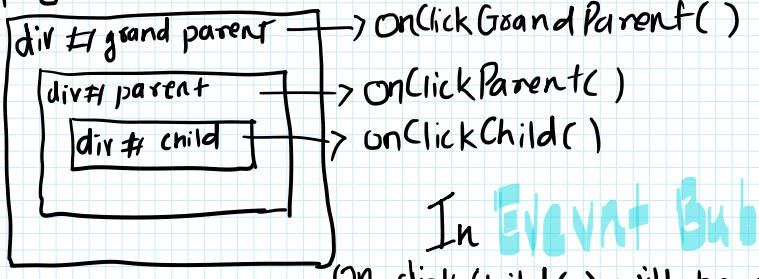
03 January 2024 09:49

Event Bubbling & Capturing

Event Bubbling and capturing are two ways of capturing event propagation in the DOM tree.



In Event Bubbling if we click on child div then onClickChild() will be called first then onClickParent() will called then onClickGrandParent() will be called this calling propogates like the water bubble which floats from the generation point to the top.

Event Capturing is opposite to the Event Bubbling if we click on child then onClickGrandParent() will be called first then onClickParent() will be called then onclickchild() will be called, this is also known as Event Trickling.

ex: addEventListener('click', () => {

}, useCapture) → this is a flag if it is true Event Capturing will happen if this is false or we don't pass the flag then Event Bubbling will happen.

In W3C model the cycle continues first Event Capturing will happen then Event Bubbling will happen but using the flag we can control whether Event Bubbling should happen or Event Capturing.

ex 1) if we click on child div event bubbling will happen as we are not passing flag which is equal to false.

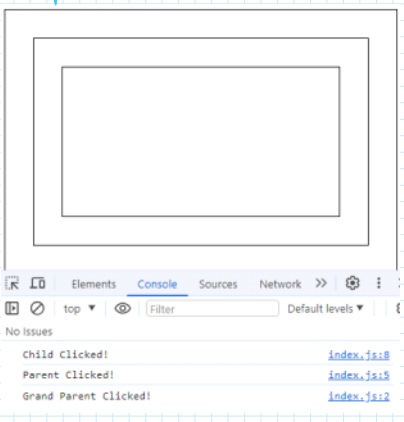
index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
    <style>
      div {
        min-width: 100px;
        min-height: 100px;
        padding: 30px;
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <div id="grandparent">
      <div id="parent">
        <div id="child"></div>
      </div>
    </div>
    <script src="./index.js"></script>
  </body>
</html>
```

index.js

```
document.querySelector("#grandparent").addEventListener("click", () => {
  console.log("Grand Parent Clicked!");
});
document.querySelector("#parent").addEventListener("click", () => {
  console.log("Parent Clicked!");
});
document.querySelector("#child").addEventListener("click", () => {
  console.log("Child Clicked!");
});
```

output



ex2: if we click on child div then Event Capturing will happen as we are passing flag as true.

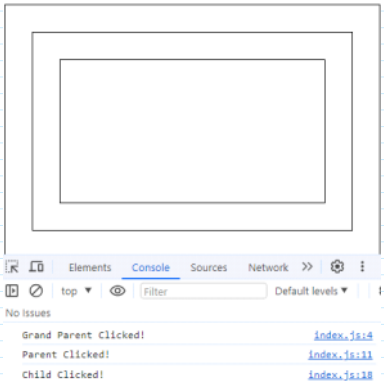
index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
    <style>
      div {
        min-width: 100px;
        min-height: 100px;
        padding: 30px;
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <div id="grandparent">
      <div id="parent">
        <div id="child"></div>
      </div>
    </div>
    <script src="./index.js"></script>
  </body>
</html>
```

index.js

```
document.querySelector("#grandparent").addEventListener(
  "click",
  () => {
    console.log("Grand Parent Clicked!");
  },
  true
);
document.querySelector("#parent").addEventListener(
  "click",
  () => {
    console.log("Parent Clicked!");
  },
  true
);
document.querySelector("#child").addEventListener(
  "click",
  () => {
    console.log("Child Clicked!");
  },
  true
);
```

output



ex3: According to W3C the whole chain happens like first Event Capturing will happen then Event bubbling will happen, so if put the parent flag false then that event will not happen in Event capturing which happen first but the parent will happen when the Event capturing will finish and Event Bubbling will start.

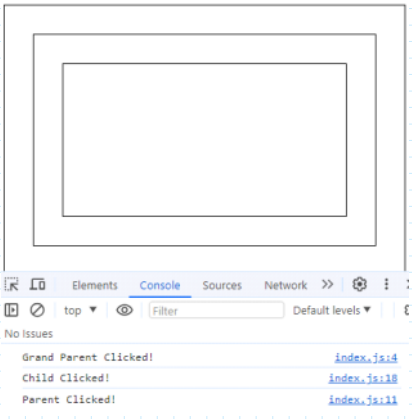
index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
    <style>
      div {
        min-width: 100px;
        min-height: 100px;
        padding: 30px;
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <div id="grandparent">
      <div id="parent">
        <div id="child"></div>
      </div>
    </div>
    <script src="./index.js"></script>
  </body>
</html>
```

index.js

```
document.querySelector("#grandparent").addEventListener(
  "click",
  () => {
    console.log("Grand Parent Clicked!");
  },
  true
);
document.querySelector("#parent").addEventListener(
  "click",
  () => {
    console.log("Parent Clicked!");
  },
  false
);
document.querySelector("#child").addEventListener(
  "click",
  () => {
    console.log("Child Clicked!");
  },
  true
);
```

output



ex4: if we click on child and parent and child are false.

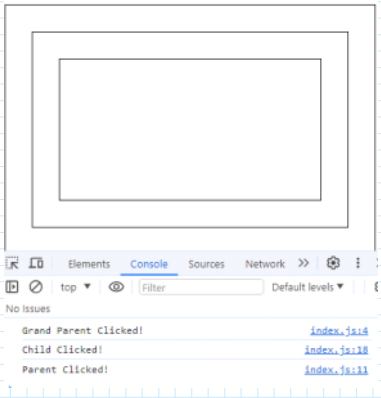
index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
    <style>
      div {
        min-width: 100px;
        min-height: 100px;
        padding: 30px;
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <div id="grandparent">
      <div id="parent">
        <div id="child"></div>
      </div>
    </div>
    <script src="./index.js"></script>
  </body>
</html>
```

index.js

```
document.querySelector("#grandparent").addEventListener(
  "click",
  () => {
    console.log("Grand Parent Clicked!");
  },
  true
); //capturing
document.querySelector("#parent").addEventListener(
  "click",
  () => {
    console.log("Parent Clicked!");
  },
  false
); //bubbling
document.querySelector("#child").addEventListener(
  "click",
  () => {
    console.log("Child Clicked!");
  },
  false
); //bubbling
```

output



- How can we stop event propogation as this are quit expensive.

We have access to the event object in event listener, the event object has special method "stopPropagation()", which will help for stopping the propagation of event.

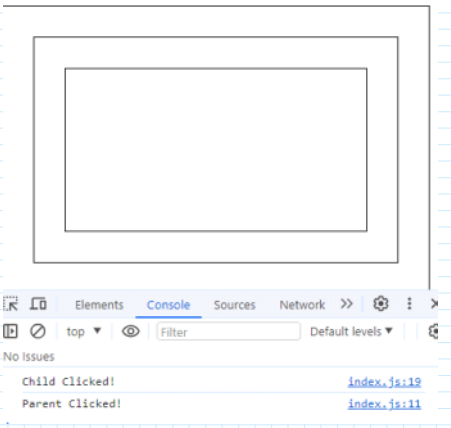
index.htm

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Document</title>
    <style>
      div {
        min-width: 100px;
        min-height: 100px;
        padding: 30px;
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <div id="grandparent">
      <div id="parent">
        <div id="child"></div>
      </div>
    </div>
    <script src="./index.js"></script>
  </body>
</html>
```

index.js

```
document.querySelector("#grandparent").addEventListener(
  "click",
  () => {
    console.log("Grand Parent Clicked!");
  },
  false
);
document.querySelector("#parent").addEventListener(
  "click",
  (e) => {
    console.log("Parent Clicked!");
    e.stopPropagation();
  },
  false
);
document.querySelector("#child").addEventListener(
  "click",
  () => {
    console.log("Child Clicked!");
  },
  false
);
```

output: if click on chid div



By: Iliyaskhan
@iliyaskhantws