Go farther with Events

In this lesson, we will introduce the concept of Event Propagation and cover all the important points related to event propagation.

WE'LL COVER THE FOLLOWING

Event Propagation

The DOM represents a web page as a hierarchy of nodes. Events triggered on a child node are going to get triggered on the parent node, then the parent node of the parent node, up until the root of the DOM (the `document` variable). This is called event *propagation*. To see propagation in action, use this HTML code to create a small DOM hierarchy.

Output

HTML

```
<html>
 <head>
 </head>
 <body>
                <p id="para">A paragraph with a <button id="propa">button</button> inside</p>
 </body>
</html>
```

Here's the complementary JavaScript code. It adds `click` event handlers on the button, its parent (the paragraph), and the parent of that too (the root of the DOM).

Output

JavaScript

HTML

```
<html>
 <head>
 </head>
 <body>
                <p id="para">A paragraph with a <button id="propa">button</button> inside</p>
 </body>
</html>
```

Console

Clear

The result in the browser console demonstrates the propagation of `click` events from the button up to the document level. You clicked the button, which means you also clicked the paragraph, which means you also clicked the document.

But maybe you only want an event to kick off once the button is clicked and not count its larger ecosystem? Event propagation can be interrupted at any moment by calling the `stopPropagation()` method on the `Event` object from an event handler. This is useful to avoid the same event being handled multiple times.

Adding a line in the button's click handler prevents the `click` event from propagating everywhere in the DOM tree.

Output

JavaScript

HTML

```
<html>
 <head>
 </head>
 <body>
                <p id="para">A paragraph with a <button id="propa">button</button> inside</p>
 </body>
</html>
```

Console

Clear

Cancelling the Default Behavior of an Action

Most of the user actions on a page are associated to a default behavior. Clicking on a link navigates to the link target, clicking anywhere with the right mouse button show a contextual menu, etc. Cancelling a default behavior is possible by calling the `preventDefault()` method on the `Event` object in an event handler.

Let's use the following HTML and JavaScript code to see this possibility in action

Output

JavaScript

HTML

```
<html>
 <head>
 </head>
 <body>
                <p>Time on your hands? <a id="forbidden" href="https://9gag.com/">Click here<
 </body>
</html>
```

Now clicking on the links shows a dialog instead of navigating to its target.