

Where are events from?

Web APIs

As mentioned, there are lots of Web APIs available straight out of the box. I mentioned a few of them, including the Fetch API, Websockets API, etc.

But, where are Events from?

Remember in Part 1 of this series, we learnt that the DOM represents a document as a tree. This means that each branch of the tree ends in a node, and each node contains objects. But importantly for us, nodes can also have event listeners and handlers attached to them; once an event is triggered, the event handlers get executed.

The DOM has many interfaces (such as the Document, Window, NodeList, etc.) but the one of interest to us is the Event interface.

So the short answer is that Events are from the DOM.

What is the Event Interface?

So, we know that Events are given to us by the DOM. But lets keep going **down the rabbit hole ... where exactly in the DOM can they be found?** This is where the Event interface comes into the picture.

The Event interface represents an event which takes place in the DOM. There are many types of events, some of which use other interfaces based **on the main Event interface. For example, the keypress event inherits from** the Event interface, but it also has a whole bunch of its own properties and methods found in the KeyboardEvent interface.

Make sense?

Lets see a picture

Pictures speak louder than words. This is how I picture where Events are from.

WHERE ARE EVENTS FROM

Web API



This is where it all begins.
When writing code, there
are lots of Web APIs
available.

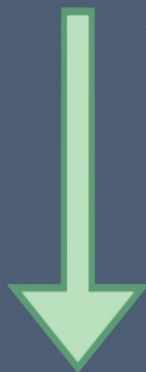
Fetch API

Storage API Battery API

Websockets API etc etc ...

But the most important for us is:

DOM API



The DOM API provides
various interfaces that help
us to interact with the DOM.

NodeList Document

Window URL

Element etc etc ...

But the most important for us is:

Event Interface

There are many types of events.

There are many types of events,
some of which use other interfaces
based on the main Event interface.

MouseEvent FocusEvent
 TouchEvent etc etc ...

