

Theory: DOM events

⌚ 8 minutes

0 / 5 problems solved

Skip this topic

Start practicing

452 users solved this topic. Latest completion was about 1 hour ago.

The DOM represents a set of capabilities that a web browser provides to the JavaScript environment. For example, when you start entering your search request in a Google search box, it gives you several recommendations that you can instantly see. When you hover your mouse over a link, the link often changes color. You can even control the video player on YouTube by pausing or changing the volume of the video. All these actions are based on the same principles. We will learn more about them in this topic.

§1. DOM events

Let's look at real-life examples. You get in the elevator and press the button for the 7th floor. This button sends a signal to the system, saying that the elevator should go up to the 7th floor. The system receives this signal, realizes that you need the 7th floor, and begins to lift the elevator to the correct altitude.

In programming words, when you clicked the button, you created an **event** to visit the 7th floor. The system processed this event by lifting the elevator to the required distance. However, if you had clicked the 13th-floor button, the event would have been different, and other actions would have been performed.

Let's draw another parallel with a web application. You are looking for a vacation hotel and click on one of the links to see the details. In this case, when you click the link, a **click event** occurs and the system starts responding to you. To handle your event, the browser downloads a new page with additional information about the hotel. If you click on a different link to a hotel, different information will be loaded. If you specify some details in a hotel filter, other events will occur that lead to different actions.

Events that occur in a web application are called **DOM events**. Developers use them on elements in an HTML page: button, img, input, form, div and so on.

§2. Types of events

In the example above, we only looked at the click event, but there are more than 100 of such events! When you resize a window, double-click a mouse button, move the mouse pointer or paste something from the clipboard, you trigger another corresponding event. [Here](#) you can find the whole list of them.

As you can see, the number of events is enormous, so it is logical to divide the list of events into several categories, that is, into types of events. The most popular ones are **mouse**, **keyboard** and **focus events**. Let's take a more in-depth look at them!

Mouse events occur when the user interacts with the mouse. The most popular ones are:

- `click` when the user clicks the left mouse button;
- `contextmenu` when the user clicks the right mouse button;
- `dblclick` when the user clicks the left mouse button twice;
- `mouseenter` when the user moves the mouse pointer towards the element;
- `mouseleave` when the user moves the mouse pointer away from the element.

Keyboard events are the results of the user's interaction with the keyboard. There are three types of such events:

- `keydown` when the user presses any key;

Current topic:

DOM events

...

Topic depends on:

✗ DOM

...

Topic is required for:

The "addEventListener" method

...

Table of contents:

1 DOM events

§1. DOM events

§2. Types of events

§3. Conclusion

Feedback & Comments

- `keypress` when the user presses any key except Shift, Fn, or CapsLock;
- `keyup` when the user releases a key.

Focus events occur when an element receives or loses focus. This is especially useful for input when developers mark the incorrectly filled input field in red, such as the email or phone fields.

- `focus` when an element receives focus;
- `blur` when an element loses focus.

§3. Conclusion

DOM events are best used when creating attractive and interactive web pages. They can be assigned to HTML elements such as button, div, input, and so on. Remember the several types of DOM events: **mouse**, **keyboard**, and **focus events**. Now you know that an interactive web application is not magic, and to create it you just need to learn how to work with DOM events.

 Report a typo

51 users liked this theory. 1 didn't like it. What about you?



Start practicing

[Comments \(0\)](#)[Hints \(0\)](#)[Useful links \(0\)](#)[Show discussion](#)