1. <script src="https://d3js.org/d3.v3.min.js"></script>
2. <script src="https://mauriciopoppe.github.io/function-plot/js/function-plot.js"></script>

We will examine this soon in a later section of the course, but, in brief, it means that we will use the d3js plotting library that is apparently located at [https://d3js.org](https://d3js.org/) (try to visit the site), and another from github (a famous repository for open source contributions), and located in the github account of a person named "mauriciopoppe", the repository is named "function plot". A rapid search will give [this URL](http://mauriciopoppe.github.io/function-plot/) as the home page of the "function plot JavaScript library

Logical operators

The logical operators are as follows:

&& (AND)

usage example : if ((x > 0) && (x < 10)) {

console.log('x is strictly positive and less than 10');

}

|| (OR)

usage example : if ((x > 0) || (x == -5)) {

console.log('x is positive or equal to -5');

}

! (NOT)

usage example : if (!(x > 0)) {

console.log('x is not positive (x is less or equal to 0');

}

&&, || operators are binary, ! is unary.

1. 1 == 1 ;
2. //true
4. 1 == 2 ;
5. //false
7. /\* Here, the interpreter will try to convert the string ‘1’
8. into a number before doing the comparison \*/
10. 1 == '1';
11. //true :

14. //with strict equal, no conversion:
16. 1 === 1;
17. //true
18. 1 === '1';
19. //false
20. **Mouse events**
21. **Event types related to mouse** **click** **dblclick**

|  |  |
| --- | --- |
| click | The event occurs when the user clicks on an element (presses a button and releases it) |
| dblclick | The event occurs when the user double-clicks on an element |
| mousedown | The event occurs when the user presses a mouse button |
| mouseup | The event occurs when a user releases a mouse button over an element |
| mousemove | The event occurs when the pointer is moving while it is over an element |
| mouseenter | The event occurs when the pointer is moved onto an element |
| mouseleave | The event occurs when the pointer is moved out of an element |
| mouseover | The event occurs when the pointer is moved onto an element, or onto one of its children |
| contextmenu | The event occurs when the user right-clicks on an element to open a context menu |

1. **MouseEvent properties**

|  |  |
| --- | --- |
| button | Returns which mouse button was pressed when the mouse event was triggered |
| clientX and clientY | Returns the coordinates of the mouse pointer, relative to the element coordinate system that triggered the event. **If you click in the left top corner the value will always be (0,0) independent of scroll position, these coordinates are relative to the VIEWPORT (the visible part of the document page)** |
| pageX and pageY | Returns the coordinates of the mouse pointer, relative to the document, when the mouse event was triggered. **They are relative to the complete document/page, and will always be relative to the very beginning of the document/page, even if the top of the page is not visible because you've scrolled down. They will change when the page scrolls and the mouse does not move!** |
| screenX and screenY | Returns the coordinates of the mouse pointer, relative to the screen, when an event was triggered. |
| altKey, ctrlKey, shiftKey | Returns whether the "alt, ctrl and shif" key was pressed when an event was triggered |
| detail | Returns a number that indicates how many times the mouse was clicked |

Any CSS  selector can be passed as a parameter for these methods.

* + While **querySelector(selector)** will return the first element in the DOM that matches the selector (and you will be able to work with it directly),
  + **querySelectorAll(selector) returns a collection of HTML elements corresponding to all elements matching the selector**. To process the results, it will be necessary to loop over each of the elements in the collection.