# https://eslint.org/docs/latest/rules/array-callback-return

# [usage of spread syntax code with map function](https://stackoverflow.com/questions/73965886/usage-of-spread-syntax-code-with-map-function)

# The point of .map() and .filter() is to iterate over the array and produce a new array. When a developer reads some code and sees .map() or .filter() being used, they expect that there should be a returned array. When they don't see it being done that way, they will be confused, will initially feel like they don't understand the code. If I were doing a code review on code like this, I would not approve of code like this.

1. Instead, use an iteration method that does not produce an output array such as for/of, a regular for loop or .forEach().

# Using .map() or .filter() without returning anything from the callback is, just as the linter says, "probably a programming mistake" because that is not how those functions are designed to be used and there are appropriate alternatives when you don't want a returned array.

# So, if you're just trying to do an iteration without creating any resulting array, use for/of or .forEach() or some other iteration scheme that isn't specifically designed to create an output array that you don't want.

forEach: This iterates over a list and applies some operation with side effects to each list member (example: saving every list item to the database) and does not return anything.

map: This iterates over a list, transforms each member of that list, and returns another list of the same size with the transformed members (example: transforming list of strings to uppercase). It does not mutate the array on which it is called (although the callback function may do so).

References

# https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/forEach

# https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/maphttps://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\_Objects/Array/map

ESLint is correct - if you don't have a return statement in a map or filter callback then you are indeed most likely misusing it. They *rely* on the return value to function. Otherwise you are just looping over the array for no real reason. In that case, you should be using .forEach

# https://css-tricks.com/snippets/javascript/loop-queryselectorall-matches/

for (const button of document.querySelectorAll('#wrapper button')) {

button.addEventListener('click', calculate);

}

# [The docs](https://developer.mozilla.org/en-US/docs/Web/API/NodeList)- are silent on this (apart from mentioning that for … of works), but [the NodeList interface](https://dom.spec.whatwg.org/#interface-nodelist) in the spec is marked as [iterable](https://webidl.spec.whatwg.org/#idl-iterable), which [spread syntax](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Spread_syntax) is making use of.

# ---------------------------------------

# [Debugging vanilla javascript promises in 2022 [closed]](https://stackoverflow.com/questions/73940437/debugging-vanilla-javascript-promises-in-2022)

# https://stackoverflow.com/questions/25827234/how-to-debug-javascript-promises

# <https://stackoverflow.com/questions/47216353/chrome-debugger-promises-dont-resolve-while-paused>

debugger;

console.log('start')

async function withAwait() {

await 0;

console.log('with await')

return Promise.resolve(100);

}

withAwait()

console.log('end')

# answer:

# withAwait().then((val) => { console.log("withAwait() promise resolved", val)});

# [Blinking cursor forever during tailwind install process](https://stackoverflow.com/questions/73846771/blinking-cursor-forever-during-tailwind-install-process)

[Ask Question](https://stackoverflow.com/questions/ask)

I am unable to search on SO hence detailing it here, this can be a duplicate question.Refer the Step 4:'Start the Tailwind CLI build process' [TailwindInstallation](https://tailwindcss.com/docs/installation) the question is once I run the MSDOS command for CSS processing: npx tailwindcss -i input.css -o output.css --watch , the rebuilding and done is OK with no errors (refer attached command image) but after that my cursor keeps blinking forever, so I cant issue any more DOS commands so then I need to hit CTRL C (say couple of times) to get a message 'Terminate Batch Job(Y/N)' and I choose Y and then its normal command line prompt.This is an irritant.What am I doing wrong , please advise

That command ends with --watch, so the process will keep running monitoring changes to the files and constantly update the Tailwind css output.

The idea is to leave it running while you are working on the site. If you still need the terminal for something else then open another terminal.

# [Error while executing javascript promises](https://stackoverflow.com/questions/73924489/error-while-executing-javascript-promises)

I am getting console error as:**Uncaught (in promise) TypeError: Failed to execute 'createObjectURL' on 'URL': Overload resolution failed.**

I wish to know where I am going wrong in my concepts..Thanks

const myImage = document.querySelector("img");

const myRequest = new Request('elephant.jpg');

// console.log(myRequest)

fetch(myRequest)

.then((response) => {

if (!response.ok) {

throw new Error(`HTTP error! Status: ${response.status}`)

}

else {

console.log(`no HTTP error, elephant.jpg file is present,Status: ${response.status}`)

myBlob=response.blob();

const objectURL = URL.createObjectURL(myBlob);

myImage.src = URL.createObjectURL(response)

}

})

<h1>Fetch Request example</h1>

<img src="" />

const myImage = document.querySelector("img");

const myRequest = new Request('elephant.jpg');

// console.log(myRequest)

fetch(myRequest)

.then((response) => {

if (!response.ok) {

throw new Error(`HTTP error! Status: ${response.status}`)

}

else {

console.log(`no HTTP error, elephant.jpg file is present,Status: ${response.status}`)

myBlob=response.blob();

const objectURL = URL.createObjectURL(myBlob);

myImage.src = URL.createObjectURL(response)

}

})

<h1>Fetch Request example</h1>

<img src="" />

const myImage = document.querySelector("img");

const myRequest = new Request("elephant.jpeg");

fetch(myRequest).then(async (response) => {

if (!response.ok) {

throw new Error(`HTTP error! Status: ${response.status}`);

} else {

console.log(

`no HTTP error, elephant.jpg file is present,Status: ${response.status}`

);

myBlob = await response.blob() // ensure this is defined in your file

const objectURL = window.URL.createObjectURL(myBlob);

myImage.src = window.URL.createObjectURL(myBlob);

}

});

# [How to confirm for JavaScript asynchronous calls? [closed]](https://stackoverflow.com/questions/73899801/how-to-confirm-for-javascript-asynchronous-calls)

document.addEventListener('DOMContentLoaded', () => {

flexStuff();

async function flexStuff() {

let tr = '{"https://create-react-app.dev/docs/getting-started":"React Start",\

"https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX":"ajax",\

"https://developer.mozilla.org/en-US/docs/Web/CSS/inheritance#inherited\_properties":"CSS",\

"https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Building\_blocks/Events#using\_addeventlistener":"event listeners"\

}';

const obj121 = JSON.parse(tr);

for (let key in obj121) {

d1 = document.createElement('div');

d1.setAttribute('id', obj121[key]);

d1.innerHTML = `<a href=${key}>${obj121[key]}</a>`;

document.querySelector('.my\_flexx').appendChild(d1);

}

await 2022;

}

});

# an async function without an await expression will run synchronously. If there is an await expression inside the function body, however, the async function will always complete asynchronously.

# Yes, everything in the async function will run synchronously until await is encountered.

# variant 1

async function withoutAwait() {

console.log('without await')

}

async function withAwait() {

await 0

console.log('with await')

}

console.log('start')

withoutAwait()

withAwait()

console.log('end')

# variant 2

let a = 0

async function changeA() {

a = 20

await 0

a = 30

}

console.log('start', a)

changeA().then(() => console.log('after change', a))

console.log('end', a

# variant 3

const results = []

async function f1() {

for (let i = 0; i < 10; i += 1) {

await results.push('f1')

await 0

}

}

async function f2() {

for (let i = 0; i < 10; i += 1) {

await results.push('f2')

}

}

f1()

f2()

setTimeout(() => {

console.log(results)

}, 100)

# Difference between visibility and display: <!--there u go:- To have an element take up the space that it would normally take, but without actually rendering anything, use the visibility property instead. display:block

# The element generates a block element box, generating line breaks both before and after the element when in the normal flow.

# Display:inline

# The element generates one or more inline element boxes that do not generate line breaks before or after themselves. In normal flow, the next element will be on the same line if there is space

# display:none

# Turns off the display of an element so that it has no effect on layout (the document is rendered as though the element did not exist). All descendant elements also have their display turned off.

# <body>

# <a class="a">a</a>

# <a class="b">b</a>

# <a class="c">c</a>

# <style>

# Makes the b element disappear so adds 2 space between a and c

# .b{display:inline;visibility:hidden;}

# 

# Makes the b element GO OFF so adds 1 space between a and c

# .b{display:none;visibility:hidden or visibility:visible;}

# 

# </style></body>

# CSS thinks in parent child way:

# <aside>

# <p>The Rough-skinned Newt defends itself with a deadly neurotoxin.</p>

# </aside>

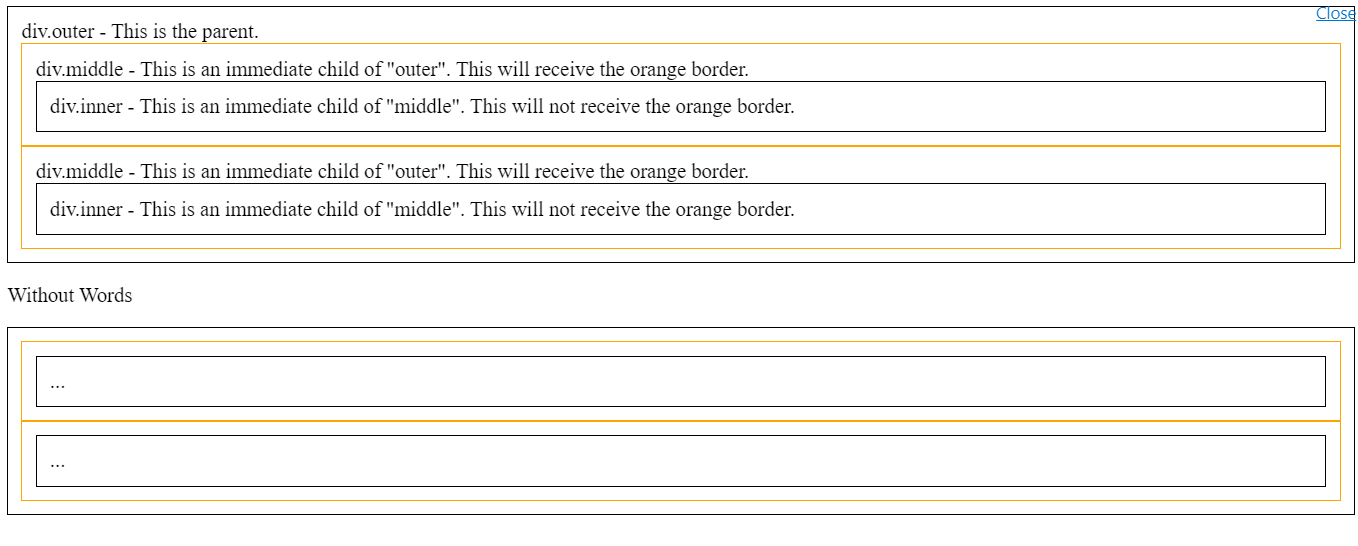
# So,

# ASIDE IS PARENT

# P IS CHILD

# > CSS Selector

# 1st variant



.css file

div {

border: 1px solid black;

padding: 10px;

}

.outer > div {

border: 1px solid orange;

}

.html file

<div class='outer'>

div.outer - This is the parent.

<div class="middle">

div.middle - This is an immediate child of "outer". This will receive the orange border.

<div class="inner">div.inner - This is an immediate child of "middle". This will not receive the orange border.</div>

</div>

<div class="middle">

div.middle - This is an immediate child of "outer". This will receive the orange border.

<div class="inner">div.inner - This is an immediate child of "middle". This will not receive the orange border.</div>

</div>

</div>

<p>Without Words</p>

<div class='outer'>

<div class="middle">

<div class="inner">...</div>

</div>

<div class="middle">

<div class="inner">...</div>

</div>

</div>

# 2nd variant

# 

.css file

div { border: 1px solid black; margin-bottom: 10px; }

.a b { color: red; } /\* every John is red \*/

.b > b { color: blue; } /\* Only John 3 and John 4 are blue \*/

# .html file Parent is <div class="a">

# Child is <p>

# Grandchild is <b>

<div class="a">

<p><b>John 1</b></p>

<p><b>John 2</b></p>

<b>John 3</b>

<b>John 4</b>

</div>

# Next….

# Parent is <div class="b">

# Child is <b>

<div class="b">

<p><b>John 1</b></p>

<p><b>John 2</b></p>

<b>John 3</b>

<b>John 4</b>

</div>

# [innerhtml throws error while displaying html link](https://stackoverflow.com/questions/73783279/innerhtml-throws-error-while-displaying-html-link):

<https://stackoverflow.com/questions/10055773/double-quote-in-javascript-string>

function toggleNav1()

{

//this below statement executes

document.getElementById('mobile').innerHTML="www.cnn.com"

//but this below statement is an error:

document.getElementById('mobile').innerHTML="<a href="https://www.google.com">google</a>"

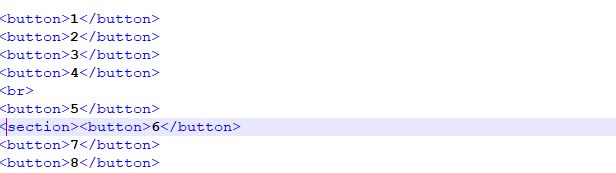
}

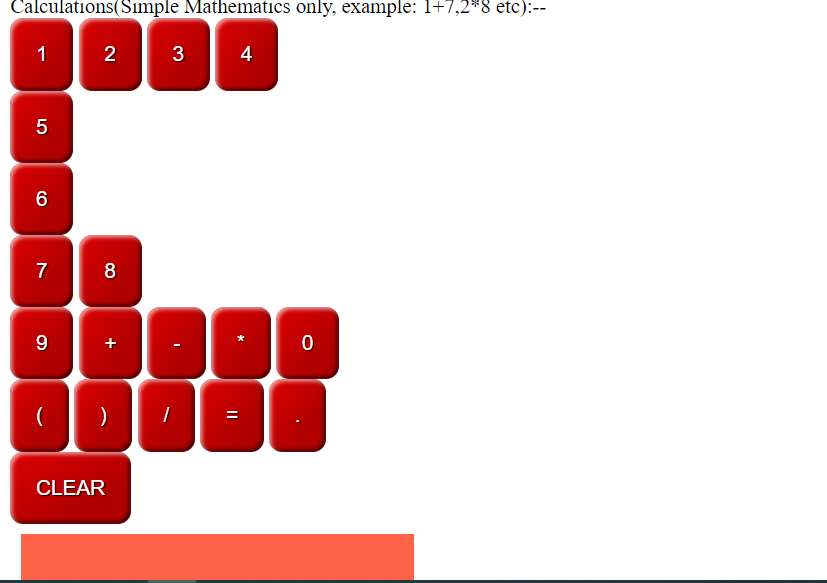
So Use:

document.getElementById('mobile').innerHTML="<a href=\"https://www.google.com\">google</a>"

document.getElementById('mobile').innerHTML='<a href="https://www.google.com">abcle</a>'

Try to use sections instead of div for HTML line breakups(example):





Never use document.write instead use textContent:

// first create an array

let resultsArray = [];

// fill the array with numbers 0 - 199

for (let i=0; i < 200; i += 1) {

resultsArray.push(i);

}

// print it to pre#log

document.querySelector("#log").textContent = resultsArray.join(" ");

# [Clicking on an empty area causes events to be generated](https://stackoverflow.com/questions/73758131/clicking-on-an-empty-area-causes-events-to-be-generated)

I am currently implementing a calculator using HTML,CSS,Javascript.Chrome v 105 used,javascript event bubbling has been used.The calculator works ok if I click on the buttons but the problem occurs if I click a little bit right on an empty area(about say 1 or 2 cm in the empty area to the right of the numbers 2 or 8 or 0 or the . symbol) then I see multiple events being generated(in the console.log and UI) and the corresponding button values in the div being captured so how do I rectify this issue, please help me understand as I am not so experienced in javascript.

That's because you set the listener to the wrapper div. set the listener only to every button who is a child of #wrapper:

works but please explain the ... dots and the [] operators you have used in : [...document.querySelectorAll('#wrapper button')].map((button)=> button.addEventListener('click',Calculat));

– [John Ray](https://stackoverflow.com/users/20021858/john-ray)

* @JohnRay they used to convert list the into array, changed to forEach .. //listen to only button events(other than the CLR)

document.querySelectorAll('#wrapper button').forEach(button=> button.addEventListener('click',Calculat));

What is the difference between class and classname in javascript?

Your title question.

## **Answer 1:**

Class is an attribute in an html element <span class='classy'></span>

While, on the other hand, .className is a property that can by called on an element to get/set its class.

var element = document.createElement('span');

element.className = 'classy'

// element is <span class='classy'></span>

Setting the class can also be accomplished with .getAttribute('class') and .setAttribute('class', 'classy'). We change manipulate classes so often, however, that it merited its own .className method.

# [Unable to render react](https://stackoverflow.com/questions/73514308/unable-to-render-react)

[Ask Question](https://stackoverflow.com/questions/ask)

<script>

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<h1>Hello, world!</h1>);

</script>

<div id="root">

</div>

I am running the html snippet but getting a console error message , this code is from <https://reactjs.org/docs/hello-world.html> ,it runs fine on codepen but not my local browser, please help as i am new to react.js

That snippet won't run in your browser's console as-is. In the code pen go to settings > JS > Add External Scripts/Pens. You'll see react and react-dom are dependencies for the code to run.

This tutorial for create-react-app will help you get everything installed correctly so you can try the hello world example - <https://create-react-app.dev/docs/getting-started/>

Aswer2:

Here's the full setup in order for this example code to run:

<!-- INCLUDE REACT LIB -->

<script src="https://unpkg.com/react@18/umd/react.production.min.js" crossorigin></script>

<!-- INCLUDE REACTDOM LIB -->

<script src="https://unpkg.com/react-dom@18/umd/react-dom.production.min.js" crossorigin></script>

<!-- INCLUDE BABEL COMPILER LIB FOR JSX -->

<script src="https://unpkg.com/babel-standalone@6/babel.min.js"></script>

<div id="root"></div>

<!-- Change the type to text/babel in order for the Babel compiler to detect and compile the JSX syntax -->

<script type="text/babel">

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<h1>Hello, world!</h1>);

</script>

You can find the CDN links used in this example for the React, ReactDOM and Babel libraries on the [link](https://create-react-app.dev/docs/getting-started/) that @abo mentioned above.

Happy Hacking and welcome to the wonderful world of React! ;)

----------------------------------------------------------------

# [Image not loading in html page](https://stackoverflow.com/questions/73268366/image-not-loading-in-html-page)

[Ask Question](https://stackoverflow.com/questions/ask)

<https://jsfiddle.net/jonathan668/vwc1hd8z/6/>

I am trying to flip images whenever user clicks the next button on the web page

Refer https://jsfiddle.net/jonathan668/vwc1hd8z/6/

Issue is My image urls generated via javascript are not loading to html

You don't need to create a new Image object to change the source. Remove line 8 of your JavaScript image5 = new Image(); and it should work perfectly.

# [I need help to overwrite images on html page upon clicking a button [closed]](https://stackoverflow.com/questions/73303209/i-need-help-to-overwrite-images-on-html-page-upon-clicking-a-button)

I have stored my images in an array called arr I have put a button on the html page called 'Next' so that whenever the user clicks the next button,the next image is picked up and overwrites the previous one the issue is the image on html page does not get overwritten by the next image( stored in the array),so how do I accomplish this

please help me as I am new to javascipt.

<body>

<script>

var arr = ["apple.jpg","kiwi.jpg", "pineapple.jpg","mango.jpg"];

function a()

{

image5 = document.getElementById("image5");

console.log(arr[0]);

image5=new Image();

image5.src = arr[0];

console.log("file is " + image5.src);

arr.splice(0,1);

}

</script>

<img src="apple.jpg" id="image5" style="width:300px;height:300px;">

<button onClick="a();">Next</button>

</body>

Ans1:

Why did you add this line: image5=new Image(); ? You want to change what #image5 shows, but you then overwrite the reference to that element (stored in image5) with something new (new Image()) and change the .src property of that new object.

– [Andreas](https://stackoverflow.com/users/402037/andreas)

[Aug 10 at 8:40](https://stackoverflow.com/questions/73303209/i-need-help-to-overwrite-images-on-html-page-upon-clicking-a-button#comment129455284_73303209)

* Sorry my typo ; now i removed it. i can see the file path being created in console log but image on html page is not getting overwritten.pls help me

– [jonathan](https://stackoverflow.com/users/19542576/jonathan)

[Aug 10 at 8:44](https://stackoverflow.com/questions/73303209/i-need-help-to-overwrite-images-on-html-page-upon-clicking-a-button#comment129455377_73303209)

* Just remove the line with that new Image() call

– [Andreas](https://stackoverflow.com/users/402037/andreas)

[Aug 10 at 8:46](https://stackoverflow.com/questions/73303209/i-need-help-to-overwrite-images-on-html-page-upon-clicking-a-button#comment129455414_73303209)

* Andreas Thanks a lot for your help..works now

– [jonathan](https://stackoverflow.com/users/19542576/jonathan)

[Aug 10 at 8:48](https://stackoverflow.com/questions/73303209/i-need-help-to-overwrite-images-on-html-page-upon-clicking-a-button#comment129455456_73303209)

Ans2:

I pretty much just cleaned up the code and had a variable that is the indicator to which picture comes up on the array and its loops. Also, I made the script tag defer so the tag loads after the whole page is so the function runs

<body>

<script defer>

var arr = ["apple.jpg","kiwi.jpg", "pineapple.jpg","mango.jpg"];

var image5 = document.getElementById("image5");

var i = 0;

function a(){

if(i>=4){i=0;}

image5.src = arr[i];

console.log("file is " + image5.src);

i++;

}

</script>

<img src="apple.jpg" id="image5" style="width:300px;height:300px;">

<button onclick="a()">Next</button>

</body>

# [Can I remove the duplicacy of javascript eventlisteners(each of the listeners listens to either a mathematical operator or a variable between 0 and 9](https://stackoverflow.com/questions/73520976/can-i-remove-the-duplicacy-of-javascript-eventlistenerseach-of-the-listeners-li)

let buffer = [];

let arrayOfNumbers=[];

document.addEventListener('DOMContentLoaded', () => {

document.getElementById('buffer1').style.color = 'purple';

document.getElementById('buffer1').style.backgroundColor = 'tomato';

document.getElementById('buffer1').style.fontWeight = 'bold';

document.getElementById('one').addEventListener('click', () => {

Calculat(document.getElementById('one').textContent);});

document.getElementById('two').addEventListener('click', () => {

Calculat(document.getElementById('two').textContent);});

document.getElementById('three').addEventListener('click', () => {

Calculat(document.getElementById('three').textContent);});

document.getElementById('four').addEventListener('click', () => {

Calculat(document.getElementById('four').textContent);});

document.getElementById('five').addEventListener('click', () => {

Calculat(document.getElementById('five').textContent);});

document.getElementById('six').addEventListener('click', () => {

Calculat(document.getElementById('six').textContent);});

document.getElementById('seven').addEventListener('click', () => {

Calculat(document.getElementById('seven').textContent);});

document.getElementById('eight').addEventListener('click', () => {

Calculat(document.getElementById('eight').textContent);});

document.getElementById('nine').addEventListener('click', () => {

Calculat(document.getElementById('nine').textContent);});

document.getElementById('zero').addEventListener('click', () => {

Calculat(document.getElementById('zero').textContent);});

document.getElementById('dot').addEventListener('click', () => {

Calculat(document.getElementById('dot').textContent);});

//operators

document.getElementById('plus').addEventListener('click', () => {

Calculat(document.getElementById('plus').textContent);});

document.getElementById('sub').addEventListener('click', () => {

Calculat(document.getElementById('sub').textContent);});

document.getElementById('multiply').addEventListener('click', () => {

Calculat(document.getElementById('multiply').textContent);});

document.getElementById('divide').addEventListener('click', () => {

Calculat(document.getElementById('divide').textContent);});

document.getElementById('lbrace').addEventListener('click', () => {

Calculat(document.getElementById('lbrace').textContent);});

document.getElementById('rbrace').addEventListener('click', () => {

Calculat(document.getElementById('rbrace').textContent);});

document.getElementById('equal').addEventListener('click', () => {

Calculat(document.getElementById('equal').textContent);});

document.getElementById('clear').addEventListener('click', () => {

buffer = [];

arrayOfNumbers=[];

document.getElementById('result').textContent=('');

document.getElementById('buffer1').textContent=('');

});

});

function Calculat(z) {

document.getElementById('buffer1').append(z);

if (z =='='){

console.log("Mathematical Array : " + arrayOfStrings);

console.log("Result is : " + eval(arrayOfStrings));

document.getElementById('result').style.color = 'red';

document.getElementById('result').style.backgroundColor = 'lightgrey';

document.getElementById('result').style.fontWeight = 'bold';

let result = Function("return " + arrayOfStrings)();

console.log(result)

document.getElementById('result').textContent=result;

return false;

}

buffer.push(z);

arrayOfStrings = buffer.join("");

}

.flex-container {

display: flex;

background-color: Blue;

justify-content: left;

}

.flex-container > div {

background-color: deepskyblue;

margin: 8px;

padding: 20px;

font-size: 30px;

}

<div class="flex-container">

<div>Array:</div>

<div id="buffer1">&nbsp;&nbsp;&nbsp;&nbsp;</div>

</div>

<div class="flex-container">

<div id="one">1</div>

<div id="two">2</div>

<div id="three">3</div>

<div id="four">4</div>

<div id="zero">0</div>

</div>

<div class="flex-container">

<div id="five">5</div>

<div id="six">6</div>

<div id="seven">7</div>

<div id="eight">8</div>

<div id="nine">9</div>

</div>

<div class="flex-container">

<div id="plus">+</div>

<div id="sub">-</div>

<div id="multiply">\*</div>

<div id="divide">/</div>

<div id="equal">=</div>

</div>

<div class="flex-container">

<div id="dot">.</div>

<div id="clear">CLR</div>

<div id="lbrace">(</div>

<div id="rbrace">)</div>

</div>

<div class="flex-container">

<div>Result:</div>

<div id="result"></div>

</div>

Ans:

1

you can use [bubbling](https://javascript.info/bubbling-and-capturing) here, add some div wrapper to html like

<div id="wrapper">

<!-- current html -->

</div>

then attach event listener to it and remove all listeners that is inside your wrapper

document.getElementById('wrapper').addEventListener('click', (e) => Calculat(e.target.textContent))

and final code will be following

document.addEventListener('DOMContentLoaded', () => {

document.getElementById('buffer1').style.color = 'purple';

document.getElementById('buffer1').style.backgroundColor = 'tomato';

document.getElementById('buffer1').style.fontWeight = 'bold';

document.getElementById('wrapper').addEventListener('click', (e) => Calculat(e.target.textContent))

document.getElementById('clear').addEventListener('click', () => {

buffer = [];

arrayOfNumbers=[];

document.getElementById('result').textContent=('');

document.getElementById('buffer1').textContent=('');

});

});

# [using a div class doesnt render html page](https://stackoverflow.com/questions/73592932/using-a-div-class-doesnt-render-html-page)

[Ask Question](https://stackoverflow.com/questions/ask)

let x=[];

const json1 = '{"https://www.w3schools.com":"w3schools"}';

const obj1 = JSON.parse(json1);

ol = document.createElement('ol');

ser.appendChild(ol);

for (var key in obj1){

x= document.createElement('li');

x.innerHTML="<a href=" + key + ">" + obj1[key]+"</a>";

ol.appendChild(x);

}

.ser{

color: red;

}

<div class="ser">welcome</div>

Due to an IE quirk, adding an id to an element makes it exist for your JS code to reference directly. This was kept for legacy reasons but in theory you would have to declare the variable first, using for instance const ser = document.getElementById('ser');. This doesn't work with classes because the whole point of classes is to apply them to multiple elements.

Answer1:

tried and it worked both ways : document.getElementById('ser').appendChild(ol); document.querySelector('.ser').appendChild(ol);

this worked too; so got conceptually your 3rd variant meaning ...( and as per MDN we can use the query selector to query by ID also, which by the way I never implemented before )..details: <style>#ser{color: cyan;}</style> <body> <div id="ser">welcome</div> <script> let x=[]; const json1 = '{"google.com":"google"}'; const obj1 = JSON.parse(json1); ol = document.createElement('ol'); document.querySelector('#ser').appendChild(ol); for (var key in obj1){ x= document.createElement('li'); x.innerHTML="<a href=" + key + ">" + obj1[key]+"</a>"; ol.appendChild(x);}

– [jonathan](https://stackoverflow.com/users/19542576/jonathan)

-----------------------------------------------------------------

# [My generated link in the unordered list is not navigable](https://stackoverflow.com/questions/73583672/my-generated-link-in-the-unordered-list-is-not-navigable)

[Ask Question](https://stackoverflow.com/questions/ask)

let x=[];

let array = []

const json1 = '{"https://www.w3schools.com/html/":"check w3 schools"}';

const obj1 = JSON.parse(json1);

array=Object.getOwnPropertyNames(obj1);

ul = document.createElement('ul');

ser.appendChild(ul);

x= document.createElement('li');

x.innerText="<a href=\"" + array[0]+ "\">"+"</a>";//hyper link not clickable

ul.appendChild(x);

<div id="ser"></div>

 Run code snippet

the javascript code renders the unordered list but the link in the list() is not navigable.Please advice on solving this issue.

innerText sets *text*. Use innerHTML instead.

– [Cerbrus](https://stackoverflow.com/users/1835379/cerbrus)

[Sep 2 at 13:56](https://stackoverflow.com/questions/73583672/my-generated-link-in-the-unordered-list-is-not-navigable#comment129940798_73583672)

* used innerHTML , and the link in the unordered list completely disappears..

– [jonathan](https://stackoverflow.com/users/19542576/jonathan)

[Sep 2 at 13:59](https://stackoverflow.com/questions/73583672/my-generated-link-in-the-unordered-list-is-not-navigable#comment129940870_73583672)

* 1

You also need to add some text inside the tag - <a href="...">link text here</a>

– [phuzi](https://stackoverflow.com/users/592958/phuzi)

[Sep 2 at 14:01](https://stackoverflow.com/questions/73583672/my-generated-link-in-the-unordered-list-is-not-navigable#comment129940942_73583672)

* 1

Because there's no text in the `<a></a> tag.

– [Cerbrus](https://stackoverflow.com/users/1835379/cerbrus)

[Sep 2 at 14:01](https://stackoverflow.com/questions/73583672/my-generated-link-in-the-unordered-list-is-not-navigable#comment129940947_73583672)

* i put console.log(x); before the /script tag and the <a></a> tag in the console is :<a href="[w3schools.com/html/"></a](https://www.w3schools.com/html/%22%3E%3C/a)>

– [jonathan](https://stackoverflow.com/users/19542576/jonathan)

[Sep 2 at 14:09](https://stackoverflow.com/questions/73583672/my-generated-link-in-the-unordered-list-is-not-navigable#comment129941128_73583672)   Delete

But there's no text inside the a tag to be rendered as the link. You would need <a href="w3schools.com/html/">W3Schools</a> to have the link show up and be rendered as "W3Schools"

– [phuzi](https://stackoverflow.com/users/592958/phuzi)

[Sep 2 at 14:11](https://stackoverflow.com/questions/73583672/my-generated-link-in-the-unordered-list-is-not-navigable#comment129941184_73583672)

* so Thanks this solved it x.innerHTML="<a href=\"" + array[0]+ "\">w3schools"+"</a>";

– [jonathan](https://stackoverflow.com/users/19542576/jonathan)

[Sep 2 at 14:12](https://stackoverflow.com/questions/73583672/my-generated-link-in-the-unordered-list-is-not-navigable#comment129941200_73583672)    Delete

Answer2:

There are multiple fixes so take a look at this

let array = []

const json1 = '{"https://www.w3schools.com/html/":"check w3 schools"}';

const obj1 = JSON.parse(json1);

array = Object.getOwnPropertyNames(obj1);

const ul = document.querySelector('#ser').appendChild(document.createElement('ul'));

const x = document.createElement('li');

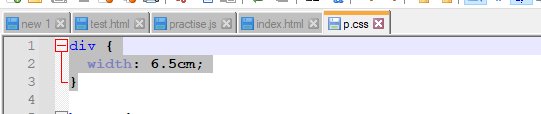
const a = document.createElement('a');

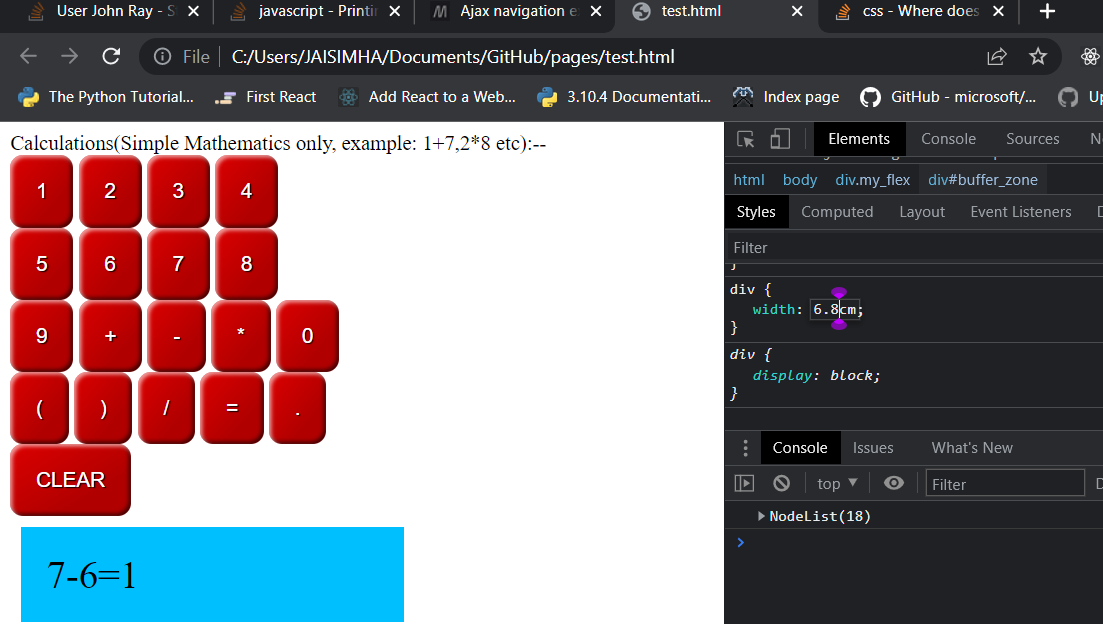
a.setAttribute("href", array[0]);

a.innerHTML = array[0];

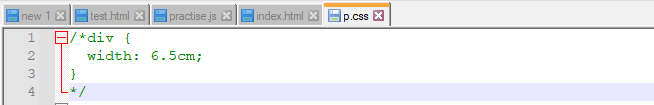
x.appendChild(a);

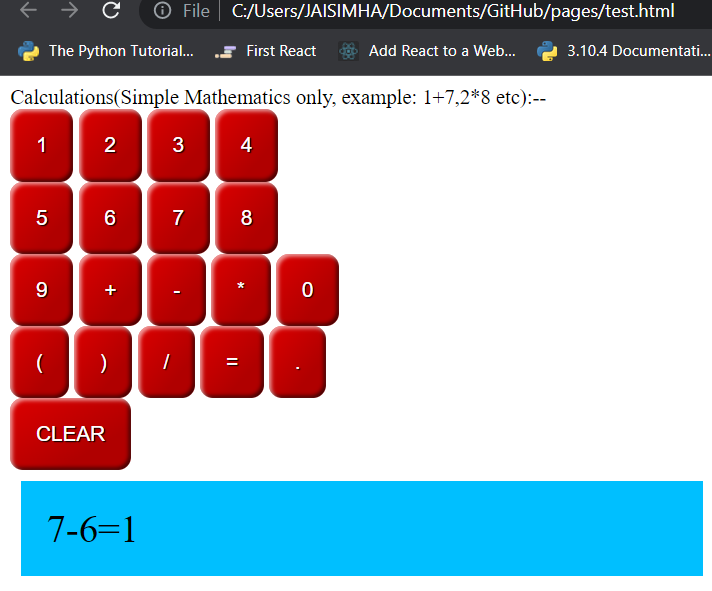
Changing width of DIV element is a WORTH IT!! :



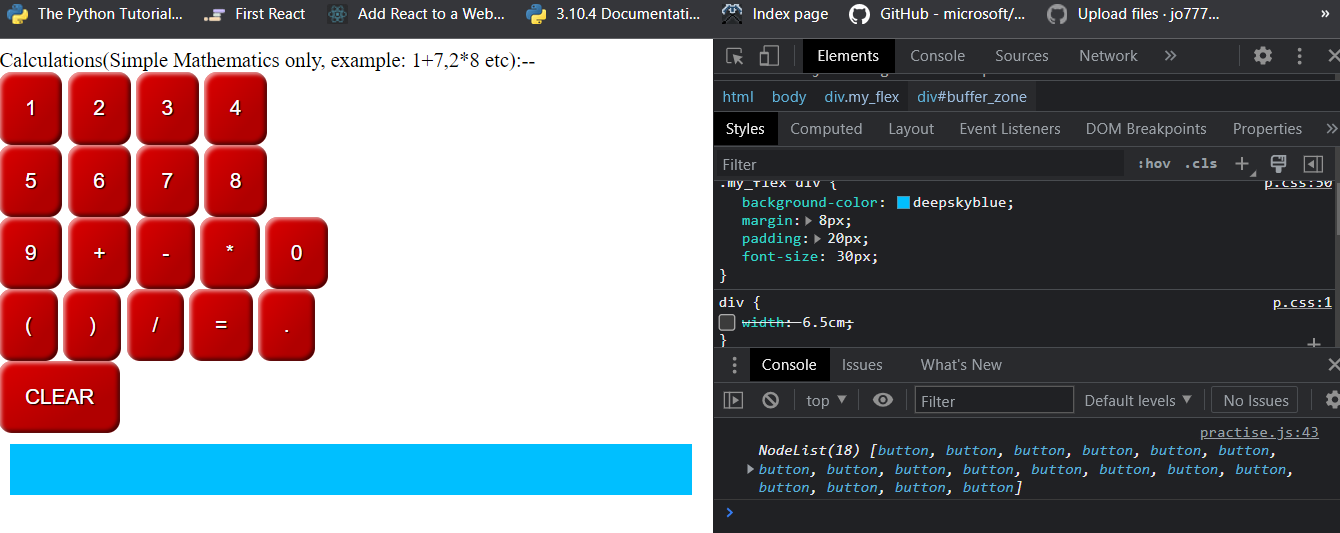


Next don’t set DIV width(this will occupy the whole window):





Next using chrome developer tools try changing the values(make div width to 0 and you can get it confirmed by seeing the strikethrough line:



Now change it to say 4 cm and observe elements are getting wrapped:

