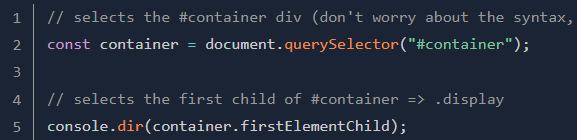
**DOM MANIPULATION & EVENTS**

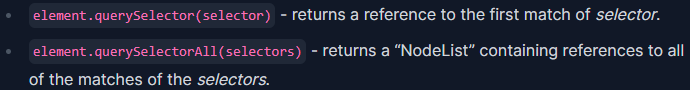
1. **Document Object Model (DOM)**

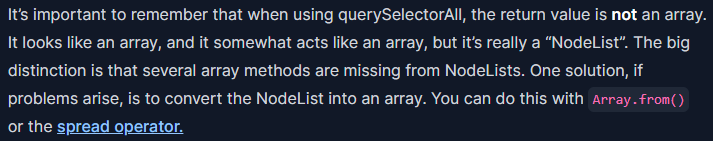
The DOM (or Document Object Model) is a tree-like representation of the contents of a webpage. Example:

1. Targeting node with selectors

You can use a combination of CSS-style selectors(i.e., .display, #container) and relationship properties(i.e., firstElementChild or lastElementChild) to target the nodes you want.

1. **DOM Methods (exericse at odin web)**
2. Query Selectors

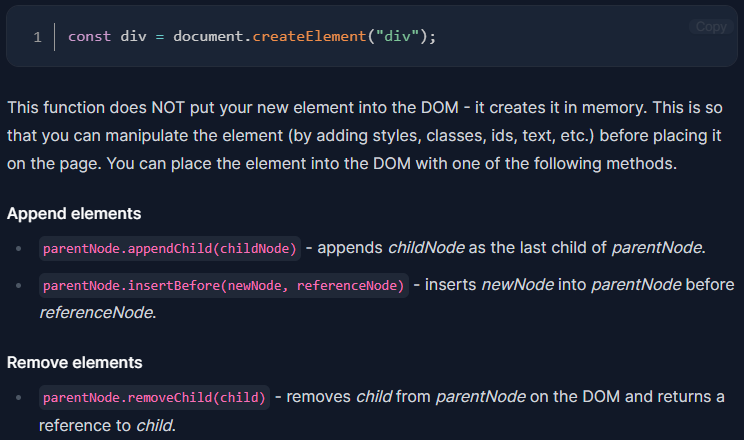
There are several other, but we won't be going over them now.

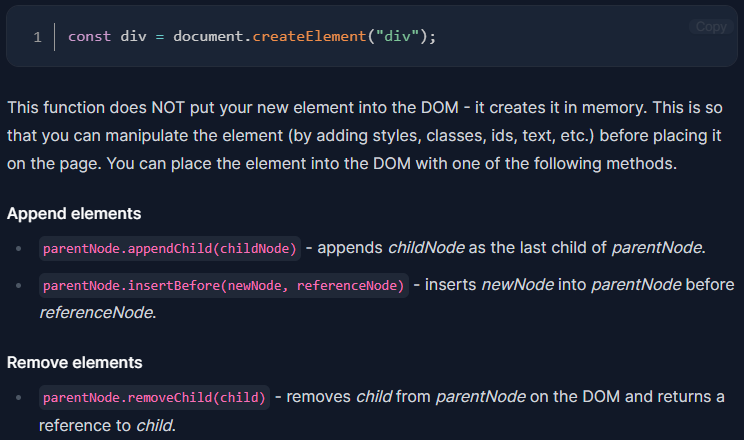
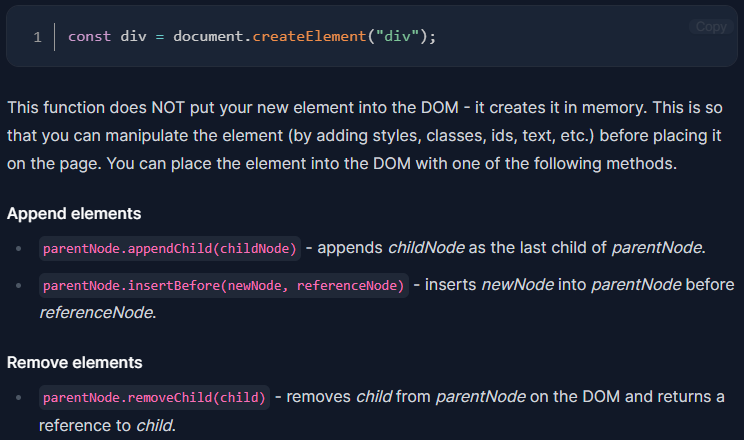
**Note**:

Spread operator link: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Spread_syntax>

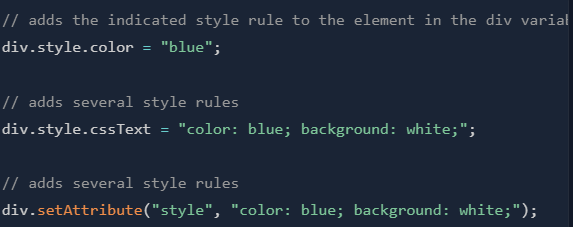
Note: There are older methods available for grabbing element references, such as Document.getElementById() & Document.getElementsByTagName(). These two work better in older browsers than the modern methods like querySelector(), but are not as convenient.

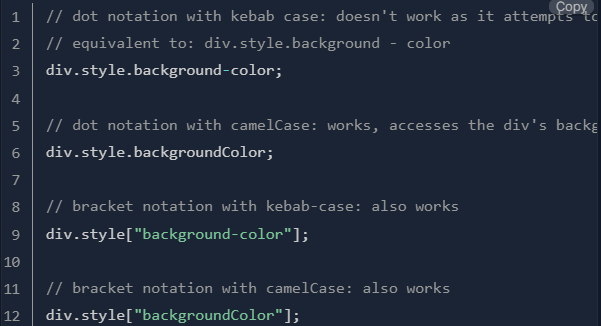
1. Element Creation

[options] in this case means you can add some optional parameters to the function, you don’t worry about these at this point.

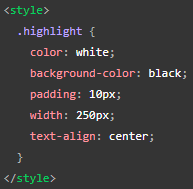
1. Append Elements
2. Remove Elements
3. Altering Elements

When you have a reference to an element, you can use that reference to alter the element’s own properties. This allows you to do many useful alterations, like adding, removing, or altering attributes, changing classes, adding inline style information, and more.

1. Adding inline style

When accessing a kebab-cased CSS property like background-color with JS, you will need to either use camelCase with dot notation or bracket notation. When using bracket notation, you can use either camelCase or kebab-case, but the property name must be a string.

 There is another common way to dynamically manipulate styles on your document:

The first method(element.style) takes less setup and is good for simple uses, whereas the second method(.setAttribute) is as you start building larger and more involved apps, you will probably start using the second method more, but it is really up to you.

1. Editing Attributes

* *element*.setAttribute() : sets a new value to an attribute. If the attribute does not exist, it create the attribute with the value. Example: 



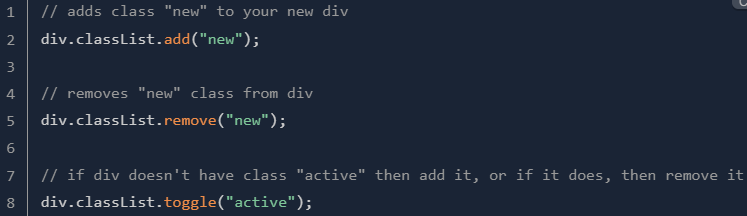
* *element.*getAttribute() : returns the value of an element's attribute. Example:

Get the value of the class attribute of an element.

* *element.*removeAttibute() : removes an attribute from an element. Example:

remove id attribute from div.

HTML attribute(full): <https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes>

1. Working with classes

It is often standard (and cleaner) to .toggle a CSS style rather than adding and removing inline CSS.

1. Adding text content

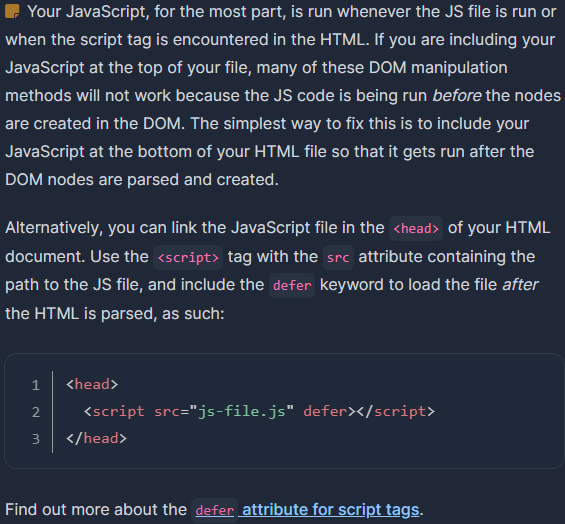
creates a text node containing "Hello World!" and inserts it in div.



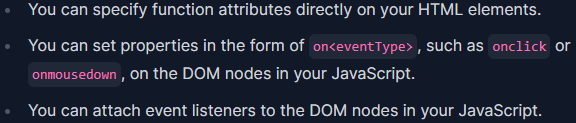
1. Adding HTML content

renders the HTML inside div

Note: using textContent is preferred over innerHTML for adding text, as innerHTML should be used sparingly to avoid potential security risks. To understand the dangers of using innerHTML, watch this: <https://youtu.be/ns1LX6mEvyM?si=wHV_xaY6MUupR6Ha>

Keep in mind that the JavaScript does not alter your HTML, but the DOM - your HTML file will look the same, but the JavaScript changes what the browser renders.

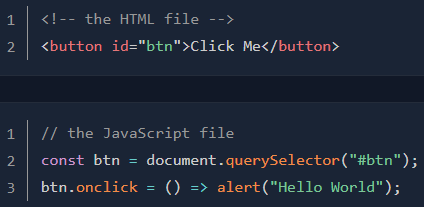
1. **Events**

Events are actions that occur on your webpage, such as mouse-clicks or key-presses. Using JavaScript, we can make our webpage listen to and react to these events. There are three method:

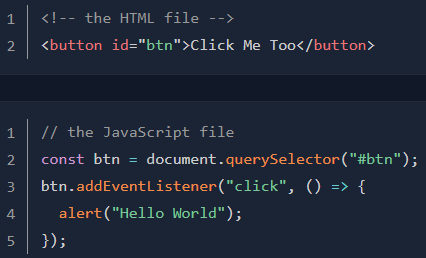
Event listeners are definitely the preferred method, but you will regularly see the others in use:

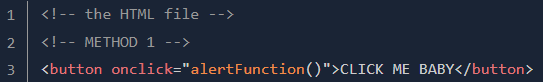
* Method 1

We can only set one “onclick” property per DOM element, so we’re unable to run multiple separate functions in response to a click event using this method. (This solution is less than ideal because we’re cluttering our HTML with JavaScript).

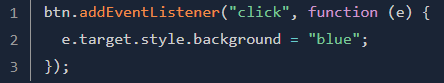
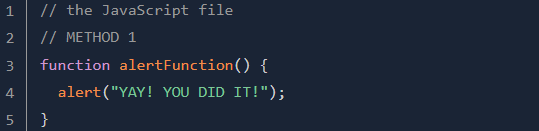
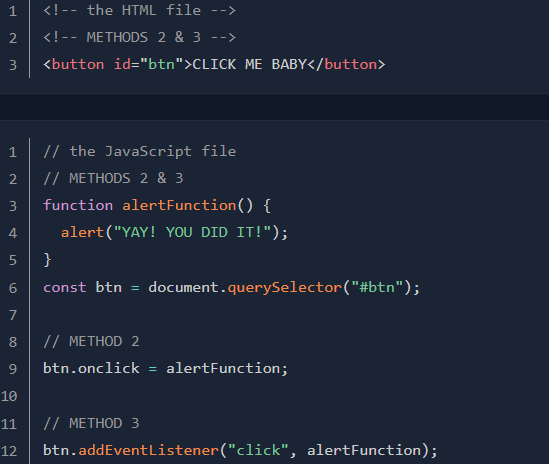
* Method 2

We still have the problem that a DOM element can only have one “onclick” property. (This is a little better. We’ve moved the JS out of the HTML and into a JS file).

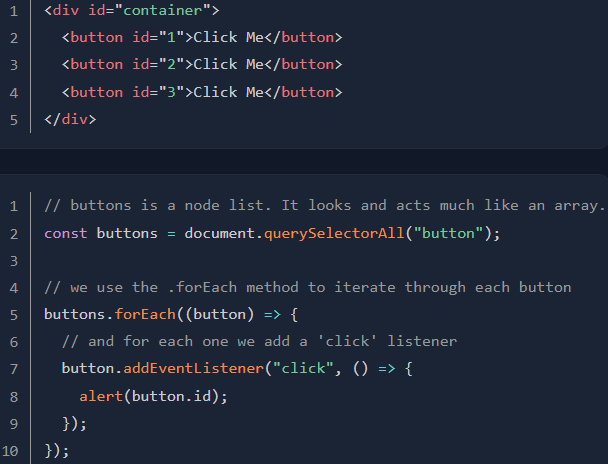
* Method 3

Now we allow multiple event listeners if the need arises.

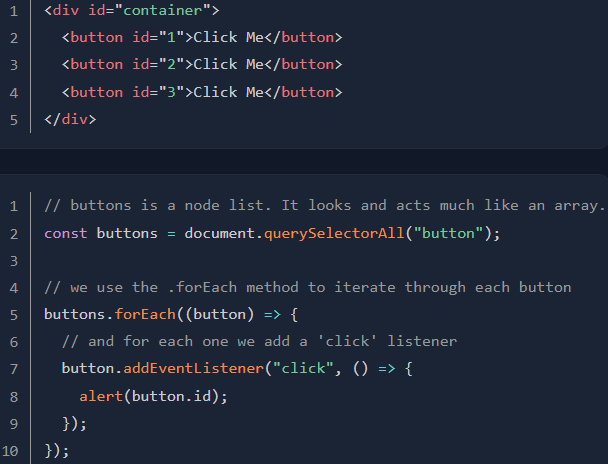
* Note that all three of these methods can be used with named functions like so:

Using named functions can clean up your code considerably, and is a really good idea if the function is something that you are going to want to do in multiple places.

The e parameter in that callback function(function as argument) contains an object that references the event itself.

* Attaching listeners to Groups of nodes

Lanjutannya di bawah(di page selanjutnya)

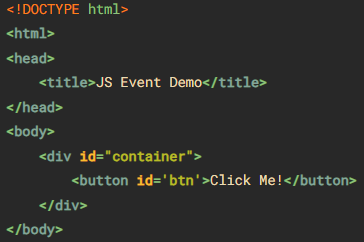
* Some useful events include
* click
* change
* dblclick
* keydown
* keyup

Full: <https://www.w3schools.com/jsref/dom_obj_event.asp>

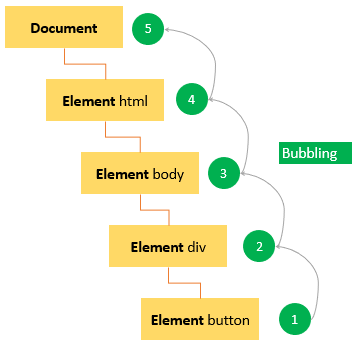
1. **Event Flow (summary is below E. Event Object)**

src: <https://www.javascripttutorial.net/javascript-dom/javascript-events/>

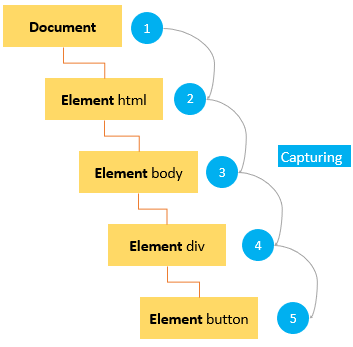
Assuming that you have the following HTML document:

There are two main event models:

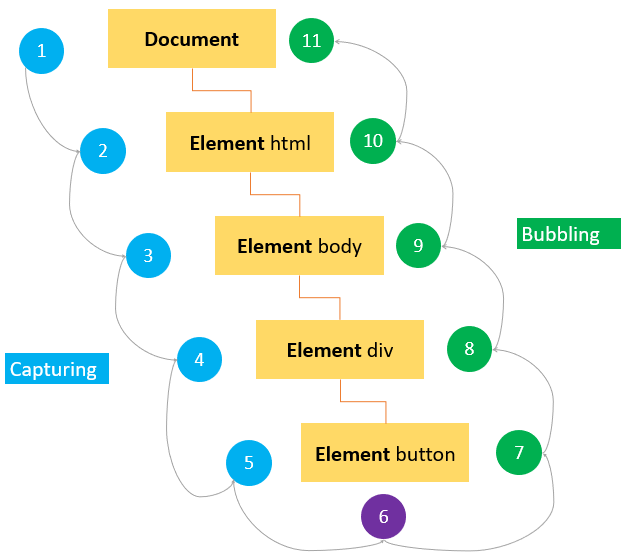
1. Event bubbling

Event starts at the most specific element and then flows upward toward the least specific element (the document or even window in modern browsers). When you click the button, the ‘click’ event occurs in the following order:

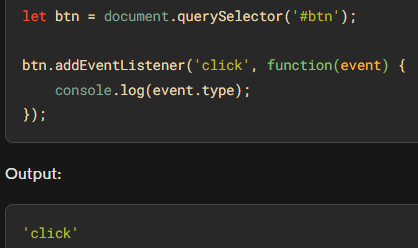
1. Event capturing

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:

1. DOM level 2 event flow

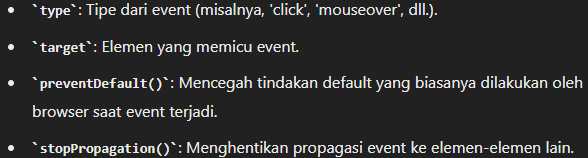
First, event capturing occurs. Then, the actual target receives the event. Finally, event bubbling occurs.

1. **Event Object**

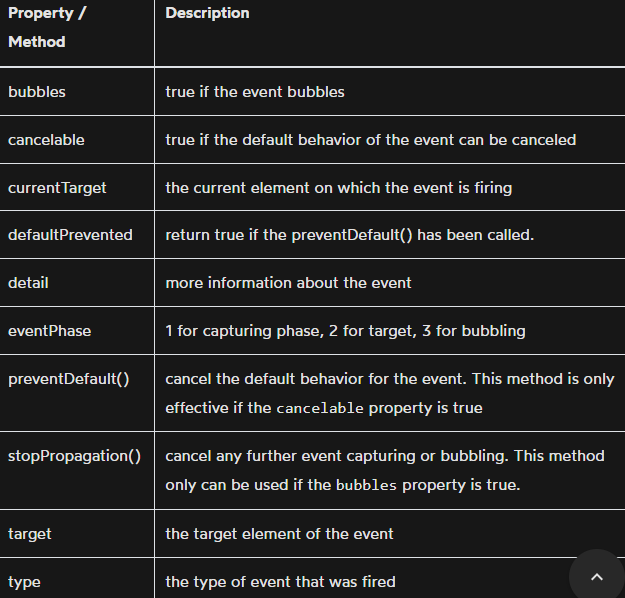
When the event occurs, the web browser passes an ‘Event’ object to the event handler:

Event handler is a function that is called when an event occurs.

Event object adalah sebuah objek yang dikirimkan ke event handler ketika sebuah event terjadi. Objek ini berisi informasi tentang event tersebut, seperti tipe event, elemen yang memicu event, dan berbagai properti lainnya yang berkaitan dengan event tersebut.

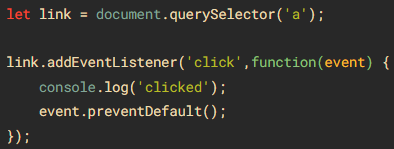
Objek event dalam JavaScript memiliki banyak properti dan metode yang berguna, seperti:

Most commonly used properties and methods of the ‘event’ object:

Note: ‘event’ object is only accessible inside the event handler.

* preventDefault() : To prevent the default behavior of an event.

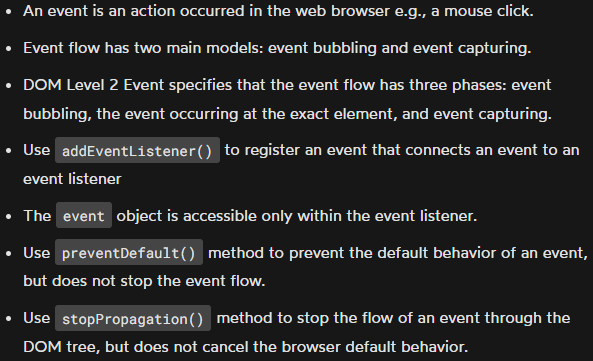
For example, when you click a link, the browser navigates you to the URL specified in the href attribute:

However, you can prevent this behavior by using the ‘preventDefault()’ method of the ‘event’ object:

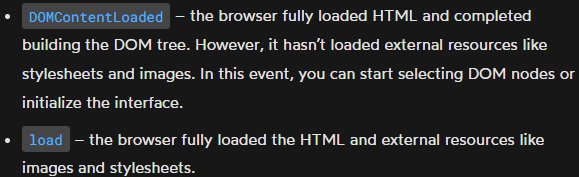
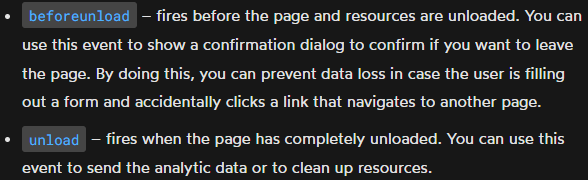
* stopPropagation() : stops the flow of an event through the DOM tree. However, it does not stop the browser’s default behavior. Example:

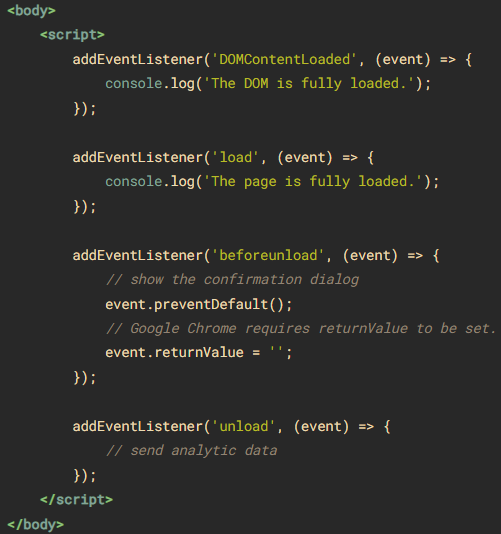
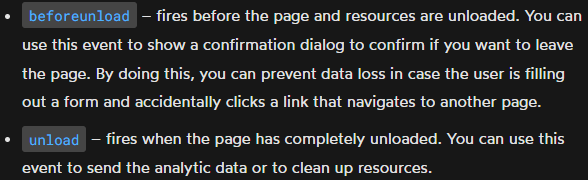
Without the ‘stopPropagation()’ method, you would see two messages on the Console window.

However, the ‘click’ event never reaches the ‘body’ because the ‘stopPropagation()’ was called on the ‘click’ event handler of the button.

Summary D & E:

1. **Page Load Events**

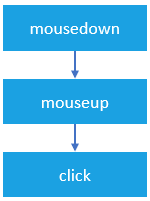
* When you open a page, the following events occur in sequence
* When you leave the page, the following events fire(occur) in sequence:

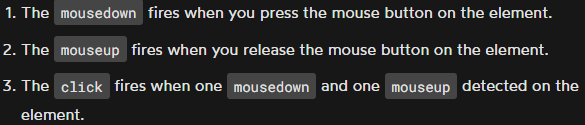
The following example illustrates how to handle the page load events:

1. **Mouse Events** (<https://www.javascripttutorial.net/javascript-dom/javascript-mouse-events/>)

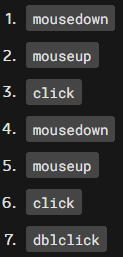
DOM Level 3 events define nine mouse events.

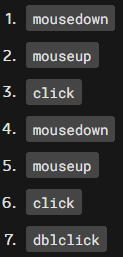
* mousedown, mouseup, and click

When you click an element, there are no less than three mouse events fire in the following sequence:



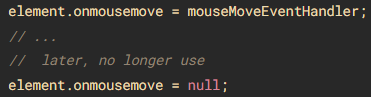
* dblclick

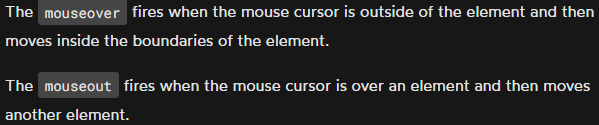
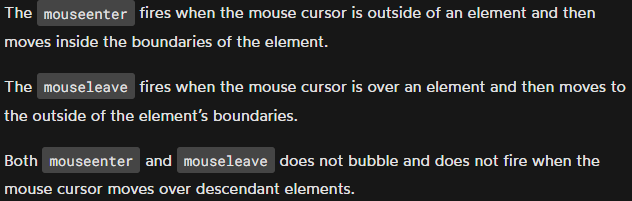
The ‘dblclick’ event fires when you double-click over an element. In practice, you rarely use the ‘dblclick’ event. The ‘dblclick’ event has four distinct events fired in the following order:



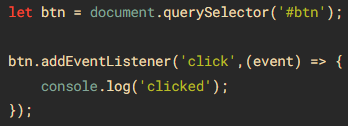
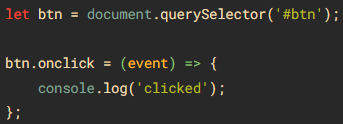
* mousemove

The ‘mousemove’ event fires repeatedly whenever you move the mouse cursor around an element. This ‘mousemove’ event fires many times per second as the mouse is moved around, even if it is just by one pixel. (This may lead to a performance issue if the event handler function is complex).

 To avoid the performance issue, it is a good practice to add ‘mousemove’ event handler only when you need it and remove it as soon as it is no longer needed, like this:

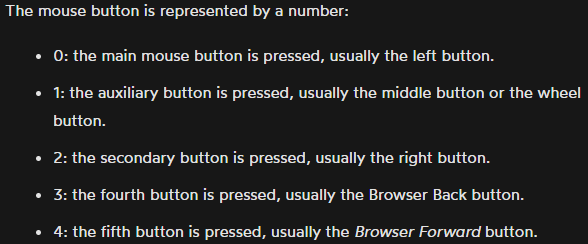
* mouseover/mouseout
* mouseenter/mouseleave

Registeing mouse event handlers:

* Recommended
* Okay
* In old system, you may find

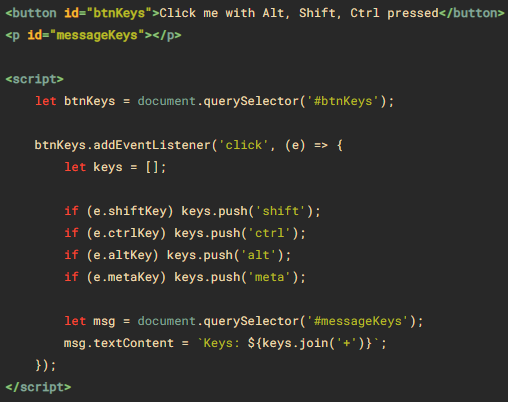
**Detecting mouse buttons**

The ‘event’ object passed to the mouse event handler has a property called ‘button’ that indicates which mouse button was pressed on the mouse to trigger the event.

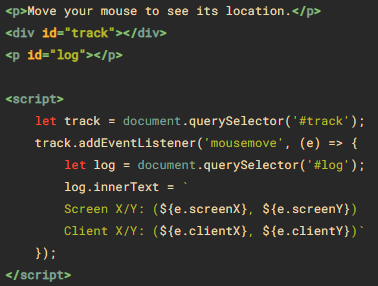
Example & Demo:

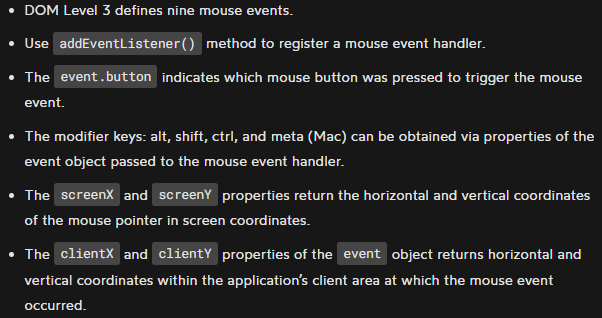
**Modifier keyboard events**

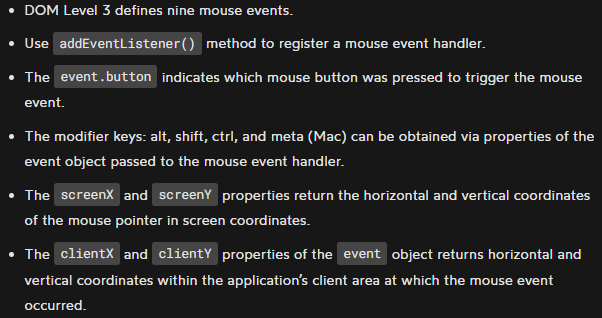
When you click an element, you may press one or more modifier keys: Shift, Ctrl, Alt, and Meta(windows key/command key on apple keyboard).

To detect if these modifier keys have been pressed, you can use the event object(‘true’ if the key is being held down or ‘false’ if the key is not pressed.) passed to the mouse event handler.

**Screen Coordinates Events**

* screenX dan screenY : memberikan koordinat X dan Y posisi relatifterhadap seluruh layar perangkat.
* clientX dan clientY : memberikan koordinat X dan Y posisi mouse relatif terhadap jendela aplikasi atau elemen HTML di mana event tersebut terjadi.

**Summary**

****

1. **Keyboard Events** (<https://www.javascripttutorial.net/javascript-dom/javascript-keyboard-events/>)
2. **Event Delegation** (<https://www.javascripttutorial.net/javascript-dom/javascript-event-delegation/>) (bisa juga seperti page 8 bagian atas di file ini)
3. **dispatchEvent** (<https://www.javascripttutorial.net/javascript-dom/javascript-dispatchevent/>)

dispatchEvent() digunakan untuk mensimulasikan klik pada button secara programatik (dalam kode), seolah-olah pengguna mengklik button tersebut.

1. **Custom Events** (<https://www.geeksforgeeks.org/javascript-custom-events/>)

Just supplement(not mandatory): <https://www.javascripttutorial.net/javascript-dom/javascript-custom-events/>