12/03/24

Event delegation in Javascript

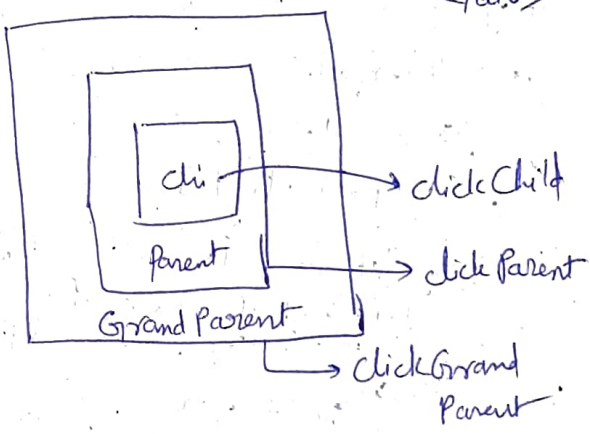
Event. Bubbling | Capturing in JS

Intro:

Event Bubbling and Event Capturing (also known as Event trickling) represents the sequence of events under nested HTML. Let's take a sample nested html as below.

Sample Code:

```
<div id="grand parent" (on Click) = "click grandParent()"></div>
    <div id = "parent" (onClick) = "click parent()">
        <div id = "child" (onClick) = "clickChild()">
        </div>
    </div>
```



Explanation:

Whenever there are seperate click functions to be executed on each div of nested divs.

In case of Event Bubbling,

a) When we click on child,

sequence of methods called:

clickChild (1) → clickParent (2) → clickGrandParent (3)

Assume as a Bubble which is always going outside

b) when we click on Parent

Click Parent → Click Grand Parent
① ②

In case of Event Capturing / Trickling

a) When we click on child
sequence of methods called

Click Grand Parent → Click Parent → Click Child
① ② ③.

b) When we click on Parent

Click Grand Parent → Click Parent
① ②

Assume as a Trickle going down.

In back days, Netscape suggested Event Capturing, but microsoft suggested Event Bubbling, in which W3C (standard) committee makes it as a choise for developer

Add Event Listener

In add event listener (which generally has two arguments 'eventName', () => { }), we will be having third argument saying useCapture: true/false, if use capture is true, event will be captured else event will be bubbled

By default it will take useCapture as false when third argument is not mentioned / mentioned as false

Tricky question

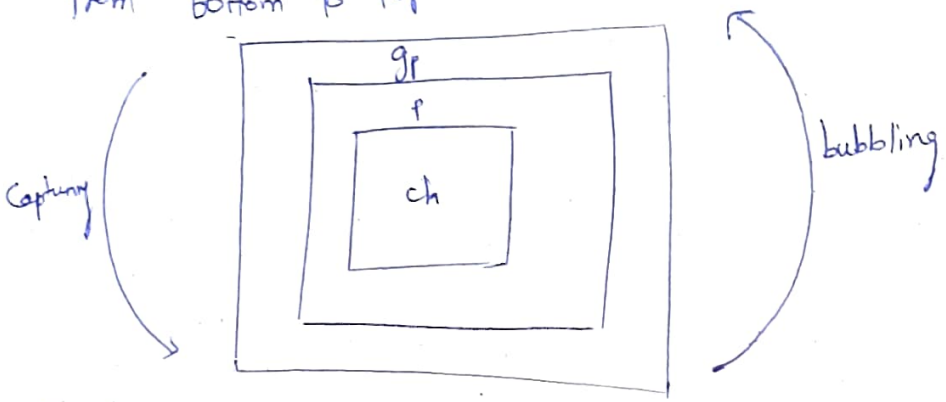
What if we give different use Capture flags to different divs

Eg: child has true, parent has false, grandParent has true

// Observe the output

Performance Issues

According to W3C, tickling down happens first from top to bottom hierarchy and then bubbling happens from bottom to top



Event propagation is a whole cycle of two halves, whenever a div/element is given capture flag as true they are executed in capture cycle and then bubbling phase arrives. If we give a mix of true/false capture flags, Then events which have capture: true, executes first

Eg:

div #grandParent	→ capture: true	o/p when clicked on child
div # parent	→ capture: false	gp clicked
div # child	→ capture: true	ch clicked
		p clicked

Eg2

div #grandParent	→ capture: false	o/p when clicked on child
div # parent	→ capture: false	child, parent, gp
div # child	→ capture: true	

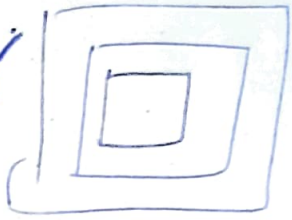
How we can stop propagation?

Event propagation generally refers to the breakage of event cycle (capturing + bubbling)

It furthers stops to propagate (all cases child clicked)

Eg:

```
d.qs("gp").aEL('cl', (e) => {
    e.stopPropagation();
}, true);
```



o/p gp clicked

Eg2

```
d.qs("gp").aEL('cl', () => { }, false);
d.qs("p").aEL('cl', () => { }, false);
d.qs("ch").aEL('cl', (e) => { }, true);
```

o/p e.stopPropagation()

ch clicked

Eg3

```
d.qs("gp").aEL('cl', () => { }, false);
d.qs("p").aEL('cl', (e) => {e.stopPropagation()}, true);
d.qs("ch").aEL('cl', () => { }, false);
```

o/p

p clicked.