### **[Charles7641@outlook.com/R...i](mailto:Charles7641@outlook.com/R...i)....**

[Simonemichelle40@gmail.com](mailto:Simonemichelle40@gmail.com)

[jfr829324@gmail.com](mailto:jfr829324@gmail.com)

ja1133@protonmail.com

https://stackoverflow.com/questions/75301106/not-being-able-to-display-button-in-reactjs

in order to render the Button, you have to render the component referencing it, in this case, MyApp.

<https://stackoverflow.com/questions/75250832/getting-unexpected-token-error-in-console-while-running-react>

JSX is not valid JavaScript, so it needs to be converted to valid JS. You can use a tool like [babel](https://babeljs.io/) to convert it to valid code.

– [code](https://stackoverflow.com/users/15359157/code)

[Jan 26 at 19:34](https://stackoverflow.com/questions/75250832/getting-unexpected-token-error-in-console-while-running-react#comment132788092_75250832)

* I never used babel so how do i go about?

– [Michael36](https://stackoverflow.com/users/19129298/michael36)

[Jan 26 at 19:35](https://stackoverflow.com/questions/75250832/getting-unexpected-token-error-in-console-while-running-react#comment132788113_75250832)

* 2

Did you use the create-react-app, or is this just in a plain html file?

– [Brenton Haerr](https://stackoverflow.com/users/12964542/brenton-haerr)

[Jan 26 at 19:42](https://stackoverflow.com/questions/75250832/getting-unexpected-token-error-in-console-while-running-react#comment132788222_75250832)

* Converted using babel and it worked-- "use strict"; const root = ReactDOM.createRoot(document.getElementById('root')); root.render( /\*#\_\_PURE\_\_\*/React.createElement("h1", null, "Hello, world!"));

– [Michael36](https://stackoverflow.com/users/19129298/michael36)

[Jan 26 at 19:45](https://stackoverflow.com/questions/75250832/getting-unexpected-token-error-in-console-while-running-react#comment132788269_75250832)

* i used just plain html file

– [Michael36](https://stackoverflow.com/users/19129298/michael36)

[Jan 26 at 19:45](https://stackoverflow.com/questions/75250832/getting-unexpected-token-error-in-console-while-running-react#comment132788281_75250832)

* 1

You can (for now) you can just use Babel standalone: [babeljs.io/docs/en/babel-standalone](https://babeljs.io/docs/en/babel-standalone)

– [code](https://stackoverflow.com/users/15359157/code)

[Jan 26 at 20:00](https://stackoverflow.com/questions/75250832/getting-unexpected-token-error-in-console-while-running-react#comment132788522_75250832)

<https://stackoverflow.com/questions/75046442/does-inline-block-not-respect-bottom-and-right-margins>

<p class="pick">hello1</p>

<p class="pick">hello2</p>

<p class="pick">hello3</p>

<div>1234</div>

body{

display: inline-block;

margin: 100px 10px 10px 90px;

/\* top: 100 px margin \*/

/\* right: 10 px margin \*/

/\* bottom: 10 px margin \*/

/\* left: 90 px margin \*/

}

<p>hello</p>

Are only top and left margins respected by inline-block elements ?I

you have nothing after the inline-block element to verify the effect of margin

– [Temani Afif](https://stackoverflow.com/users/8620333/temani-afif" \o "231,413 reputation)

[Jan 8 at 8:42](https://stackoverflow.com/questions/75046442/does-inline-block-not-respect-bottom-and-right-margins#comment132433944_75046442)

* please explain with some example code

– [Michael36](https://stackoverflow.com/users/19129298/michael36)

[Jan 8 at 8:44](https://stackoverflow.com/questions/75046442/does-inline-block-not-respect-bottom-and-right-margins#comment132433959_75046442)

* if i change top and left margin values i can see changes being reflected in browser

– [Michael36](https://stackoverflow.com/users/19129298/michael36)

[Jan 8 at 8:47](https://stackoverflow.com/questions/75046442/does-inline-block-not-respect-bottom-and-right-margins#comment132433980_75046442)   Delete

* add more elements to your code

– [Temani Afif](https://stackoverflow.com/users/8620333/temani-afif" \o "231,413 reputation)

[Jan 8 at 8:51](https://stackoverflow.com/questions/75046442/does-inline-block-not-respect-bottom-and-right-margins#comment132434015_75046442)

* added more elements to code and feel my question is wrong, pls confirm

– [Michael36](https://stackoverflow.com/users/19129298/michael36)

[Jan 8 at 10:04](https://stackoverflow.com/questions/75046442/does-inline-block-not-respect-bottom-and-right-margins#comment132434643_75046442)

for inline-block it doesnt matter

<https://stackoverflow.com/questions/74970812/why-do-we-need-to-use-a-calc-function-for-a-simple-operation-in-css>

n the [MDN web docs calc example](https://developer.mozilla.org/en-US/docs/Web/CSS/calc), I see a demo example with width: calc(10px + 100px) but I could instead simply use width: 110px.

So isn't calc(10px + 100px) complicating the code for a simple operation? How does calc really help me?

using calc is as you said, useless for two values of the same unit type, unless they're used through variables, e.g.:

:root {

--some-width: 50px;

}

.some-element {

width: calc(var(--some-width) + 100px);

}

Which would compute to 150px if --some-width is kept as 50px, however, --some-width can be changed to anything else, making the calc useful in this example - but useless in the provided example of calc(10px + 100px).

They're useful for combing relative units, such as em or % with other values (of irrelevant type).

See <https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Values_and_units> for relative length units.

The example of calc(10px + 100px) is as you said correctly, introducing more processing without any good reasoning, although not at a large scale.

efer example in : [mdn autocomplete](https://developer.mozilla.org/en-US/docs/Web/HTML/Attributes/autocomplete) Also refer quote in link:

Note: In order to provide autocompletion, user-agents might require // elements to:

Have a name and/or id attribute Be descendants of a element The form to have a submit button

CSS:

button,input

{

display:block;

margin: 4px;

}

Htm: <form id="myGeeks">

<label for="firstName">First Name:</label>

<input name="firstName" type="text" autocomplete="given-name">

<label for="lastName">Last Name:</label>

<input name="lastName" type="text" autocomplete="family-name">

<label for="MiddleName">Middle Name:</label>

<input name="MiddleName" type="text" autocomplete="additional-name">

<label for="NickName">Nick Name:</label>

<input name="nickName" type="text" autocomplete="nickname">

<label for="email">Email:</label>

<input name="email" type="email" autocomplete="email">

<button>Submit</button>

</form>

<https://stackoverflow.com/questions/74775529/cant-select-values-using-html-u-a-tags-but-works-for-ahref>

a[href="https://cnn.com/"] {

color: red;

}

<a href="https://cnn.com/">cnn</a>

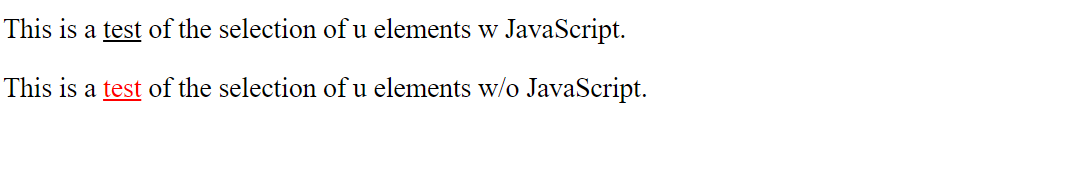
.r{

color: red;

}

<p>This is a <u>test</u> of the selection of u elements w JavaScript.</p>

<p>This is a <u class=r>test</u> of the selection of u elements w/o JavaScript.</p>



<https://stackoverflow.com/questions/74685136/how-to-reduce-letter-spacing-between-div-letters-h-2-o>

<!DOCTYPE html>

<html>

<head>

<style>

div {

display: block;

}

.sub1{

vertical-align: sub;

font-size:x-small;

}

</style>

</head>

<body>

<div>H<span class="sub1">2</span>O</div>

<div>CO<span class="sub1">2</span></div>

<div>SO<span class="sub1">4</span></div>

</div>

</body>

</html>

https://stackoverflow.com/questions/74618931/what-does-this-documentation-syntax-mean-in-mdn-for-inherit-css-tag

-1

I have a basic question , refer to section as an example

Constituent properties in [list-style](https://developer.mozilla.org/en-US/docs/Web/CSS/list-style)

how do I interpret this statement of : inherit wherever it gets mentioned in mdn docs in a layman/ generic form? I also see Inherited value as Yes in Formal definition section so is it connected to :inherit in any way or not

<https://developer.mozilla.org/en-US/docs/Web/CSS/inherit>

The inherit CSS keyword causes the element to take the computed value of the property from its parent element. It can be applied to any CSS property, including the CSS shorthand property all.

1

how I can see the parent elements and the child elements visually in chrome browser?

– [Michael36](https://stackoverflow.com/users/19129298/michael36)

[Nov 29, 2022 at 19:00](https://stackoverflow.com/questions/74618931/what-does-this-documentation-syntax-mean-in-mdn-for-inherit-css-tag#comment131712089_74618974)

* 2

Open the page in Chrome, then press Ctrl-Shift-I to open the Developer tools and select the Elements tab. [Documentation](https://developer.chrome.com/docs/devtools/). If you have specific questions about Dev Tools, ask a new question.

What is role in CSS?

The role attribute **describes the role of an element in programs that can make use of it, such as screen readers or magnifiers**. Usage Example: <a href="#" role="button">Button Link</a> Screen Readers will read this element as “button” instead of “link”.13-Jan-2020

https://stackoverflow.com/questions/74598383/why-does-displayinline-in-ul-css-tag-causes-elements-to-shift

# [why does display:inline in ul css tag causes elements to shift](https://stackoverflow.com/questions/74598383/why-does-displayinline-in-ul-css-tag-causes-elements-to-shift)

ul{display: inline;}

<ul>hi

<li>1234</li>

<li>5678</li>

</ul>

<ul>hello

<li>abcdef</li>

<li>ghijkl</li>

</ul>

question:the ul items(hi,hello) in above css code moved a couple of places to the right if I used the css display:inline tag . But They do not get moved if I execute with a css ul tag having no display:inline value..please explain. and second question why have the circle markers disappeared ?

answer1:

li elements have a display value equal to list-item and following [the specification](https://w3c.github.io/csswg-drafts/css-display/#the-display-properties) they generate a block box so you end having **a block element inside and inline element**.

The above behavior is also defined in the specification and leads to the result you get. More detail: [Is it wrong to change a block element to inline with CSS if it contains another block element?](https://stackoverflow.com/q/746531/8620333)

why have the circle markers disappeared ?

It's still there but hidden on the left because the default behavior is list-style-position: outside

ul{display: inline;}

li {margin-left: 20px;}

<ul>hi

<li>1234</li>

<li>5678</li>

</ul>

<ul>hello

<li>abcdef</li>

<li>ghijkl</li>

</ul>

aNs2:

ul gets a default padding-left applied from the user agent stylesheet, 40px or something.

With an inline element, padding-left works only before the first line of content, and padding-right only after the last line.

Make it inline-block instead, if you want that padding applied to the whole element.

Ans3:

Because the **inline** value of the **display** property is something that makes the elements inside to behaves inline.

That means you have not much options to position and move them.

The **inline** value is most useful for a text paragraphs to wrap theentire paragraph. Where you would like the text to position in a couple of lines one below another.

<https://stackoverflow.com/questions/74591932/children-dont-inherit-style-despite-declaration-on-ancestor>

# [Children don't inherit style despite declaration on ancestor](https://stackoverflow.com/questions/74591932/children-dont-inherit-style-despite-declaration-on-ancestor)

I declared display: inline on body but the elements are still displayed in block fashion. Please help me understand the concept.

Declaring display: inline directly on li works: They will display in inline fashion. But why doesn't inline apply to them when declaring on body?

body {display: inline}

<body>

<ul>

<li>1234</li>

<li>5678</li>

</ul>

</body>

 Comment:

I also tried with display: inherit on li, but it still displays in block fashion, and also removes the markers which I don't want:

body {display: inline}

li {display: inherit}

<body>

<ul>

<li>1234</li>

<li>5678</li>

</ul>

</body>

* Answer:

display is not inherited: Setting the <BODY> to display: inline won't affect its children's display value.

– [Oskar Grosser](https://stackoverflow.com/users/13561410/oskar-grosser)

[Nov 27, 2022 at 17:00](https://stackoverflow.com/questions/74591932/children-dont-inherit-style-despite-declaration-on-ancestor#comment131665780_74591932)

f I wanna display inline to be inhertiable by all the children's display value ,then what would be the code?. I tried <style> body { display: inline; } li { display: inherit; } </style> but it still shows in BLOCK

– [Michael36](https://stackoverflow.com/users/19129298/michael36)

Setting display on an element will not have its children inherit the value:

.wrapper {display: inline}

<div class="wrapper">

<div>First text.</div>

<div>Second text.</div>

</div>



While you can declare elements to inherit display from their ancestors, I do not recommend this (generally):

* Declaring elements to inherit display requires a declaration just as much as declaring a specific display value.
* If the selector is too unspecific, elements with sensible default values will lose those values.

It is important to choose good selectors for your use case. For your short example it is fine to have a generic selector, like body \* (in words: "All descendants of BODY") which uses the [universal selector \*](https://developer.mozilla.org/en-US/docs/Web/CSS/Universal_selectors) and a [descendant combinator](https://developer.mozilla.org/en-US/docs/Web/CSS/Descendant_combinator) .

**Note:** Using a combinator generally does not select "sub-selections" itself, e.g. BODY of selector body \* will not be selected. Also note that the *descendant combinator* is less specific than the [child combinator >](https://developer.mozilla.org/en-US/docs/Web/CSS/Child_combinator).

Here is an analogous example: **also note difference between .wrapper {display: inline}**

**And .wrapper \* {display: inline}**

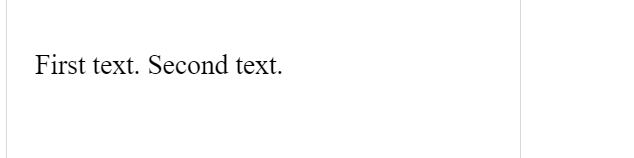
.wrapper \* {display: inline}

<div class="wrapper">

<div>First text.</div>

<div>Second text.</div>

</div>



If however you still want to declare the elements to *inherit* display, here is an example:

.wrapper {display: inline} /\*Set the value to inherit\*/

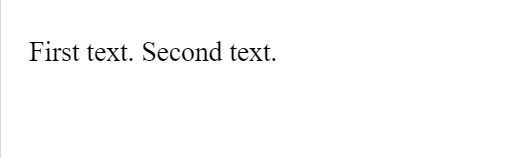
.wrapper \* {display: inherit}

<div class="wrapper">

<div>First text.</div>

<div>Second text.</div>

</div>



**Lists and display**

Lists' default display value is block, and the value of its children (LI) is list-item. If you simply want to get rid of the markers, declare the list list-style-type: none:

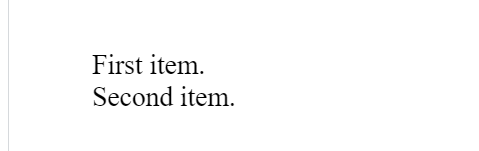
ul {list-style-type: none}

<ul>

<li>First item.

<li>Second item.

</ul>



But you can also declare the list and its children (ul, ul>\*) as display: inline. Note that removing padding requires another declaration.

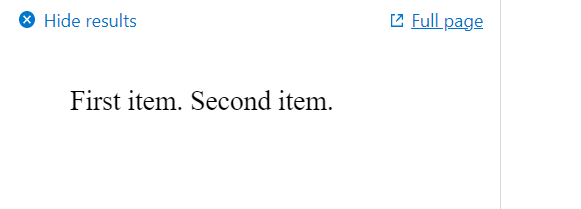
ul, ul>\* {display: inline}

<ul>

<li>First item.

<li>Second item.

</ul>



## **Inline list with markers**

If you want to inline the list-items with markers, there are several possibilities:

You can keep [::marker](https://developer.mozilla.org/en-US/docs/Web/CSS/::marker)s of inline list-items by declaring the [list as display: inline-flex](https://stackoverflow.com/a/69368934/13561410).

**Sidenote:** ::markers are by default [outside](https://developer.mozilla.org/en-US/docs/Web/CSS/::marker) the list-item's [principal box](https://stackoverflow.com/a/69368934/13561410), which means they can obstruct view of other boxes. Declare the list (or its items) as [list-style-position: inside](https://developer.mozilla.org/en-US/docs/Web/CSS/flex-wrap#nowrap) to keep the markers inside the list-item's box to prevent such overflow.

**Note** that [flex-wrap: nowrap](https://developer.mozilla.org/en-US/docs/Web/CSS/flex-wrap#nowrap) is its default, which is unlike regular text flow. Use [flex-wrap: wrap](https://developer.mozilla.org/en-US/docs/Web/CSS/flex-wrap#wrap) to have the list-items (and their content) flow like regular text:

// Ignore; for handling radio inputs

const form = document.querySelector("form");

const list = document.querySelector("ol");

form.addEventListener("input", evt => {

const className = evt.target.name === "position" ? "inside" : "wrap";

// Classes for applying non-default values are named after the values

const state = evt.target.value === className;

list.classList.toggle(className, state);

});

ol {display: inline-flex}

.inside {list-style-position: inside}

.wrap {flex-wrap: wrap}

/\*Ignore; for presentational purposes\*/

ol {margin: 0; padding: 0}

li {border: 1px solid darkgray}

section {margin-block: 1.2rem}

<form>

<div>

List type:

<label for="i-outside">

<input id="i-outside" name="position" type="radio" value="outside" checked> Outside

</label>

<label for="i-inside">

<input id="i-inside" name="position" type="radio" value="inside"> Inside

</label>

</div>

<div>

Wrapping:

<label for="i-nowrap">

<input id="i-nowrap" name="wrap" type="radio" value="nowrap" checked> No-wrap

</label>

<label for="i-wrap">

<input id="i-wrap" name="wrap" type="radio" value="wrap"> Wrap

</label>

</div>

</form>

<ol>

<li>A very long named item that hopefully overflows its box, and

<li>An item, and

<li>Another short item, and

<li>another very long named item that too should overflow its box.

</ol>

## **List-items don't inherit display?**

You asked why the following didn't result in the LIs to inherit the BODY's display value:

body {display: inline}

li {display: inherit}

<body>

<ul>

<li>First item.

<li>Second item.

</ul>

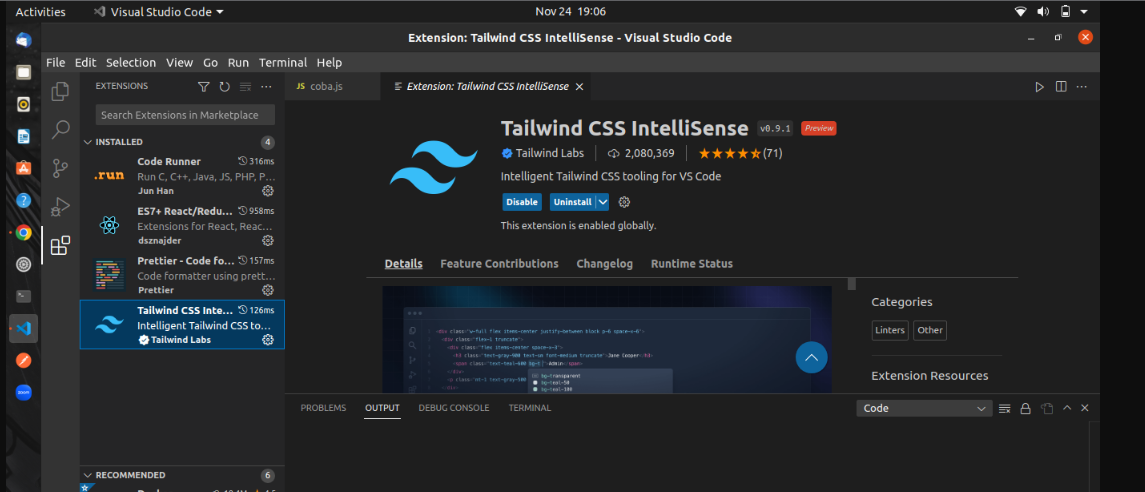
</body>



That is because—while BODY is declared as display: inline—the list(ol) in between has the default display value of block. Since inheritance happens from the element's immediate ancestor, the list-items will inherit the value block from the list

# [Tailwind css design concept and implementation](https://stackoverflow.com/questions/74559160/tailwind-css-design-concept-and-implementation)

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



# [scss : the ampersand css operator is not working for my html page](https://stackoverflow.com/questions/74292937/scss-the-ampersand-css-operator-is-not-working-for-my-html-page)

a {

&:visited {

color: cyan;

}

&:hover {

color: orangered;

}

}

<h1>CSS operator</h1>

<dl>

<dt><a href="https://getbootstrap.com/">Bootstrap CSS Home page</a></dt>

<dd>BOOTSTRAP CSS</dd>

<dt><a href= "https://docs.python.org/3/howto/sorting.html#sortinghowto">Sorting</a></dt>

<dd>Python Sorting</dd>

</dl>

 Run code snippet

Expand snippet

Issue regarding the & CSS operator and pseudo selector:

(1) If I am hovering over the a href link for say BOOTSTRAP CSS, it is not changing color to orangered as per  
&:hover {color: orangered;}

(2) The link color is not changing to cyan after I visit the ahref link(click) as per: &:visited {color: cyan;}

* [html](https://stackoverflow.com/questions/tagged/html)
* [css](https://stackoverflow.com/questions/tagged/css)
* [sass](https://stackoverflow.com/questions/tagged/sass)

[Share](https://stackoverflow.com/q/74292937/19129298)

[Edit](https://stackoverflow.com/posts/74292937/edit)

Delete

Flag

[edited just now](https://stackoverflow.com/posts/74292937/revisions)

asked Nov 2, 2022 at 16:53

[[](https://stackoverflow.com/users/19129298/michael36)](https://stackoverflow.com/users/19129298/michael36)

[Michael36](https://stackoverflow.com/users/19129298/michael36)

**131**1111 bronze badges

* 3

You cannot run scss in browser you need to compile it to standard css

– [kennarddh](https://stackoverflow.com/users/14813577/kennarddh" \o "1,581 reputation)

[Nov 2, 2022 at 16:57](https://stackoverflow.com/questions/74292937/scss-the-ampersand-css-operator-is-not-working-for-my-html-page#comment131161782_74292937)

* how to do that? if possible can you send some documentation links

– [Michael36](https://stackoverflow.com/users/19129298/michael36)

[Nov 2, 2022 at 16:58](https://stackoverflow.com/questions/74292937/scss-the-ampersand-css-operator-is-not-working-for-my-html-page#comment131161818_74292937)

* 1

You can read about scss and sass [here](https://sass-lang.com/)

– [kennarddh](https://stackoverflow.com/users/14813577/kennarddh" \o "1,581 reputation)

[Nov 2, 2022 at 16:59](https://stackoverflow.com/questions/74292937/scss-the-ampersand-css-operator-is-not-working-for-my-html-page#comment131161834_74292937)

### [**why does Javascript force me to use a IFFE sometimes [duplicate]**](https://stackoverflow.com/questions/73996350/why-does-javascript-force-me-to-use-a-iffe-sometimes)

# https://stackoverflow.com/questions/8228281/what-is-the-function-construct-in-javascript

It’s an [Immediately-Invoked Function Expression](http://benalman.com/news/2010/11/immediately-invoked-function-expression/), or [**IIFE**](https://en.wikipedia.org/wiki/Immediately-invoked_function_expression) for short. It executes immediately after it’s created.

It has nothing to do with any event-handler for any events (such as document.onload).  
Consider the part within the first pair of parentheses: (**function(){}**)();....it is a regular function expression. Then look at the last pair (function(){})**()**;, this is normally added to an expression to call a function; in this case, our prior expression.

This pattern is often used when trying to avoid polluting the global namespace, because all the variables used inside the IIFE (like in any other normal function) are not visible outside its scope.  
This is why, maybe, you confused this construction with an event-handler for window.onload, because it’s often used as this:

(function(){

// all your code here

var foo = function() {};

window.onload = foo;

// ...

})();

// foo is unreachable here (it’s undefined)

**Correction suggested by**[**Guffa**](https://stackoverflow.com/users/69083/guffa)**:**

The function is executed right after it's created, not after it is parsed. The entire script block is parsed before any code in it is executed. Also, parsing code doesn't automatically mean that it's executed, if for example the IIFE is inside a function then it won't be executed until the function is called.

**Update** Since this is a pretty popular topic, it's worth mentioning that IIFE's can also be written with [ES6's arrow function](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions) (like [Gajus](https://stackoverflow.com/users/368691/gajus) has pointed out [in a comment](https://stackoverflow.com/questions/8228281/what-is-the-function-construct-in-javascript/8228308#comment43521302_8228308)) :

((foo) => {

// do something with foo here foo

})('foo value')

console.log("==PARALLEL with await Promise.all==");

parallel();

// Definition of parallel function to Start 5 "jobs" in parallel and wait for both of them to complete

async function parallel() {

await Promise.all([

(async () => await a()),

(async () => await b()),

(async () => await c()),

(async () => await d()),

(async () => await e()),

]);

}

function a()

{

console.log("In function a ");

return Promise.resolve(1);

}

function b(){

console.log("In function b ");

return Promise.resolve(2);

}

function c(){

console.log("In function c ");

return Promise.resolve(3);

}

function d(){

console.log("In function d ");

return Promise.resolve(4);

}

function e(){

console.log("In function e ");

return Promise.resolve(5);

}

Andreas refer to your latest comment i tried out an array of function -await Promise.all([a(), b(), c(), d(), e()]) ,and it worked.credit @emiel-zuurbier

– [Charles](https://stackoverflow.com/users/20189585/charles)

# [Oct 8 at 11:06](https://stackoverflow.com/questions/73996350/why-does-javascript-force-me-to-use-a-iffe-sometimes#comment130653394_73996350)

# [not able to render all images](https://stackoverflow.com/questions/74117372/not-able-to-render-all-images)

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

# <https://stackoverflow.com/questions/74117372/not-able-to-render-all-images>

I can retrieve 3 images (apple.jpg, banana.jpg, kiwi.jpg) using the JS Fetch API: promises fulfilled and blob created but can render only one the kiwi.jpg image and not the remaining two jpg images maybe because of overlapping . Kindly advise any concepts I am missing and help.

Note: All the above mentioned .jpg images are present in the same location as the HTML and JS code.

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

var x=["apple.jpg","banana.jpg","kiwi.jpg"]

x.forEach((x1)=>{

myRequest = new Request(x1)

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

myBlob=await response.blob()

objectURL = URL.createObjectURL(myBlob)

myImage.src = URL.createObjectURL(myBlob)

URL.revokeObjectURL(objectURL)

}

)}

)

img{

display:inline

}

<!--below apple.jpg image should appear -->

<img class="someImage" src</img>

<!--below banana.jpg image should appear -->

<img src=" " />

<!--below kiwi.jpg image should appear -->

<img src=" " />

# Answers:

and the JS Snippet to: const myImages = document.querySelectorAll("img"); const imagePaths = ["./potato.jpg", "./banana.jpg", "./kiwi.jpg"]; imagePaths.forEach((path, i) => { fetch(path).then(response => response.blob()).then(blob => { const objectURL = URL.createObjectURL(blob); myImages[i].src = objectURL; // URL.revokeObjectURL(objectURL); }); })

–

# Comment:using arr.fill method:

const myImages = document.querySelectorAll("img");

//const imagePaths = ["./apple.jpg", "./banana.jpg", "./kiwi.jpg"];

const placeholderUrl = "https://www.gravatar.com/avatar/5c9a8b9f51420f0a4f548de5a6e39bd3?s=64&d=identicon&r=PG";

const imagePaths = Array(3).fill(placeholderUrl);

imagePaths.forEach((path, i) => {

fetch(path).then(response => response.blob()).then(blob => {

const objectURL = URL.createObjectURL(blob);

myImages[i].src = objectURL;

//URL.revokeObjectURL(objectURL);

});

})

# SAAS-CSS Preprocessing:

# [the ampersand css operator is not working for my html page](https://stackoverflow.com/questions/74292937/the-ampersand-css-operator-is-not-working-for-my-html-page)

# <https://stackoverflow.com/questions/74292937/the-ampersand-css-operator-is-not-working-for-my-html-page>

a {

&:visited {

color: cyan;

}

&:hover {

color: orangered;

}

}

<h1>CSS operator</h1>

<dl>

<dt><a href="https://getbootstrap.com/">Bootstrap CSS Home page</a></dt>

<dd>BOOTSTRAP CSS</dd>

<dt><a href= "https://docs.python.org/3/howto/sorting.html#sortinghowto">Sorting</a></dt>

<dd>Python Sorting</dd>

</dl>

Issue regarding the & CSS operator and pseudo selector:

(1) If I am hovering over the a href link for say BOOTSTRAP CSS, it is not changing color to orangered as per  
&:hover {color: orangered;}

(2) The link color is not changing to cyan after I visit the ahref link(click) as per: &:visited {color: cyan;}

You cannot run scss in browser you need to compile it to standard css

– [kennarddh](https://stackoverflow.com/users/14813577/kennarddh" \o "1,366 reputation)

# You can read about scss and sass [here](https://sass-lang.com/)

# https://sass-lang.com/

# [Please explain how one can know the parent html element [closed]](https://stackoverflow.com/questions/74547531/please-explain-how-one-can-know-the-parent-html-element)

# https://stackoverflow.com/questions/74547531/please-explain-how-one-can-know-the-parent-html-element

.box {

margin-top: 40px;

margin-right: 30px;

margin-bottom: 40px;

margin-left: 2em;

}

<div class="container">

<div class="box">Change my margin.</div>

</div>

refer to [EM Calculations](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Values_and_units) and go to section ems and rems which talks about to calculate ems..so if my margin-left is 2 em , what's its parent element (so that i calculate margin -left corresponding value in px)

# Answers:

# By default <html> (or :root) gets a font size of 16px or normal (usually 16px too) depeding on the browser, the *root font-size* (= 1rem). The em unit means: multiply a value with the *parent font-size*. Child elements inherit the font-size from their parent, so if no font sizes have been set for an element or its ancestors then 1em = 16px, so 2em means 2 x 16px = 32px.

# [how can I see rejections being generated for a promise](https://stackoverflow.com/questions/74008292/how-can-i-see-rejections-being-generated-for-a-promise)

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

# <https://stackoverflow.com/questions/74008292/how-can-i-see-rejections-being-generated-for-a-promise>

"use strict";

let promiseCount = 0;

function testPromise() {

const thisPromiseCount = ++promiseCount;

const log = document.getElementById("log");

// begin

log.insertAdjacentHTML("beforeend", `${thisPromiseCount}) Started<br>`);

// We make a new promise: we promise a numeric count of this promise,

// starting from 1 (after waiting 3s)

const p1 = new Promise((resolve, reject) => {

// The executor function is called with the ability

// to resolve or reject the promise

log.insertAdjacentHTML(

"beforeend",

`${thisPromiseCount}) Promise constructor<br>`

);

// This is only an example to create asynchronism

setTimeout(() => {

// We fulfill the promise

resolve(thisPromiseCount);

}, Math.random() \* 2000 + 1000);

});

// We define what to do when the promise is resolved with the then() call,

// and what to do when the promise is rejected with the catch() call

p1.then((val) => {

// Log the fulfillment value

log.insertAdjacentHTML("beforeend", `${val}) Promise fulfilled<br>`);

}).catch((reason) => {

// Log the rejection reason

console.log(`Handle rejected promise (${reason}) here.`);

});

// end

log.insertAdjacentHTML("beforeend", `${thisPromiseCount}) Promise made<br>`);

}

const btn = document.getElementById("make-promise");

btn.addEventListener("click", testPromise);

<button id="make-promise">Make a promise!</button>

<div id="log"></div>

# with refer to [promises](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise) how do i manually generate a rejected promise or under what circumstances will this code run rejections,please provide some practical concepts so that I can see following error message, please refer in snippet: console.log(Handle rejected promise (${reason}) here.);

## **1 Answer**

Sorted by:



1

Try testing promise rejection by generating rejected promises in the timeout code run within the promise executor function - meaning in the function passed to the global Promise constructor as its argument.

A simple approach might be to reject odd numbered promises while resolving even numbered promises with a settled value. The code below does this if you need an example. It also logs promise rejection in HTML.see below

"use strict";

let promiseCount = 0;

function testPromise() {

const thisPromiseCount = ++promiseCount;

const log = document.getElementById("log");

// begin

log.insertAdjacentHTML("beforeend", `${thisPromiseCount}) Started<br>`);

// We make a new promise: we promise a numeric count of this promise,

// starting from 1 (after waiting 3s)

const p1 = new Promise((resolve, reject) => {

// The executor function is called with the ability

// to resolve or reject the promise

log.insertAdjacentHTML(

"beforeend",

`${thisPromiseCount}) Promise constructor<br>`

);

// This is only an example to create asynchronism

setTimeout(() => {

// We fulfill the promise on odd calls, reject on even calls

(thisPromiseCount % 2 ? reject : resolve) (thisPromiseCount);

}, Math.random() \* 2000 + 1000);

});

// We define what to do when the promise is resolved with the then() call,

// and what to do when the promise is rejected with the catch() call

p1.then((val) => {

// Log the fulfillment value

log.insertAdjacentHTML("beforeend", `${val}) Promise fulfilled<br>`);

}).catch((reason) => {

// Log the rejection reason

console.log(`Handle rejected promise (${reason}) here.`);

log.insertAdjacentHTML("beforeend", `${thisPromiseCount}) Promise rejected<br>`)

});

// end

log.insertAdjacentHTML("beforeend", `${thisPromiseCount}) Promise made<br>`);

}

const btn = document.getElementById("make-promise");

btn.addEventListener("click", testPromise);

<button type="button" id="make-promise">#make-promise</button>

<div id="log"></div>

# [how to bypass JSON errors in Javascript [closed]](https://stackoverflow.com/questions/74027702/how-to-bypass-json-errors-in-javascript)

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

@Charles: As expected, this isn't about JSON at all. You have syntactically invalid string literals. You're essentially asking, "How do I tell the computer to ignore syntax errors and just know what I meant." You can't. Code needs to be syntactically valid. When you *parse JSON* you can put that in a try/catch and handle errors in the parsing. But *the code which tries to parse the JSON* needs to be valid code.

– [David](https://stackoverflow.com/users/328193/david)

# [Oct 11 at 12:49](https://stackoverflow.com/questions/74027702/how-to-bypass-json-errors-in-javascript#comment130709056_74027702)

### [**For parallel processing,'Promise All' not waiting for all promises to be resolved [duplicate]**](https://stackoverflow.com/questions/73995274/for-parallel-processing-promise-all-not-waiting-for-all-promises-to-be-resolve)

# There is a MDN spec [Async function](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/async_function),refer function [async function parallel()](https://i.stack.imgur.com/Sbl2L.png)which I am trying to implement as part of a larger project for parallel processing ,but All my promises are not being generated or being resolved .Only one promise of the PromiseAll has gone "fulfilled" state(as seen in debugger).I am new student to JS and Promises.

console.log("==PARALLEL with await Promise.all==");

parallel();

// Definition of parallel function to Start 5 "jobs" in parallel and wait for both of them to complete

async function parallel() {

await Promise.all([

(async () => await a()),

(async () => await b()),

(async () => await c()),

(async () => await d()),

(async () => await e()),

]);

}

function a()

{

console.log("In function a ");

return Promise.resolve(1);

}

function b(){

console.log("In function b ");

return Promise.resolve(2);

}

function c(){

console.log("In function c ");

return Promise.resolve(3);

}

function d(){

console.log("In function d ");

return Promise.resolve(4);

}

function e(){

console.log("In function e ");

return Promise.resolve(5);

}

<div>Test Promises</div>

you are not calling the functions. See the example below. It was difficult to explain what 'you are not calling' means, so once you see the working example let me know and we will delete this question

console.log("==PARALLEL with await Promise.all==");

parallel();

// Definition of parallel function to Start 5 "jobs" in parallel and wait for both of them to complete

async function parallel() {

await Promise.all([

(async () => await a())(),

(async () => await b())(),

(async () => await c())(),

(async () => await d())(),

(async () => await e())(),

]).then(console.log);

}

function a()

{

console.log("In function a ");

return Promise.resolve(1);

}

function b(){

console.log("In function b ");

return Promise.resolve(2);

}

function c(){

console.log("In function c ");

return Promise.resolve(3);

}

function d(){

console.log("In function d ");

return Promise.resolve(4);

}

function e(){

console.log("In function e ");

return Promise.resolve(5);

}

<div>Test Promises</div>

### [**async await execution in javascript**](https://stackoverflow.com/questions/73909214/async-await-execution-in-javascript)

# found some SO questions/answers but I am unable to get the async await javascript concept right..please advise ..my question is **Should not console.log('end') be executed two times?** .

debugger;

async function withoutAwait() {

console.log('without await')

}

async function withAwait() {

await 0

console.log('with await')

}

console.log('start')

withoutAwait()

withAwait()

console.log('end')

# 

# [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;

}

});

<div class="my\_flexx"></div>

 The part of the function before and including await 2022; runs synchronously. The part after await 2022

Just use different functions that you can chain. Since your execution is async, then you need to stop thinking synchronously.

async function loadStuff() {

// perform async loading...

}

async function doMoreStuff() {

// continue with more stuff

}

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

loadStuff().then(() => doMoreStuff());

});

Just use different functions that you can chain. Since your execution is async, then you need to stop thinking synchronously.

async function loadStuff() {

// perform async loading...

}

async function doMoreStuff() {

// continue with more stuff

}

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

loadStuff().then(() => doMoreStuff());

});

### **Edit 1**

These two functions are essentially equivalent :

async function somethingA() {

// synchronous code

return "SomeValue";

}

function somethingB() {

return new Promise(function (resolve) {

// synchronous code

resolve("SomeValue");

});

}

The async/await syntax is just cleaner.

By the way is the below code valid:async function foo() { await(1); return Promise.resolve(2009); }

– [Michael36](https://stackoverflow.com/users/19129298/michael36)

[Sep 29, 2022 at 19:34](https://stackoverflow.com/questions/73899801/how-to-confirm-for-javascript-asynchronous-calls#comment130487920_73899916)

* 1

@logogomez technically, yes, you can "await" any expression; await 1 is equivalent to Promise.resolve(1), so the above code would be like return Promise.resolve(1).then(() => Promise.resolve(2009)). It will not generate any error, but it won't do anything useful either.

– [Yanick Rochon](https://stackoverflow.com/users/320700/yanick-rochon)

[Sep 29, 2022 at 19:48](https://stackoverflow.com/questions/73899801/how-to-confirm-for-javascript-asynchronous-calls#comment130488130_73899916)

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.

Examples:

async function withoutAwait() {

console.log('without await')

}

async function withAwait() {

await 0

console.log('with await')

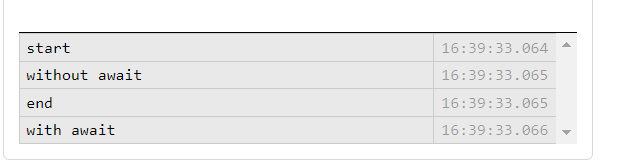
}

console.log('start')

withoutAwait()

withAwait()

console.log('end')



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)

# 

JavaScript code runs in one thread. No two lines of code run at the same time.

Asnychrouns code just stops executing when it encounters await and goes back to the next fragment when the thread is free

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)

# 

# 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/" \l "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>

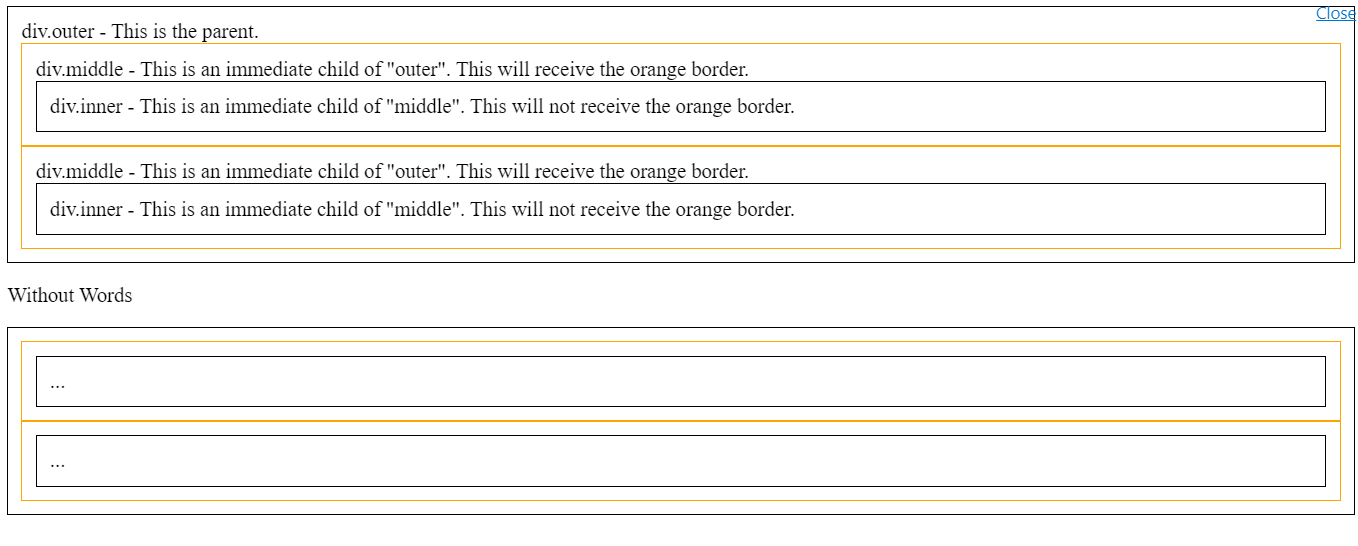
# So,

# 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>"

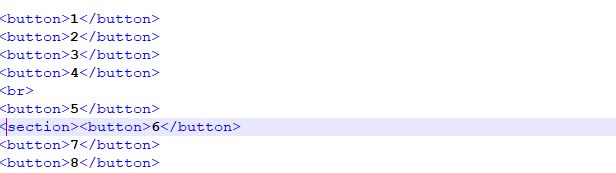
}

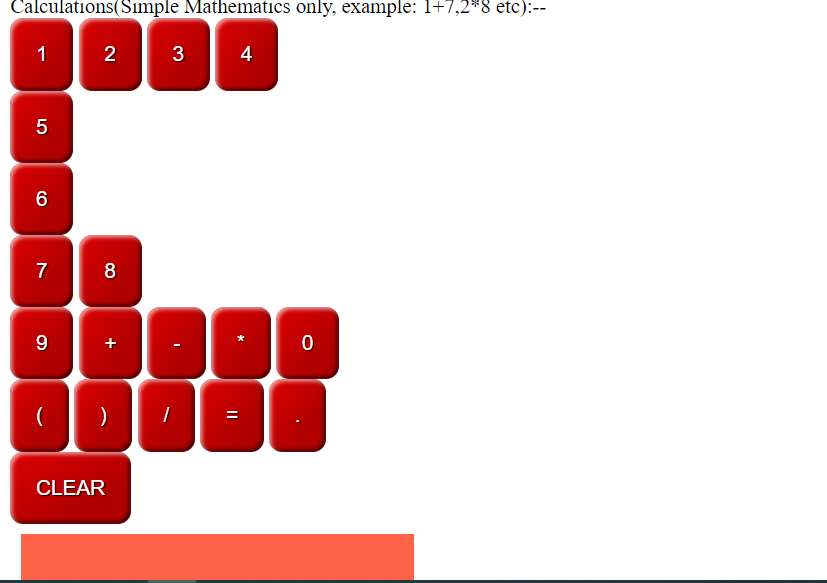
So Use:

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" \o "13 reputation)

[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" \o "13 reputation)

[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" \o "13 reputation)

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

# [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" \o "67,040 reputation)

[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" \o "13 reputation)

[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" \o "10,675 reputation)

[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" \o "67,040 reputation)

[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" \o "13 reputation)

[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" \o "10,675 reputation)

[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" \o "13 reputation)

[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');

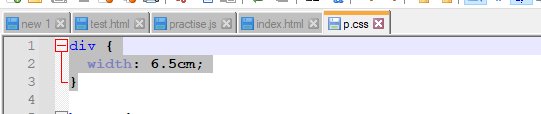
const a = document.createElement('a');

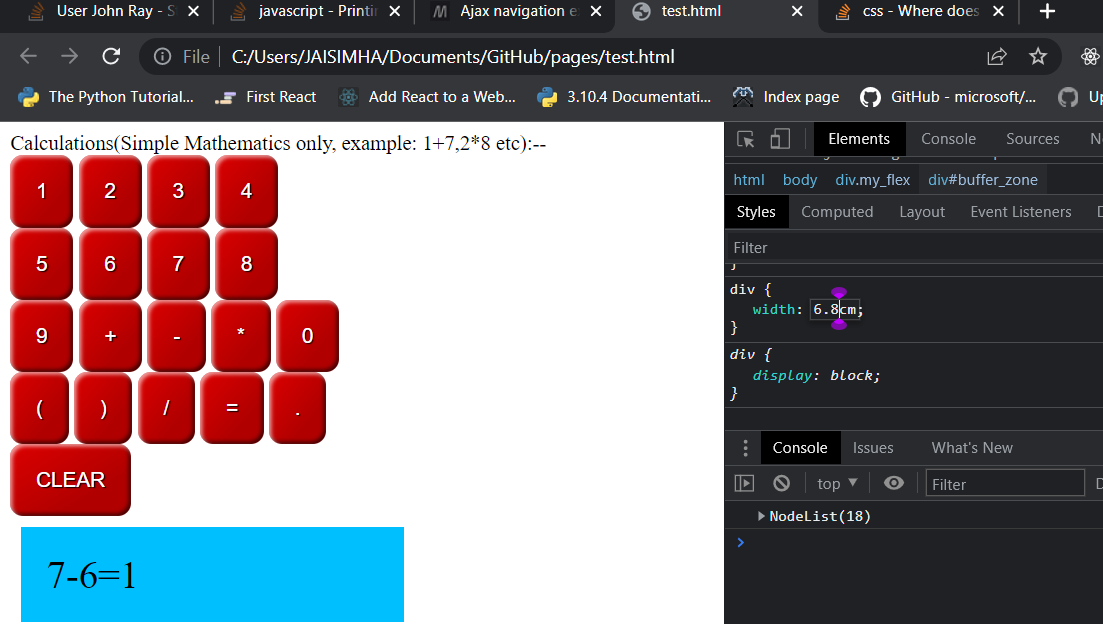
a.setAttribute("href", array[0]);

a.innerHTML = array[0];

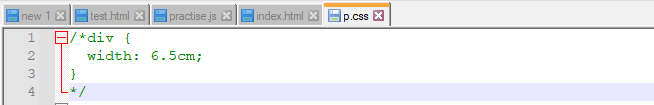
x.appendChild(a);

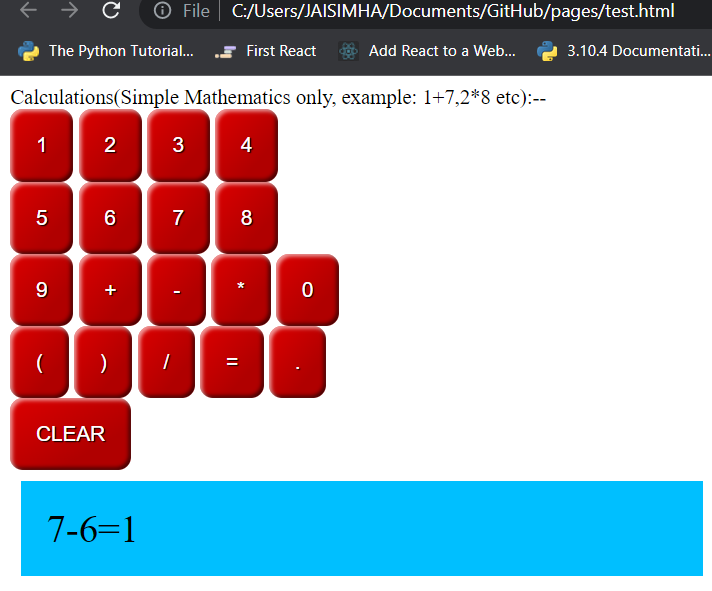
Changing width of DIV element is a WORTH IT!! :



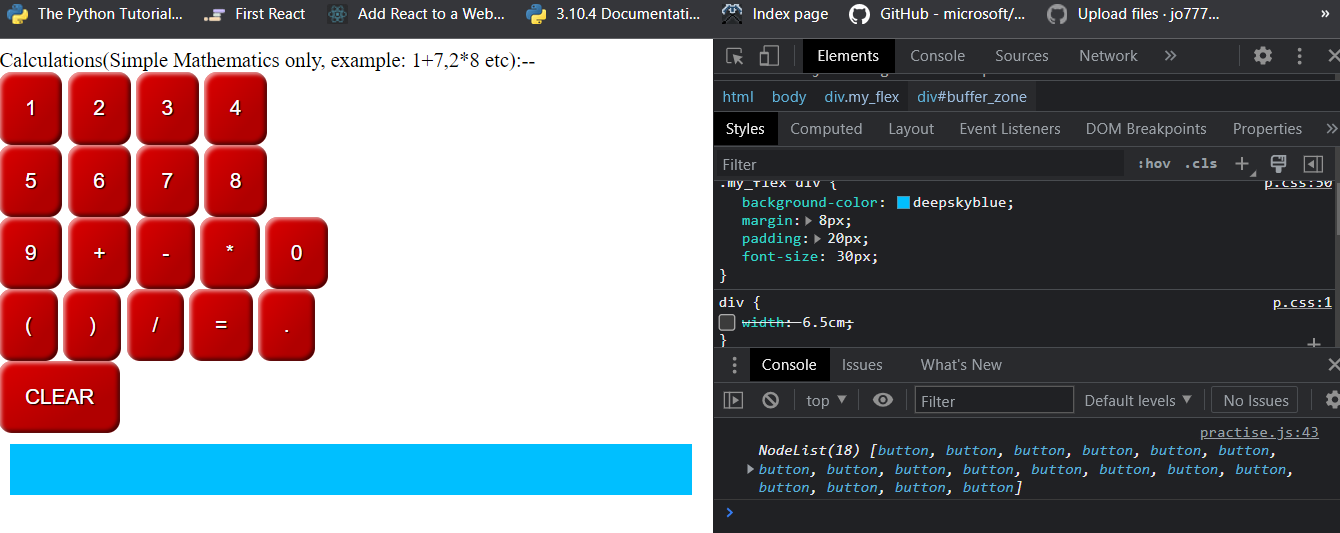


Next don’t set DIV width(this will occupy the whole window):

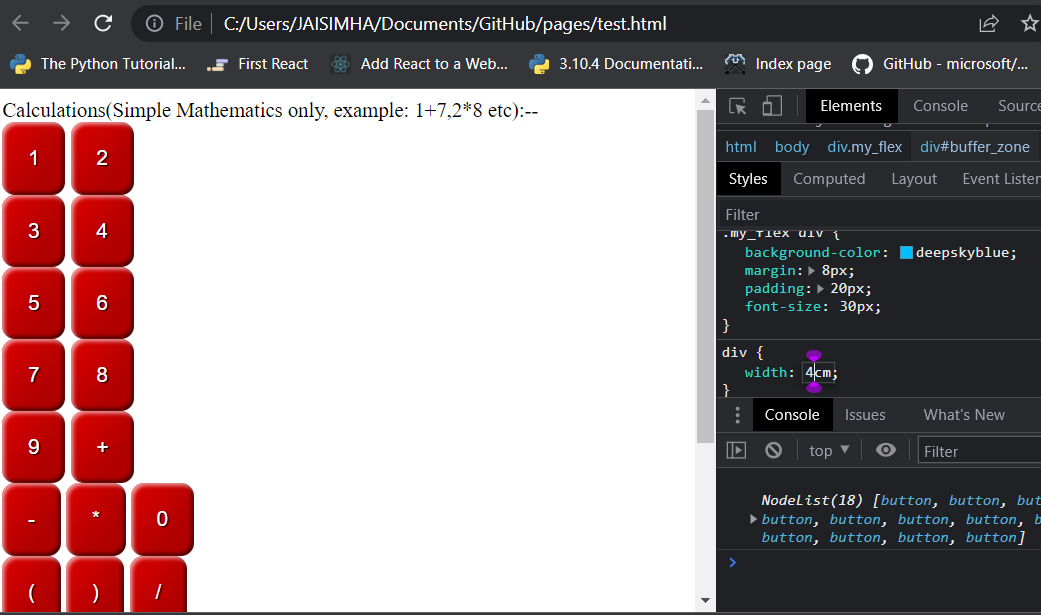




Next using chrome developer tools try changing the values(make div width to 0 and you can get it confirmed by seeing the strikethrough line:



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



### [**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>

Run code snippetHide resultsExpand snippet

I have made a working Javascript calculator for basic mathematics operations,It take in a integer or float and performs the mathematical evaluation;I feel some functions in the javascript like Calculat are duplicated. Please suggest some code optimisation ideas (for example for action event listeners or any other code)

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))

ocument.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=('');

});

});