Tutorial : Event Capturing and Bubbling

Capturing, at target, and bubbling phases

Read the Explanation ,then run the html files uploaded at the bottom to get clarity.

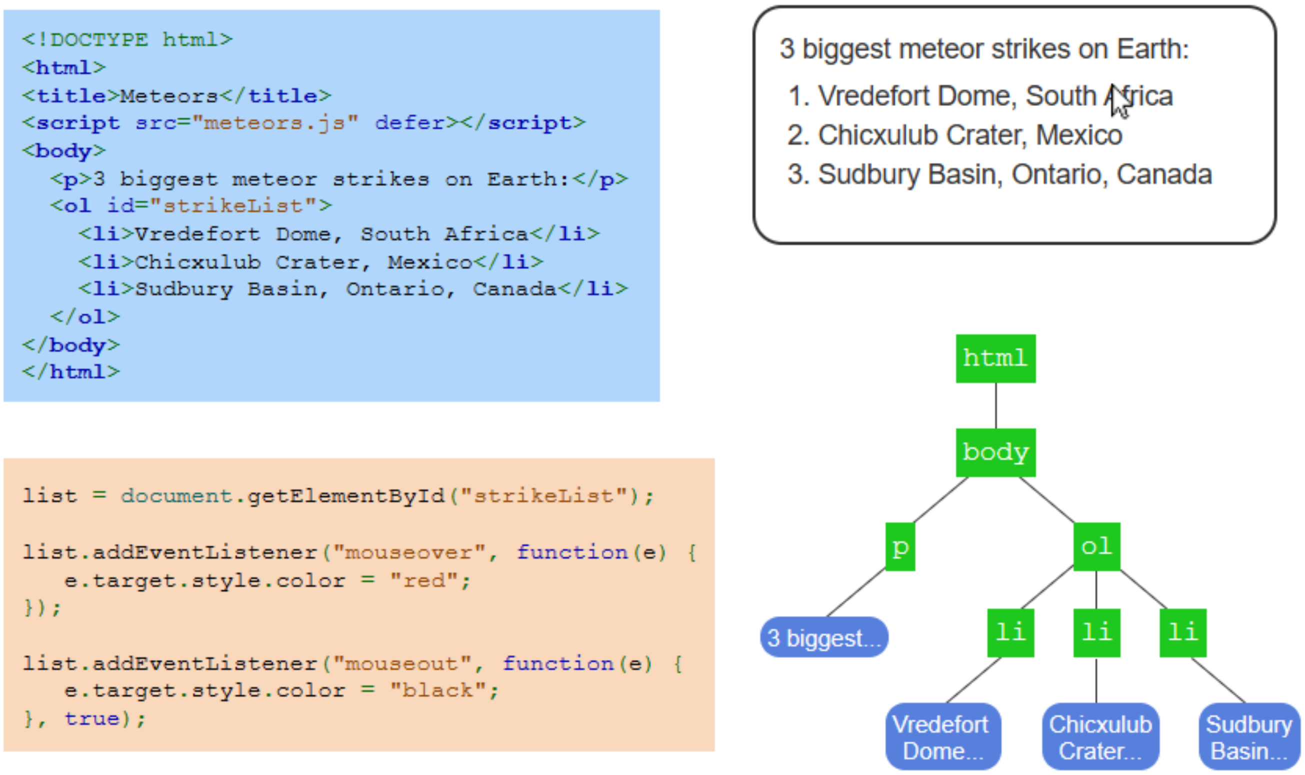
When an event occurs, the browser follows a simple DOM traversal process to determine which handlers are relevant and need to be called. This traversal process follows three phases: capturing, at target, and bubbling.

1. In the event capturing phase, the browser traverses the DOM tree from the root to the event target node, at each node calling any event-specific handlers that were explicitly registered for activation during the capturing phase.
2. In the at target phase, the browser calls all event-specific handlers registered on the target node.
3. In the event bubbling phase, the browser traverses the DOM tree from the event target node back to the root node, at each node calling all event-specific handlers registered for the bubbling phase on the current node.

The optional third parameter for the `addEventListener()` method indicates whether the handler is registered for the capturing phase or bubbling phase. If the third parameter is `false` or not specified, or if the event handler is registered using any other mechanism, the browser registers the handler for the event bubbling phase. If the parameter is `true`, the browser registers the handler for the capturing phase.

Some events do not bubble, such as blur, focus, and change. When a non-bubbling event occurs, the browser will follow the event capturing phase, the at target phase, and then stop.

1. User moves the mouse cursor over item one in the list. A mouseover event occurs with the first li node as the target node.



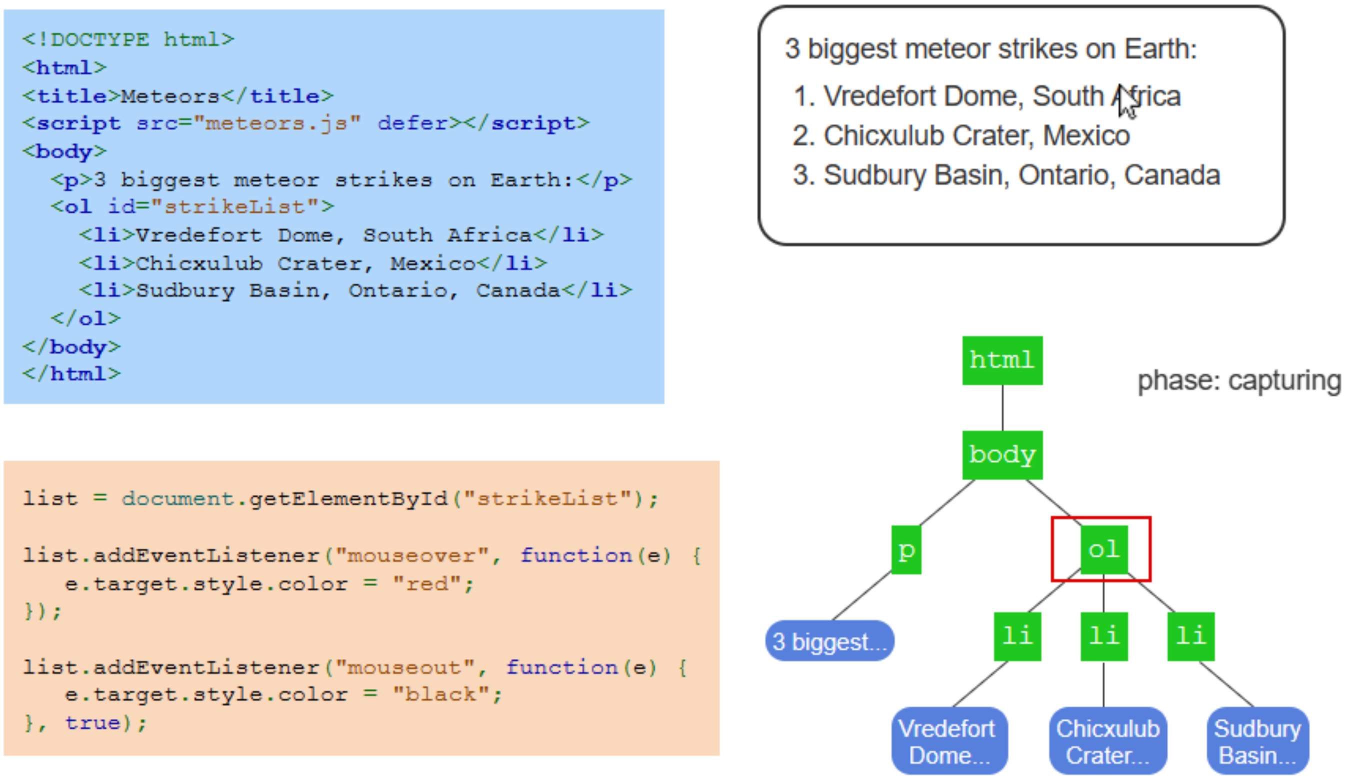
```
<!DOCTYPE html>
<html>
<title>Meteors</title>
<script src="meteors.js" defer></script>
<body>
  <p>3 biggest meteor strikes on Earth:</p>
  <ol id="strikeList">
    <li>Vredefort Dome, South Africa</li>
    <li>Chicxulub Crater, Mexico</li>
    <li>Sudbury Basin, Ontario, Canada</li>
  </ol>
</body>
</html>
```

```
list = document.getElementById("strikeList");

list.addEventListener("mouseover", function(e) {
   e.target.style.color = "red";
});

list.addEventListener("mouseout", function(e) {
   e.target.style.color = "black";
}, true);
```

3 biggest meteor strikes on Earth:

1. Vredefort Dome, South Africa
2. Chicxulub Crater, Mexico
3. Sudbury Basin, Ontario, Canada

2. Event capturing phase traverses the DOM tree from the root to event target node. No capturing handlers are registered for the mouseover event.



```
<!DOCTYPE html>
<html>
<title>Meteors</title>
<script src="meteors.js" defer></script>
<body>
  <p>3 biggest meteor strikes on Earth:</p>
  <ol id="strikeList">
    <li>Vredefort Dome, South Africa</li>
    <li>Chicxulub Crater, Mexico</li>
    <li>Sudbury Basin, Ontario, Canada</li>
  </ol>
</body>
</html>
```

```
list = document.getElementById("strikeList");

list.addEventListener("mouseover", function(e) {
   e.target.style.color = "red";
});

list.addEventListener("mouseout", function(e) {
   e.target.style.color = "black";
}, true);
```

3 biggest meteor strikes on Earth:

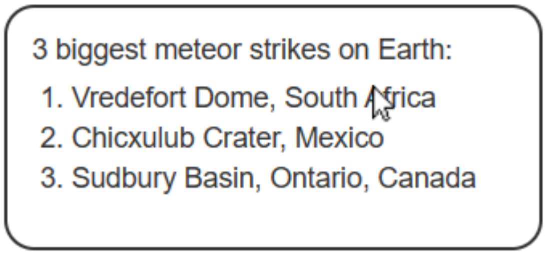
1. Vredefort Dome, South Africa
2. Chicxulub Crater, Mexico
3. Sudbury Basin, Ontario, Canada

3. At target phase looks for mouseover event handlers registered on the target node, but no

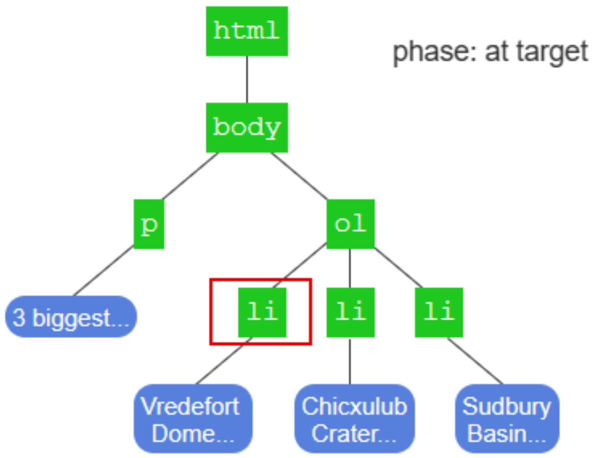
handlers are registered for the mouseover event.

```
<!DOCTYPE html>
<html>
<title>Meteors</title>
<script src="meteors.js" defer></script>
<body>
  <p>3 biggest meteor strikes on Earth:</p>
  <ol id="strikeList">
    <li>Vredefort Dome, South Africa</li>
    <li>Chicxulub Crater, Mexico</li>
    <li>Sudbury Basin, Ontario, Canada</li>
  </ol>
</body>
</html>
```

```
list = document.getElementById("strikeList");

list.addEventListener("mouseover", function(e) {
   e.target.style.color = "red";
});

list.addEventListener("mouseout", function(e) {
   e.target.style.color = "black";
}, true);
```


3 biggest meteor strikes on Earth:

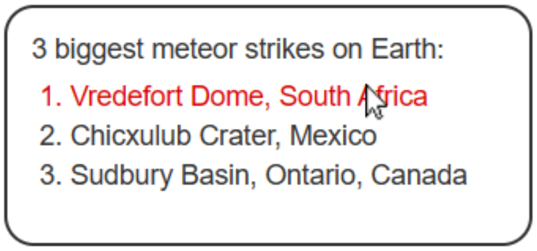
1. Vredefort Dome, South Africa
2. Chicxulub Crater, Mexico
3. Sudbury Basin, Ontario, Canada



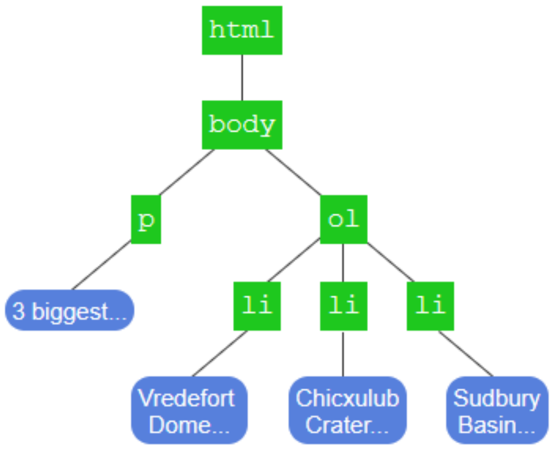
4. Event bubbling traverses DOM tree from the event node back to the root node. The ol node's bubbling event handler is called and changes the item's text to red.

```
<!DOCTYPE html>
<html>
<title>Meteors</title>
<script src="meteors.js" defer></script>
<body>
  <p>3 biggest meteor strikes on Earth:</p>
  <ol id="strikeList">
    <li>Vredefort Dome, South Africa</li>
    <li>Chicxulub Crater, Mexico</li>
    <li>Sudbury Basin, Ontario, Canada</li>
  </ol>
</body>
</html>
```

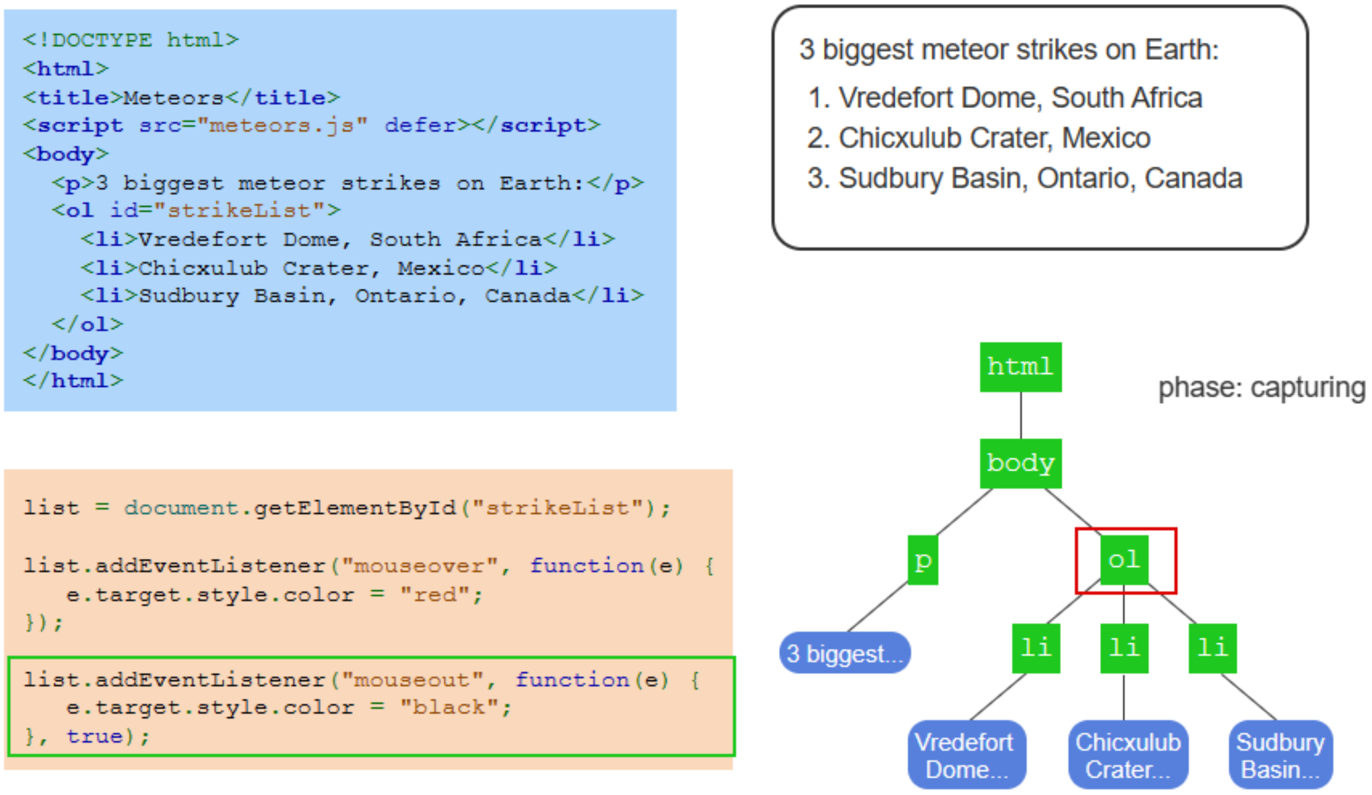
```
list = document.getElementById("strikeList");

list.addEventListener("mouseover", function(e) {
   e.target.style.color = "red";
});

list.addEventListener("mouseout", function(e) {
   e.target.style.color = "black";
}, true);
```


3 biggest meteor strikes on Earth:

1. Vredefort Dome, South Africa
2. Chicxulub Crater, Mexico
3. Sudbury Basin, Ontario, Canada



5. A mouseout event occurs with the first li node as the target node. Event capturing phase traverses the DOM tree from the root to event target node. The mouseout event handler turns the list item black.



```
<!DOCTYPE html>
<html>
<title>Meteors</title>
<script src="meteors.js" defer></script>
<body>
  <p>3 biggest meteor strikes on Earth:</p>
  <ol id="strikeList">
    <li>Vredefort Dome, South Africa</li>
    <li>Chicxulub Crater, Mexico</li>
    <li>Sudbury Basin, Ontario, Canada</li>
  </ol>
</body>
</html>
```

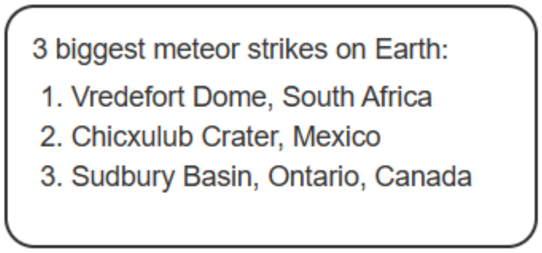
```
list = document.getElementById("strikeList");

list.addEventListener("mouseover", function(e) {
   e.target.style.color = "red";
});

list.addEventListener("mouseout", function(e) {
   e.target.style.color = "black";
}, true);
```

3 biggest meteor strikes on Earth:

1. Vredefort Dome, South Africa
2. Chicxulub Crater, Mexico
3. Sudbury Basin, Ontario, Canada

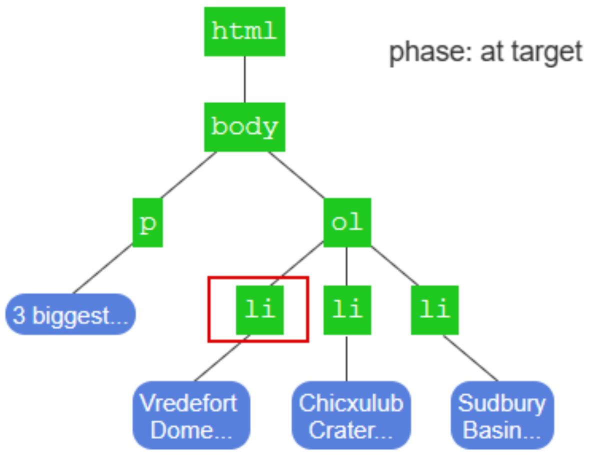
6. The at target phase looks for relevant mouseout event handlers registered on the target node, but no handlers are registered for the mouseout event.

```html
<!DOCTYPE html>
<html>
<title>Meteors</title>
<script src="meteors.js" defer></script>
<body>
  <p>3 biggest meteor strikes on Earth:</p>
  <ol id="strikeList">
    <li>Vredefort Dome, South Africa</li>
    <li>Chicxulub Crater, Mexico</li>
    <li>Sudbury Basin, Ontario, Canada</li>
  </ol>
</body>
</html>
```


3 biggest meteor strikes on Earth:

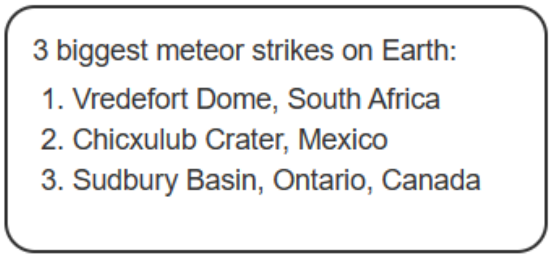
1. Vredefort Dome, South Africa
2. Chicxulub Crater, Mexico
3. Sudbury Basin, Ontario, Canada

```javascript
list = document.getElementById("strikeList");

list.addEventListener("mouseover", function(e) {
   e.target.style.color = "red";
});

list.addEventListener("mouseout", function(e) {
   e.target.style.color = "black";
}, true);
```



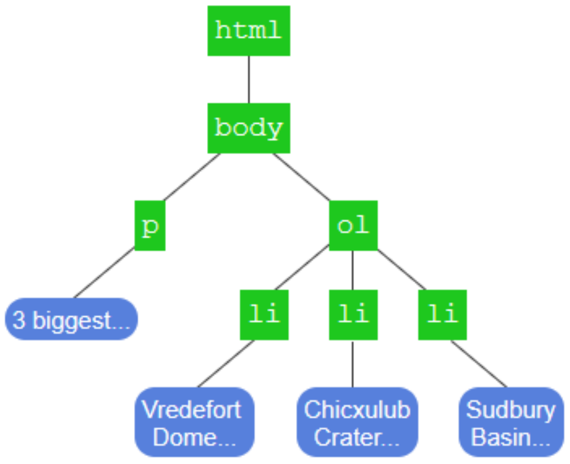
7. The event bubbling phase looks for any relevant mouseout event handlers by moving up to the DOM tree, but no elements have mouseout event handlers registered.

```html
<!DOCTYPE html>
<html>
<title>Meteors</title>
<script src="meteors.js" defer></script>
<body>
  <p>3 biggest meteor strikes on Earth:</p>
  <ol id="strikeList">
    <li>Vredefort Dome, South Africa</li>
    <li>Chicxulub Crater, Mexico</li>
    <li>Sudbury Basin, Ontario, Canada</li>
  </ol>
</body>
</html>
```


3 biggest meteor strikes on Earth:

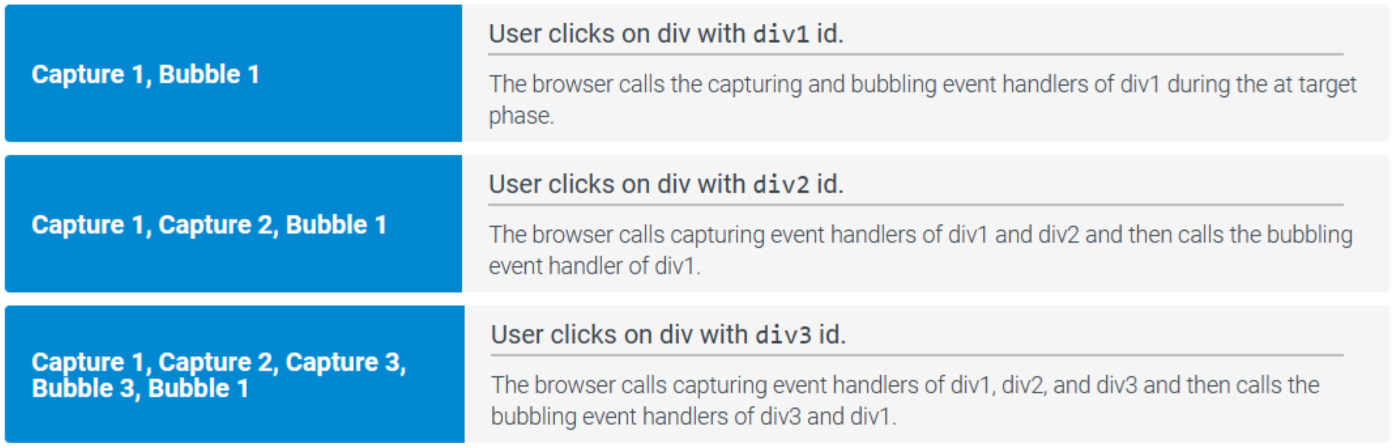
1. Vredefort Dome, South Africa
2. Chicxulub Crater, Mexico
3. Sudbury Basin, Ontario, Canada

```javascript
list = document.getElementById("strikeList");

list.addEventListener("mouseover", function(e) {
   e.target.style.color = "red";
});

list.addEventListener("mouseout", function(e) {
   e.target.style.color = "black";
}, true);
```



Event Bubbling and Capturing :

Given the HTML and JavaScript below, match the order of alerts to the action performed by the user.



Capture 1, Bubble 1	User clicks on div with `div1` id. The browser calls the capturing and bubbling event handlers of div1 during the at target phase.
Capture 1, Capture 2, Bubble 1	User clicks on div with `div2` id. The browser calls capturing event handlers of div1 and div2 and then calls the bubbling event handler of div1.
Capture 1, Capture 2, Capture 3, Bubble 3, Bubble 1	User clicks on div with `div3` id. The browser calls capturing event handlers of div1, div2, and div3 and then calls the bubbling event handlers of div3 and div1.

Example :

Run the code. Click on div , body and button and observe the output in console

[BubblingCapturing.html (https://sjeccd.instructure.com/courses/39451/files/7712436?wrap=1)](https://sjeccd.instructure.com/courses/39451/files/7712436?wrap=1)

[EventBubblingCapturing.html (https://sjeccd.instructure.com/courses/39451/files/7712438?wrap=1)](https://sjeccd.instructure.com/courses/39451/files/7712438?wrap=1)