Zooming

Reusability

* (function(){})() // this method is used in packages, to run an anonyms function on definition

(function(window) {

let defineLibrary = () => ({

init : function(galleryId) {

console.log('setup gallery')

}

})

if(typeof(vanillaZoom) == 'undefined') {

window.vanillaZoom = defineLibrary();

} else {

console.log('library already defined.')

}

})(window)

* In Html file

vanillaZoom.init('#my-gallery')

* The if statement is used to create only one vanillaZoom. for example, when you have two script sources by mistake from this file, it only creates one instance. <script src=""></script>
* Window.vanillaZoom // creates a global variable in the widow and holds a function call, which means that it contains the returned value of that function.
* defineLibrary function returns an object. To not confuse it with a block, we wrap it in parenthesis. This object is assigned in vanillaZoom. The object has some functions and variables.

Initial styling

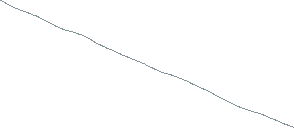
* We have to put a listener for every image to show it on click. But we can also put a listener on the container and use event.target. The problem is that everywhere we click, this will be triggered. For this problem, you can use classList.contains("className") to only perform it on specific elements you want.

How to zoom

* background-size('250%')
* clientX and clientY // on mouse move we can get these two values from the top left corner of screen.
* getBoundingClientRect() //The div having the background position from top left corner of screen.
* mouseEvent.clientX - el.getBoundingClientRect().top // get the position of mouse relative to background div.
* Use the result value in percent ( based on div width and height) in background-position= x% y%. This style gets two parameters x and y in percent.

Zoom style 2

* You can create two divs, one as zoom and one as zoom result.
* Use mouseenter and get the positon of mouse, and on mousemove, set the x and y. the boundingX and Y and clientX and clientY are related to the scroll (view port), you have to notice that. If the zoom div is getting out of the image, we should not let it cross the border.
* The zoomed amount(result) can be relative to the zoom div width and height.



Original size

zoomed size