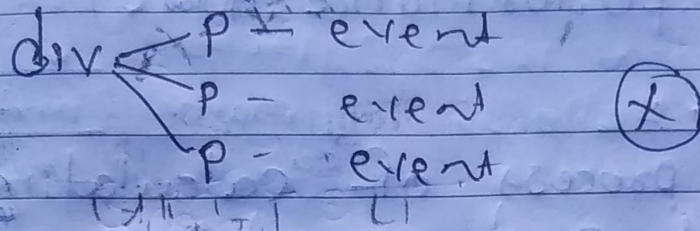
When clicked link is open

→ a.preventDefault()

→ new link not open



\* Avoid too many events instead of giving individual element give parent element event.



\* nodeName

→ give node name.

`<div id="wrapper">`

`<p> <span> </span> </p>`

`</div>`

we have to give event only clicked on span.

Let wrapper = document.getElementById('wrapper')

```

wrapper.addEventListener('click', function(e) {
    if (e.target.nodeName === "SPAN") {
        console.log("span clicked");
    }
});
    
```