

```

/*
=====
=====
    Load css links in document head
    =====
    I googled the link below and found this cool method to link CSS
    files dynamically into the HTML page
    https://www.geevent.keysforgeevent.keys.org/how-to-load-css-files-
    using-javascript/
*/

```

```

document.addEventListener("DOMContentLoaded", function(event) {
    const head = document.getElementsByTagName('HEAD')[0];

    const resetLink = document.createElement('link');
    resetLink.rel = 'stylesheet';
    resetLink.type = 'text/css';
    resetLink.href = 'https://cdnjs.cloudflare.com/ajax/libs/meyer-
    reset/2.0/reset.min.css';
    head.appendChild(resetLink);

    const link = document.createElement('link');
    link.rel = 'stylesheet';
    link.type = 'text/css';
    link.href = 'style.css';
    head.appendChild(link);
});

```

```

/*
=====
=====
    SET INSTRUCTIONS
    =====
    Here is the instructions part on top of the gallery to display the
    required KEYS to press to navigate the gallery
    _____
*/

```

```

const $instructions = document.getElementById('instructions');
$instructions.title = "<h2>Instructions</h2>"
$instructions.description = "<p>In order to navigate using Keyboard,
<strong> K </strong> key <strong>MUST</strong> be pressed to begin to
the first image, then <strong>LEFT</strong> and <strong>RIGHT</strong>
to move to the previous / next images. Press <strong>Enter</strong> to
select the image and view it in a modal mode. Press <strong>ESC</
strong> key to go back to the normal view.</p><p><strong>PLEASE DO NOT
USE</strong> <strong class='disabled'>Tab</strong> OR <strong
class='disabled'>SHIFT + Tab</strong> keys as I applied the functions
to the LEFT and RIGHT arrow keys only </p>"

```

```

$instructions.innerHTML = $instructions.title +
$instructions.description;

/*
=====
=====
    GALLERY BUILD
    =====
    1- Retrieve gallery and lightbox from HTML
    2- set the image address path
    3- create a galleryItems array
    4- Loop through the images
    5- Add zero before numbers less than ten
    6- Push dynamic tags to Array
    7- Include TABINDEX to conform with ARIA
    8- Insert galleryItems to gallery using JOIN
    9- Set an ARRAY for the items which will be used many times later
*/
const $gallery = document.getElementById('gallery');
const $lightbox = document.getElementById('lightbox');

const $closeBtn = document.querySelector('.closeBtn');
const $leftBtn = document.querySelector('.leftBtn');
const $rightBtn = document.querySelector('.rightBtn');

const imgAddress = "./img/";
const galleryItems = [];
const lightboxItems = [];

for(i=1; i<13; i++) {
    let j = i;
    if (j < 10) {
        j = `0${j}`;
    }

    galleryItems.push(`
        <li class="gallery-item" tabindex="${i}">
            
            <h2>img-${j}</h2>
        </li>
    `);

    lightboxItems.push(`
        <li class="lightbox-item">
            <div class="lightbox-header">
                <span class="closeBtn">&times;</span>
                <h2>img${j}</h2>
            </div>
    `);
}

```

```

        <div class="lightbox-content">
            
            <h4 class="caption">This is a bigger view of img${j}</
h4>
        </div>

        <div class="lightbox-footer">
            <span class="leftBtn">&lt;</span>
            <span class="rightBtn">&gt;</span>
        </div>
    </li>
`);
}

$gallery.innerHTML = galleryItems.join('');
const $galleryItems = document.querySelectorAll(".gallery-item");

$lightbox.innerHTML = lightboxItems.join('');
const $lightboxItems = document.querySelectorAll(".lightbox-item");

// let $lightbox = document.createElement('div');
// $lightbox.setAttribute("id", "lightbox");
// $lightbox.appendChild(document.createTextNode('top div'));
// let $container = document.getElementsByTagName('container');
// $container.innerHTML.appendChild($lightbox);

/*
=====
=====
    INIT SHOW / HIDE GALLERY & LIGHTBOX
    =====
    This is the initial SHOW / HIDE Gallery and lightbox control
*/

// $gallery.style.display = "none";
$lightbox.style.display = "none";

// $gallery.style.visibility = "hidden";
// $lightbox.style.visibility = "hidden";

/*
=====
=====
    GALLERY FUNCTIONS
    =====
    1- checkItem fn that resets the focus of all elements in the list
    then apply it on the new focused / selected list element
*/

```

```

// =====
// reset selection variables
let activeItem;
let activeLightboxItem;

let currentItem;
let currentLightboxItem;

let navStatus = false;
let lightboxOn = false;

// =====
// reset items focus
function resetItems () {
    for (const $galleryItem of $galleryItems) {
        $galleryItem.classList.remove("activeItem");
        $galleryItem.blur();
    }
}

// =====
// reset lightbox items focus
function resetLightboxItems () {
    for (const $lightboxItem of $lightboxItems) {
        $lightboxItem.classList.remove("largeImg");
        $lightboxItem.style.display = "none";
    }
}

// =====
// Check Gallery item
function checkItem(param) {
    console.log("|| START checkItem AND currentItem = " + currentItem
+ " || activeItem = " + activeItem);

    let flagType = typeof param;
    console.log("param => " + param + " | param.key => " + param.key +
" | flagType = " + flagType);

    if (flagType === 'number') {
        activeItem = param;

        navStatus = true;
        lightboxOn = true;
        // $lightbox.style.visibility = "visible";
        lightboxOpen();
    } else if (flagType === 'string' && !navStatus) {

```

```

        if (param === 'k' && !lightboxOn) {
            activeItem = 1;
            navStatus = true;
        }
    } else if (flagType === 'string' && navStatus) {
        if (param === 'next' && !lightboxOn) {

            if (currentItem < $galleryItems.length) {
                activeItem = activeItem + 1;
            } else {
                activeItem = 1;
            }
        } else if (param === 'tabNext' && !lightboxOn) {
            if (currentItem < $galleryItems.length) {
                // activeItem = activeItem + 1;
                activeItem = activeItem + 1;
            } else {
                activeItem = 1;
            }
        } else if (param === 'prev' && !lightboxOn) {
            if (currentItem > 1) {
                activeItem = activeItem - 1;
            } else {
                activeItem = $galleryItems.length;
            }
        } else if (param === 'tabPrev' && !lightboxOn) {
            if (currentItem > 1) {
                activeItem = activeItem - 1;
            } else {
                activeItem = $galleryItems.length;
            }
        } else if (param === 'enter' && !lightboxOn) {
            lightboxOn = true;
            // $lightbox.style.visibility = "visible";
            lightboxOpen();
            return;
        } else if (param === 'esc' && lightboxOn) {
            lightboxOn = false;
            // $lightbox.style.visibility = "hidden";
            lightboxClose();
            return;
        }
    }
}

console.log("activeItem => " + activeItem + " | TYPE = " + typeof
activeItem);

resetItems();

if (!lightboxOn) {
    $galleryItems[activeItem - 1].classList.add("activeItem");
}

```

```

        $galleryItems[activeItem -1].focus();
    }

    currentItem = activeItem;
    console.log("|| END checkItem AND currentItem = " + currentItem +
" || activeItem = " + activeItem);
}

// =====
// Open lightbox
function lightboxOpen() {
    lightboxOn = true;
    $lightbox.style.display = "block";
    console.log("LB open "+ activeItem);
    // $lightboxItems[activeItem].style.display = "block";
    // $lightboxItems[activeItem -1].classList.add("largeImg");

    for (const $lightboxItem of $lightboxItems) {
        $lightboxItem.classList.remove("largeImg");
        $lightboxItem.style.display = "none";
    }
    activeLightboxItem = activeItem;
    $lightboxItems[activeLightboxItem-1].style.display = "block";

    $lightboxItems[activeLightboxItem -1].classList.add("largeImg");

    currentLightboxItem = activeLightboxItem;
    // $lightbox.innerHTML =

}

// =====
// lightbox left
function lightboxLeft() {
    resetLightboxItems();

    if (currentLightboxItem > 1) {
        activeLightboxItem = activeLightboxItem - 1;
    } else {
        activeLightboxItem = $lightboxItems.length;
    }
    $lightboxItems[activeLightboxItem - 1].style.display = "block";
    $lightboxItems[activeLightboxItem - 1].classList.add("largeImg");
    currentLightboxItem = activeLightboxItem;
}

// =====

```

```
// lightbox left
function lightboxRight() {
    resetLightboxItems();

    if (currentLightboxItem < $lightboxItems.length) {
        activeLightboxItem = activeLightboxItem + 1;
    } else {
        activeLightboxItem = 1;
    }
    $lightboxItems[activeLightboxItem - 1].style.display = "block";
    $lightboxItems[activeLightboxItem - 1].classList.add("largeImg");
    currentLightboxItem = activeLightboxItem;
}

```

```
// =====
// Close lightbox
function lightboxClose() {
    lightboxOn = false;
    $lightbox.style.display = "none";
}

```

```
// =====
// Close If Outside Click
function outsideClick(e) {
    if (e.target == $lightbox) {
        lightboxOn = false;
        $lightbox.style.display = "none";
    }
}

```

```
/*
=====
=====
EVENT LISTENERS
=====
1- Add an MOUSE CLICK Event Listener to the gallery
    a- If the element exists
2- Add an KEY Event Listener to the document
*/

```

```
// =====
// GALLERY
$gallery.addEventListener('click', function (event) {
    const $targetItem = event.target.closest('.gallery-item');

    if($targetItem) {
        console.log(">> checkItem >>" + $targetItem.tabIndex);
        checkItem($targetItem.tabIndex);
    }
}

```

```

})

// =====
// LIGHTBOX
$lightbox.addEventListener('click', function (event) {
    const $lightboxCloseBtn = event.target.closest('.closeBtn');
    const $lightboxLeftBtn = event.target.closest('.leftBtn');
    const $lightboxRightBtn = event.target.closest('.rightBtn');

    if($lightboxCloseBtn) lightboxClose();
    else if($lightboxLeftBtn) lightboxLeft();
    else if($lightboxRightBtn) lightboxRight();
})

document.addEventListener('click', outsideClick);

// =====
// KEYBOARD LISTENER
document.addEventListener('keydown', function (event) {
    // if (event.key === '1') checkItem(1);
    if (event.key === 'k') checkItem('k');

    if (navStatus === true && !lightboxOn) {
        if (event.key === 'ArrowRight')      checkItem('next');
        else if (event.key === 'ArrowLeft')  checkItem('prev');
        else if (event.key === 'Tab') {
            if(event.shiftKey) {
                checkItem('tabPrev');
                console.log("tabPrev");
            }
            else {
                checkItem('tabNext');
                console.log("tabNext");
            }
        }
        else if (event.key === 'Enter')      checkItem('enter');
        else if (event.key === 'Escape')     checkItem('esc');
    } else if (lightboxOn === true) {
        // return false;

        // TIME HAS NOT HELPED ME FINISH THIS PART
        // =====
        // THIS IS WHAT WAS BROKEN IN CODE
        // =====

        if (event.key === 'ArrowRight')      lightboxRight();
        else if (event.key === 'ArrowLeft')  lightboxLeft();
        else if (event.key === 'Escape')     lightboxClose('esc');
    }
});

```



```

    } else {
        console.log("That is not an option but " + event.key);
        // return false;
    }
});

```

```

/*
=====
=====

```

REFERENCE HTML SAMPLE TREE

The following is the tree that I'm building using JS

```

<div id="container">
    <div class="instructions">.....</div>
    <ul id="gallery">
        <li class="gallery-item">
            
            <h2>img-01</h2>
        </li>

        <li>....</li>
    </ul>

    <ul id="lightbox" class="lightbox">

        <li class="lightbox-item">
            <div class="lightbox-header">
                <span class="close">&times;</span>
                <h2>lightbox Header</h2>
            </div>

            <div class="lightbox-content">
                

                <h4 class="caption">A large imag</h4>

            </div>

            <div class="lightbox-footer">
                <span class="left">&lt;</span>
                <span class="right">&gt;</span>
            </div>
        </li>
        <li>....</li>
    </ul>

```

`*/` `</div>`