Unit 2: HTML5, JQuery And Ajax Handling Events in JavaScript

Event flow

Assuming that you have the following HTML document:

```
<!DOCTYPE html>
<html>
<head>
    <title>JS Event Demo</title>
</head>
<body>
    <div id="container">
        <button id='btn'>Click Me!</button>
    </div>
</body>
```

When you click the button, you're clicking not only the button but also the button's container, the div, and the whole webpage.

Event flow explains the order in which events are received on the page from the element where the event occurs and propagated through the DOM tree.

There are two main event models: **event bubbling** and **event capturing**.

Event bubbling

In the event bubbling model, an event starts at the most specific element and then flows upward toward the least specific element (the document or even window).

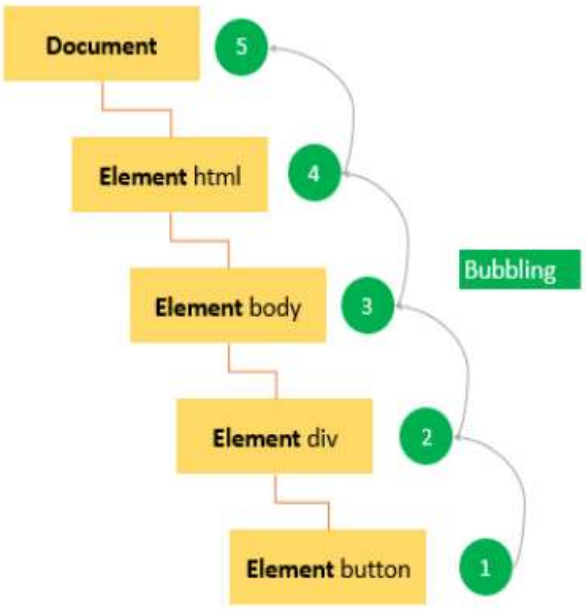
When you click the button, the click event occurs in the following order:

- button
- div with the id container

- body
- html
- document

The click event first occurs on the button which is the element that was clicked. Then the click event goes up the DOM tree, firing on each node along its way until it reaches the document object. Modern web browsers bubble the event up to the window object.

The following picture illustrates the event bubbling effect when users click the button:



Event capturing

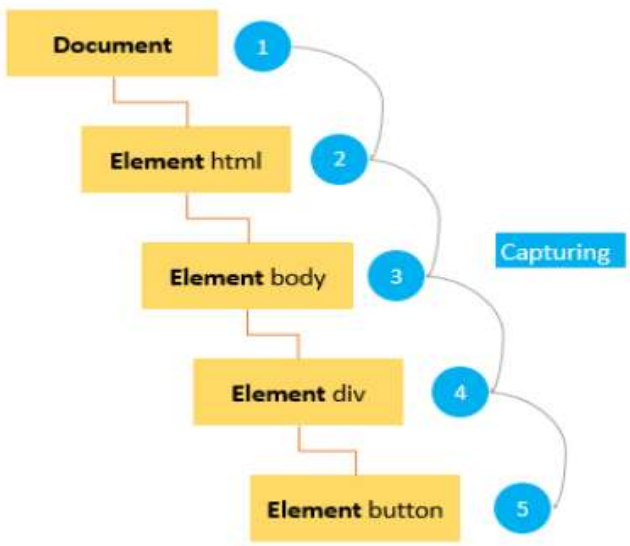
In the event capturing model, an event starts at the least specific element and flows downward toward the most specific element.

When you click the button, the click event occurs in the following order:

- document
- html

- body
- div with the id container
- button

The following picture illustrates the event capturing effect:

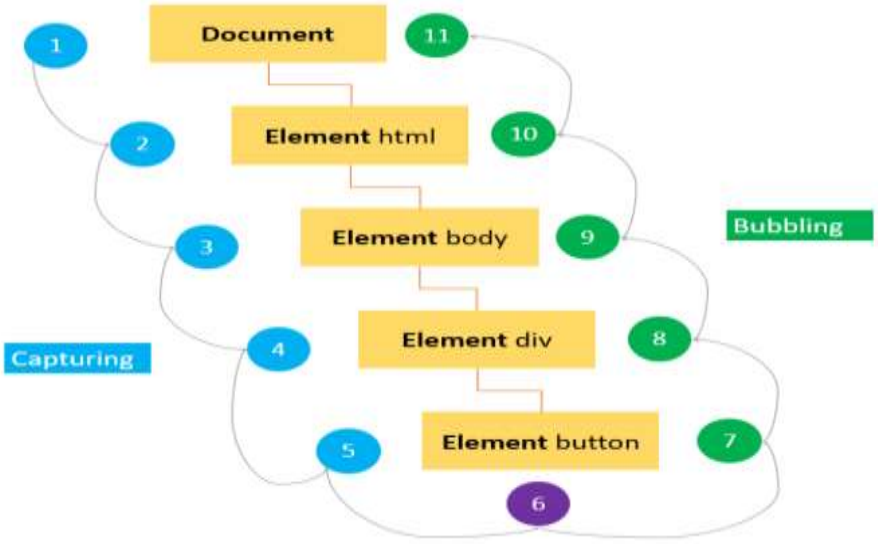


DOM Level 2 Event flow

DOM level 2 events specify that event flow has three phases:

1. First, event capturing occurs, which provides the opportunity to intercept the event.
2. Then, the actual target receives the event.
3. Finally, event bubbling occurs, which allows a final response to the event.

The following picture illustrates the DOM Level 2 event model when users click the button:



Submit & Reset Buttons

Buttons are defined by changing the input element's type attribute. There is two different kinds of buttons – the submit button and the reset button. Submit and Reset buttons help the user tell the browser what to do with a form on the website.

The INPUT element defines an input field. When specify "submit" (or "reset") for the type attribute of this element, a submit button (or a reset button) is created.

<input type="submit" value="Submit">

<input type="reset" value="Reset">

Attribute	Value	Explanation
type=" "	submit	the type of input field submit : creates a submit button reset : creates a reset button
	reset	
name=" "	button	a unique name for the button

	name	
value=" "	button text	the text displayed on the button

type="submit"

Creates a submit button on the form.

When this button is clicked, the form data is submitted to the server.

type="reset"

Creates a reset button on the form.

When this button is clicked, the input is reset.

name=""

The button name is used to identify the clicked submit button.

value=""

Value is the text displayed on the button.

CODE:

```
<form action="url-to-formmail-provided-by-your-ISP" method="post">
    First name: <input type="text" name="firstname" /> <br /></br>
    Surname: <input type="text" name="surname" /><br /></br>
    <input type="reset" value="Clear form" />
    <input type="submit" value="Submit now" />
</form>
```

OUTPUT:

First name:

Surname:

Clear form Submit now