



**JAVASCRIPT**



**SIÊU TỐC**



**DOM EVENT**

**Event Capturing**

**stopImmediatePropagation ( );**

**37.12**



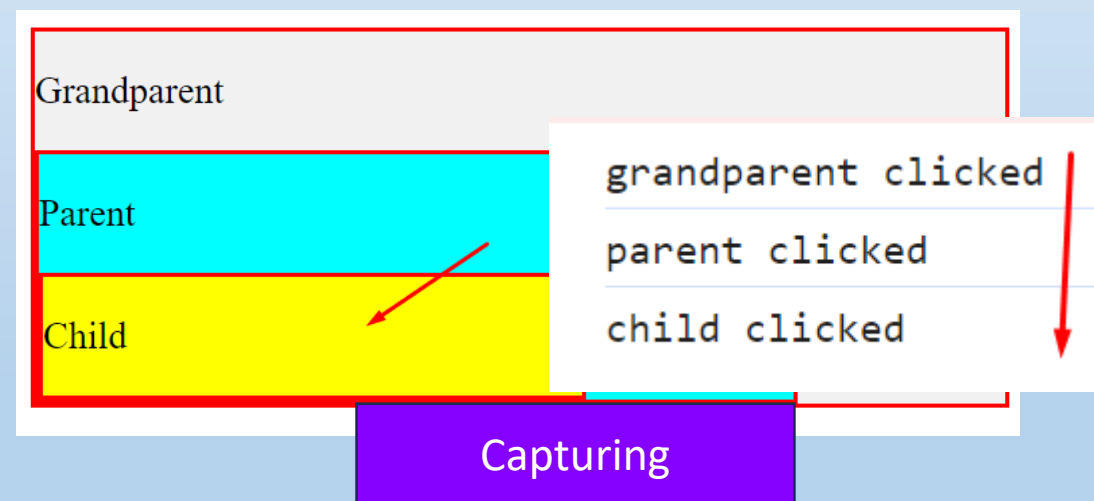
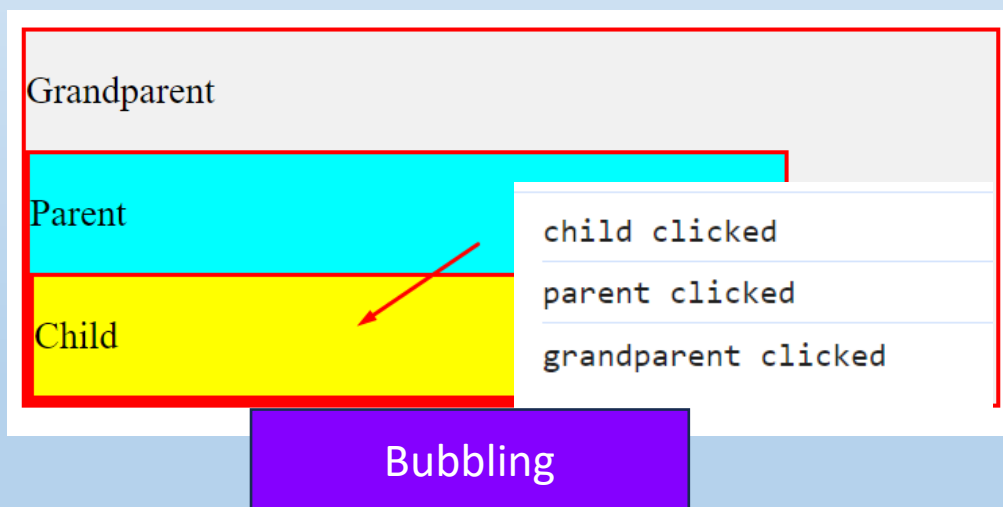
12

## Bubbling và Capturing

### Bubbling và Capturing

**Bubbling** : Sự kiện được phát đi từ phần tử đích lên cây DOM đến gốc (từ trong ra ngoài). Mặc định `{capture: false}`

**Capturing** : Sự kiện được phát đi từ gốc của cây DOM đến phần tử đích (từ ngoài vào trong). nếu `{capture: true}`





12


# Bubbling và Capturing


index.html

```
<div id="grandparent">
  <p>Grandparent</p>
  <div id="parent">
    <p>Parent</p>
    <div id="child">
      <p>Child</p>
    </div>
  </div>
</div>
```

style.css

```
/* set màu nền cho khối div grandparent */
div {
  border:  red 2px solid;
}
#grandparent {
  width: 90vw;
  background-color:  #f1f1f1;
}

#parent {
  width: 70vw;
  background-color:  cyan;
}

#child {
  width: 50vw;
  background-color:  yellow;
}
```

12

## Bubbling và Capturing

```
const child = document.querySelector("#child");
const parent = document.querySelector("#parent");
const grandparent = document.querySelector("#grandparent");

child.addEventListener("click", function () {
  console.log("child clicked");
});

parent.addEventListener("click", function () {
  console.log("parent clicked");
});

grandparent.addEventListener("click", function () {
  console.log("grandparent clicked");
});
```

Grandparent

Parent

Child

child clicked

parent clicked

grandparent clicked

```
child.addEventListener(
  "click",
  function () {
    console.log("child clicked");
  },
  {capture: true}
);
```

```
parent.addEventListener(
  "click",
  function () {
    console.log("parent clicked");
  },
  {capture: true}
);
```

```
grandparent.addEventListener(
  "click",
  function () {
    console.log("grandparent clicked");
  },
  {capture: true}
);
```

grandparent clicked

parent clicked

child clicked

12

## *stopPropagation và stopImmediatePropagation*

1. Dùng *stopPropagation()*; chỉ ngăn chặn sự kiện lan truyền từ con lên cha, không ngăn chặn sự kiện của cùng 1 element
2. Dùng *stopImmediatePropagation()*; ngăn chặn sự kiện của cùng 1 element

```
child.addEventListener("click", function (e) {
  e.stopPropagation();
  console.log("child clicked");
});

child.addEventListener("click", function (e) {
  e.stopPropagation();
  console.log("child clicked 2");
});
```

```
child clicked
child clicked 2
```

```
child.addEventListener("click", function (e) {
  e.stopImmediatePropagation();
  console.log("child clicked");
});

child.addEventListener("click", function (e) {
  e.stopPropagation();
  console.log("child clicked 2");
});
```

```
child clicked
```





**JAVASCRIPT**



**SIÊU TỐC**



**DOM EVENT**

**37.13**

**Event Delegation**



13

## Event delegation

```
<body>
  <!-- Ví dụ về Event delegation -->
  <h1>Bài học Event Delegation</h1>
  <form action="" id="form">
    <input type="text" name="" id="username" placeholder="username">
    <input type="text" placeholder="message" id="message">
    <input type="submit" value="Send">
  </form>
  <ul id="list_item">
    <li>Tin nhắn 1</li>
    <li>Tin nhắn 2</li>
    <p>Đoạn văn bản</p>
  </ul>
  <!-- js -->
  <script src="./main.js"></script>
</body>
```

## Bài học Event Delegation




- Tin nhắn 1
- Tin nhắn 2

Đoạn văn bản

13

## Event delegation

```
//ngăn submit mặc định của form
const form = document.getElementById("form");
const list_item = document.getElementById("list_item");
form.addEventListener("submit", function (e) {
  e.preventDefault(); //ngăn submit mặc định của form
  //lấy giá trị của ô username và message
  const username = document.getElementById("username").value;
  const message = document.getElementById("message").value;
  //tao thẻ li
  const newLi = document.createElement("li");
  newLi.textContent = username + ": " + message;
  //thêm vào ul
  list_item.append(newLi);
  //reset form
  form.reset(); //phương thức reset sẽ reset lại tất cả các ô input trong form
});
```



13

## Event delegation

**Vấn đề:** khi thêm 1 li mới vào ul, thì li đó không có sự kiện click

```
list_item.addEventListener("click", function (e) {
  const li = e.target;
  // li.remove();
  //hoặc nếu cần kiểm tra xem phần tử đó có phải là li không
  if (li.tagName === "LI") {
    li.remove();
  }
});
```

### Bài học Event Delegation

username  message

- Tin nhắn 1 ✓
- Tin nhắn 2 ✓

Xóa được

Đoạn văn bản

- batman: có thích khách ✗
- tuhoc.cc: thích khách ở đâu ✗

KHÔNG Xóa được

```
> lis
< ▼ NodeList(2) [li, li] i
  ► 0: li
  ► 1: li
    length: 2
  ► [[Prototype]]: NodeList
>
```

Danh sách các thẻ li vẫn chỉ có 2 thẻ ban đầu ?

13

## Event delegation

Cách thực hiện *Event delegation*:

1. *thêm sự kiện vào phần tử cha* (trong ví dụ là sự kiện **click** vào **ul**)
2. *kiểm tra xem phần tử nào xảy ra sự kiện* (**event.target**)
3. *thực hiện công việc cần thiết*

```
//Xóa phần tử li khi click vào dùng event delegation
list_item.addEventListener("click", function (e) {
  const li = e.target;
  // li.remove();
  //hoặc nếu cần kiểm tra xem phần tử đó có phải là li không
  if (li.tagName === "LI") {
    li.remove();
  }
});
```

Trong đoạn code trên: Chúng ta *click vào phần tử con li* thì nó sẽ xóa phần tử đó đi mặc dù chúng ta *lắng nghe sự kiện click trên phần tử cha ul*  
 → điều này gọi là *event delegation*: sự kiện được giao phó từ phần tử *cha* xuống phần tử *con*



**JAVASCRIPT**



**SIÊU TỐC**



**DOM EVENT**

**37.14**

**Mini project 02**  
**Slide ảnh**







**JAVASCRIPT**



**SIÊU TỐC**



**DOM EVENT**

**37.15**

**Mini project 02**  
**Slide ảnh - p2**





## Event JS Project 02



Bỏ lại tiên cảnh 02







# JAVASCRIPT



## SIÊU TỐC



# DOM EVENT

37.16

## Mini project 03



Prev

Next



<http://Tuhoc.CC>



# JAVASCRIPT



## SIÊU TỐC



# DOM EVENT

37.17

## Mini project 03



Prev

Next



<http://Tuhoc.CC>





# JAVASCRIPT



## SIÊU TỐC



# DOM EVENT

37.18

## Mini project 04



Nhập nội dung tìm kiếm ...





**JAVASCRIPT**  
**SIÊU TỐC**



**DOM EVENT**

**37.19**

**Mini project 04**



Nhập nội dung tìm kiếm ...







# JAVASCRIPT



## SIÊU TỐC



# DOM EVENT

37.20

## Mini project 05

CLICK VÀO EM



<http://Tuhoc.CC>





**JAVASCRIPT**  
**SIÊU TỐC**



**DOM EVENT**

**Mini project 05**

**37.21**

CLICK VÀO EM



<http://Tuhoc.CC>