1. What is HTML?

* HTML stands for Hyper Text Markup Language
* HTML describes the structure of a Web page

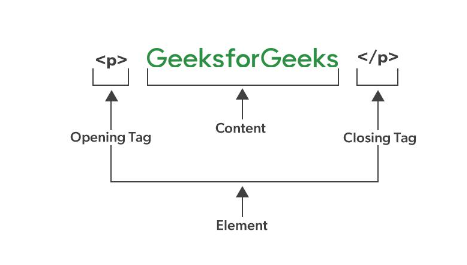
2. What are semantic tags in html

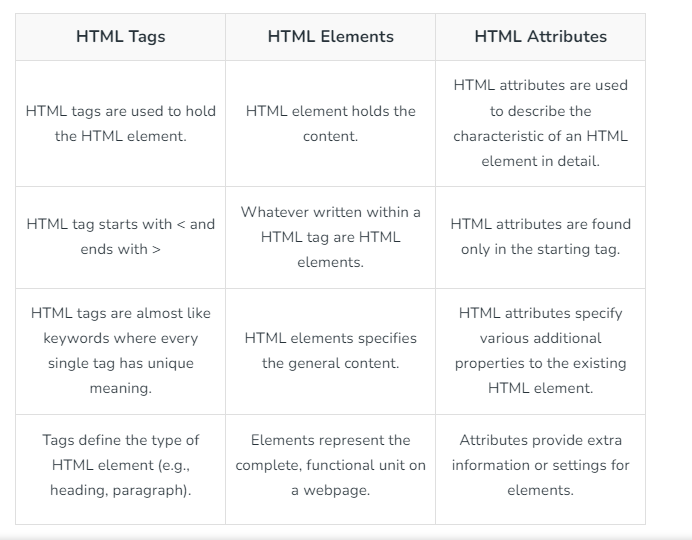
Semantic HTML tags are tags that define the meaning of the content they contain. Providing meaning to developer

* For example, tags like **<header>**, **<article>**, and **<footer>** are semantic HTML tags. They clearly indicate the role of the content they contain.
* Semantic HTML tags are important for [SEO](https://www.semrush.com/blog/what-is-seo/) (search engine optimization) because they indicate the role of the content within the tags.

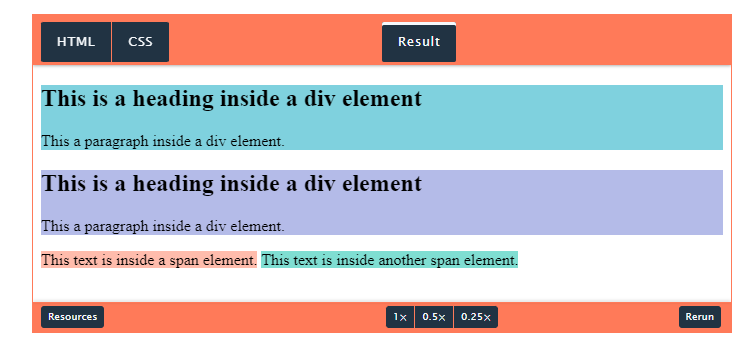
3. Difference between element and tag

HTML Elements contain a starting tag, content, and an ending tag.



­

4. Difference between div and span

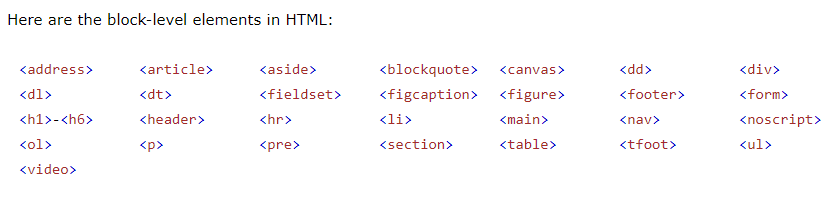




5. Inline and block level elements

## Block-level Elements

* A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.
* A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

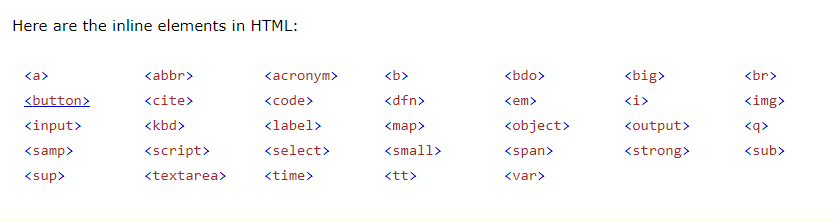


## Inline Elements

An inline element does not start on a new line.

An inline element only takes up as much width as necessary.

This is a <span> element inside a paragraph.



6. Difference between canvas vs svg in html

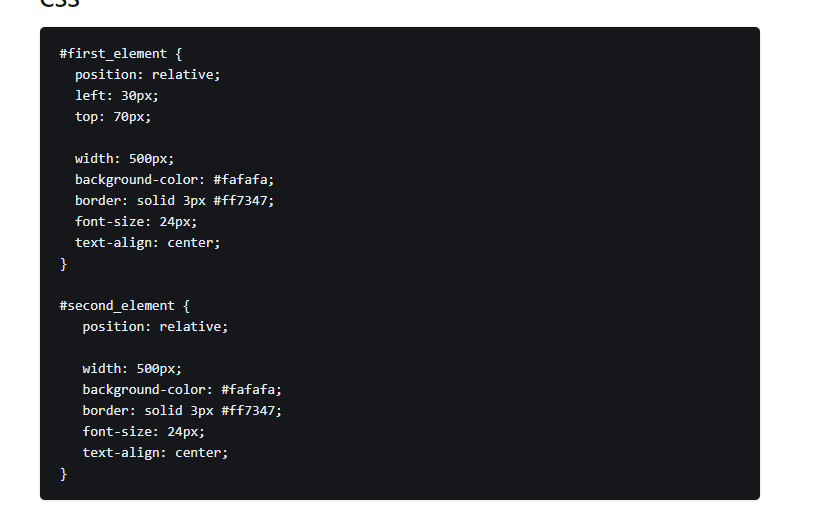
SVG is used to create vector-based graphics, whereas Canvas can render both vector and raster graphics. Canvas is better for quickly rendering graphics and animations with less control than SVG.

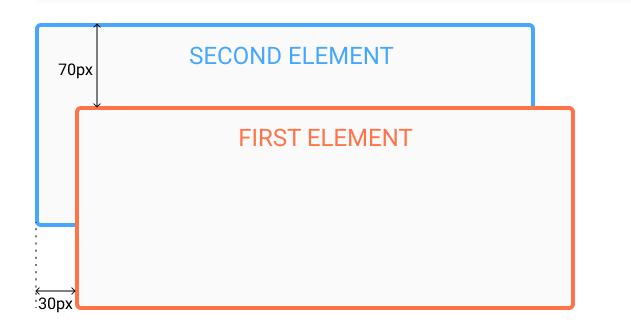
7. Default value of position property in css

the default value is set to static. Any element that has its position value altered to anything other than static will show over those that are set to static.

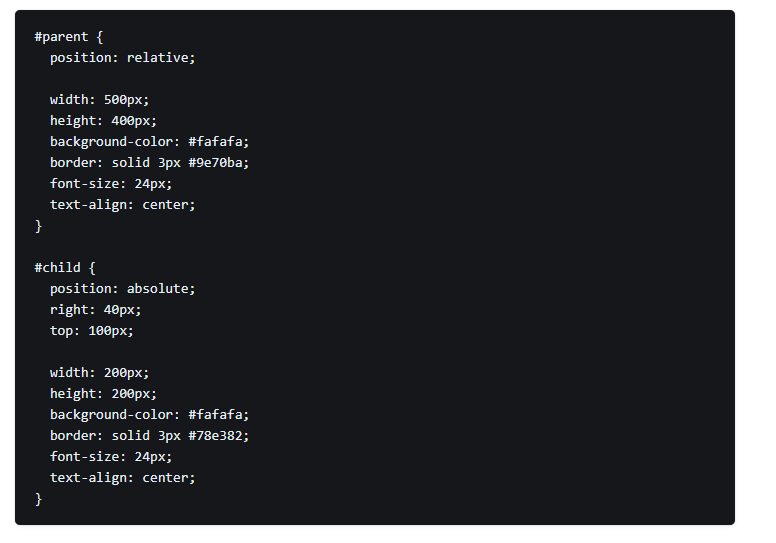
8. Difference between position relative and absolute

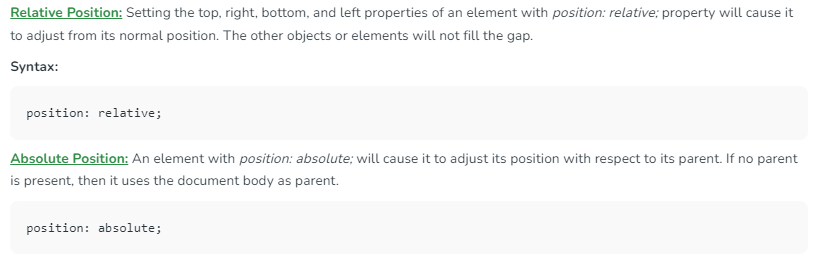


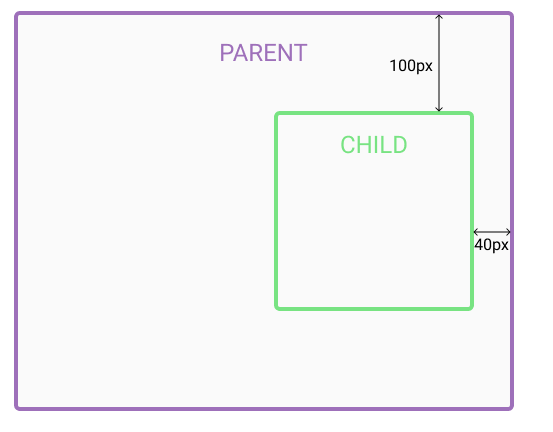






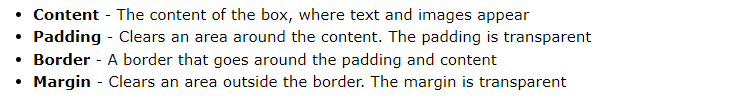


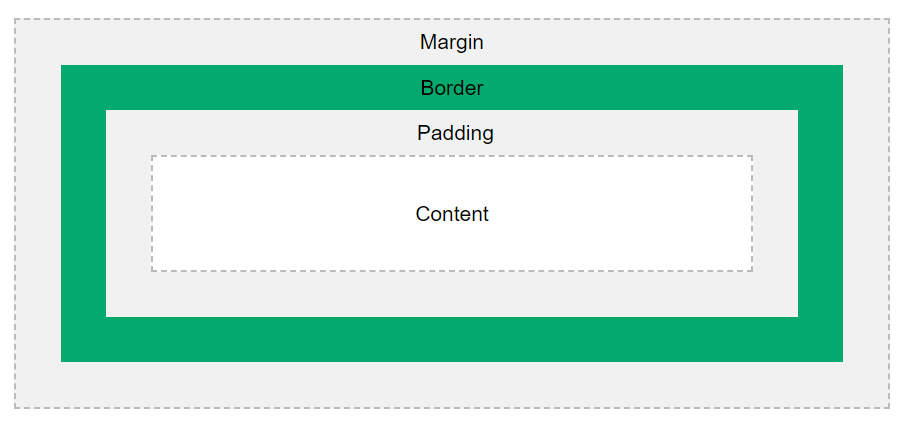




9. What is flex box

Layout module. If the parent has display flex property then their child were becomes block level elements.

10. What is box model in css 



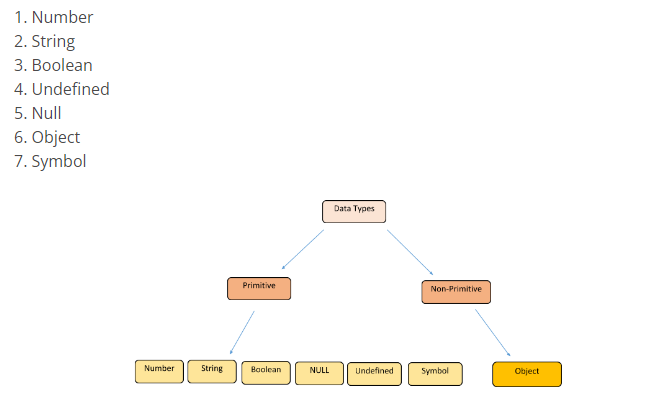
11.what is box-sizing

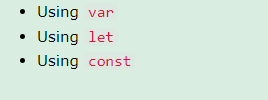
The box-sizing property allows us to include the padding and border in an element's total width and height.

* **border-box:**In this value, not only width and height properties are included but you will find padding and border inside of the box for example
* **content-box:**This is the default value of box-sizing. The dimension of element only includes ‘height’ and ‘width’ and does not include ‘border’ and ‘padding’ given to element. Padding and Border take space outside the element.

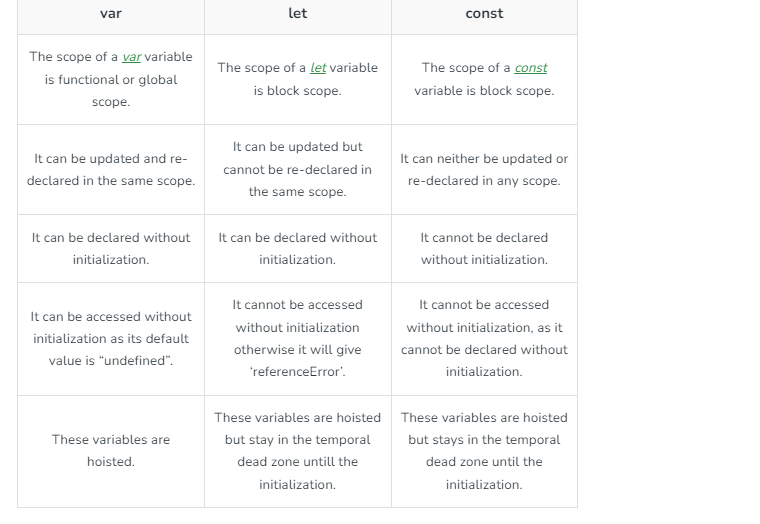
12. Why do we need javascript

to add interactivity and features to improve the user experience and make the internet much

13. Data types in javascript 

14. Ways of creating variables in javascript 

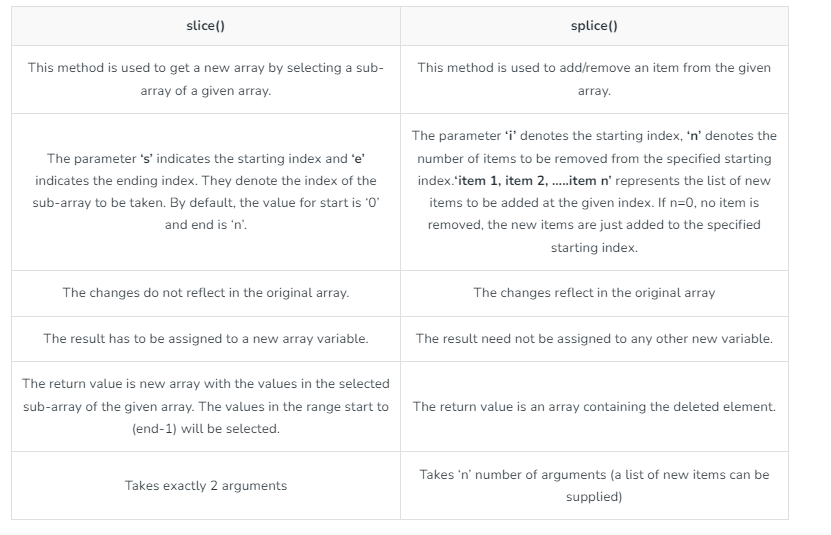
15. Difference between var let and const in javascript



16. Difference between parameter and argument in javascript

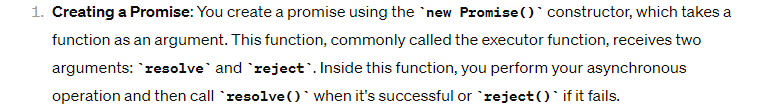
a parameter is a variable in the function definition, while an argument is the actual value that is passed to the function when it is called.

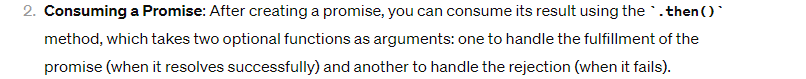
17. Difference between slice and splice in javascript

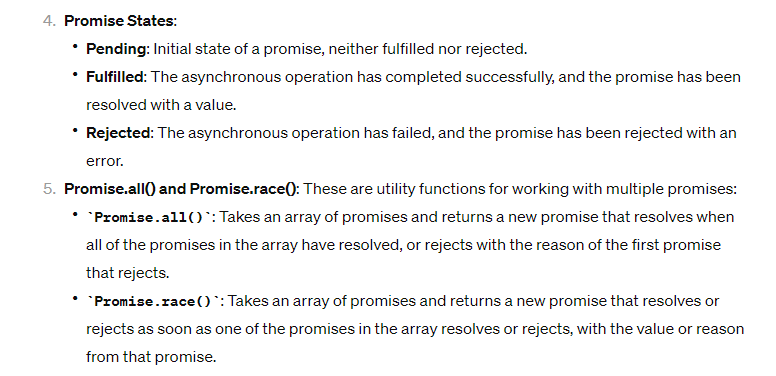


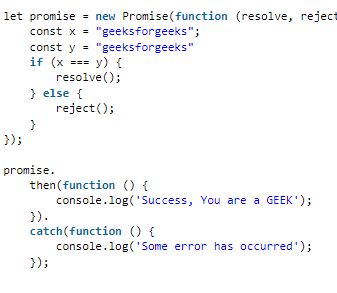
18. Promises in javascript

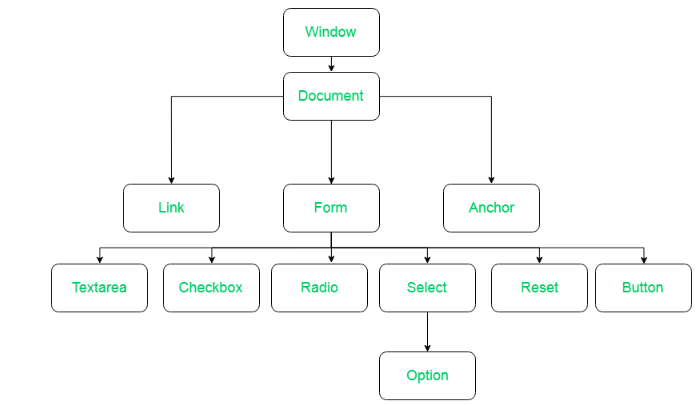
The **Promise** object represents the eventual completion (or failure) of an asynchronous operation and its resulting value.



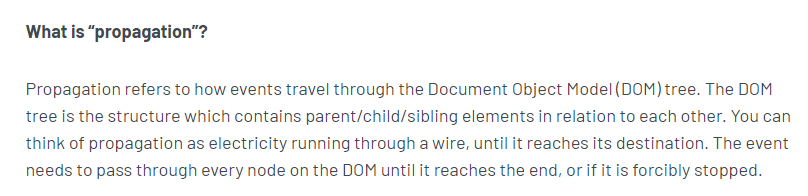


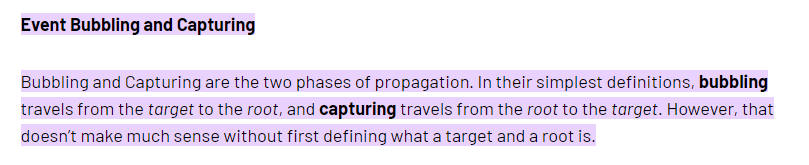


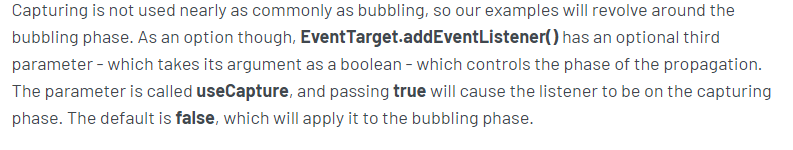


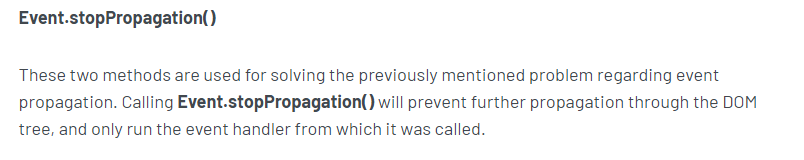
19. Dom in javascript 

20. Event bubbling and event capturing



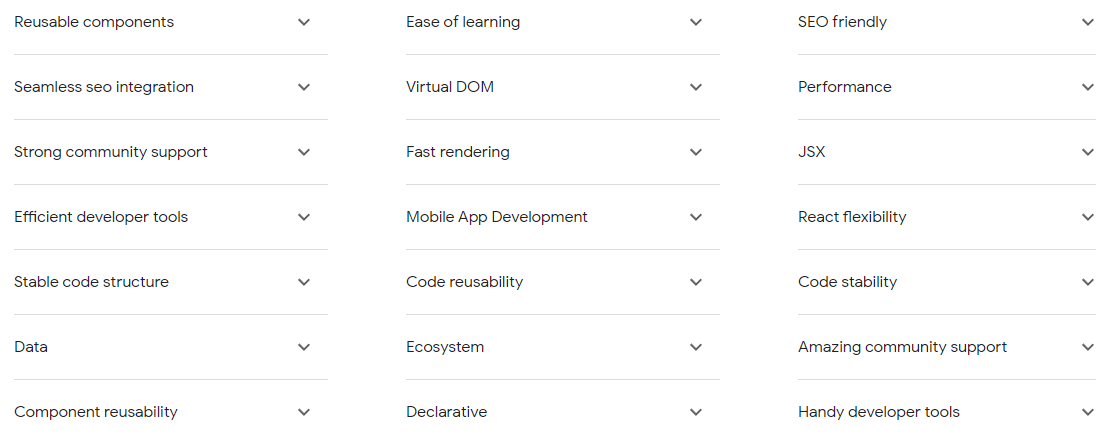








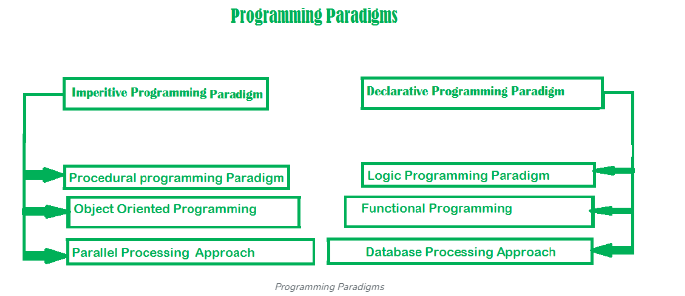
21. Why we need react



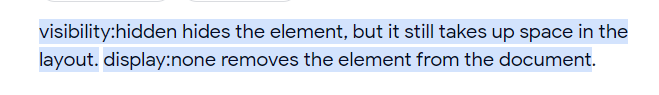
22. Declarative and imperative

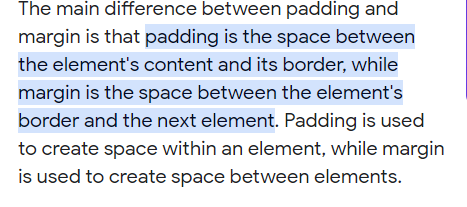
For starters,**imperative programming** refers to a programming paradigm where we provide the computer with step-by-step instructions on **how** to perform a particular task.

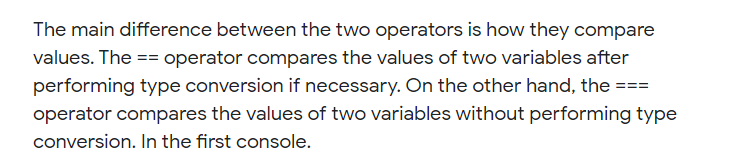
By contrast, **declarative programming** involves specifying **what** result we're expecting from our code. This is mainly achieved through special functions and tools that are provided by different frameworks and libraries of a programming language.



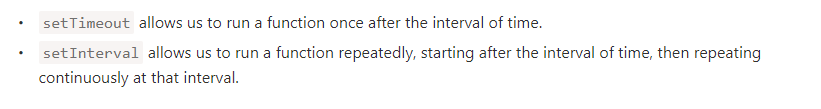
23. Difference between display none vs visibility hidden

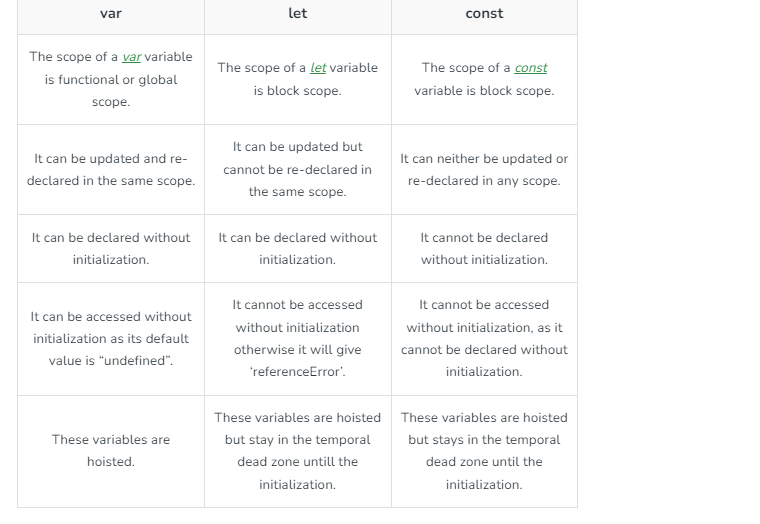


24. Difference between margin and padding 

25. Difference between == and === in javascript 

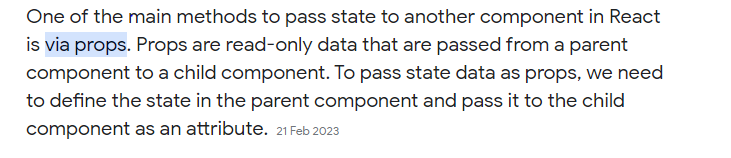
26. Difference between settimeout and setinterval

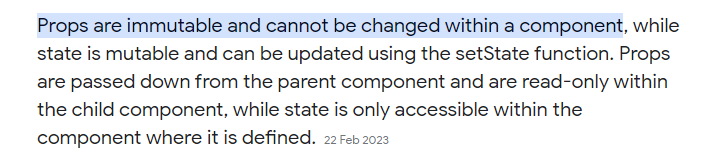


27. Difference between var let and const in javascript 

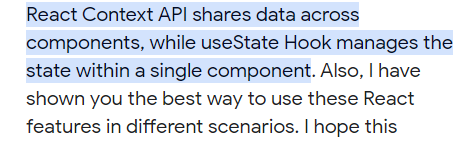
28. Advantages of react

* Virtual dom
* Seo
* Reuseable componenet
* Declarative
* Component based ui

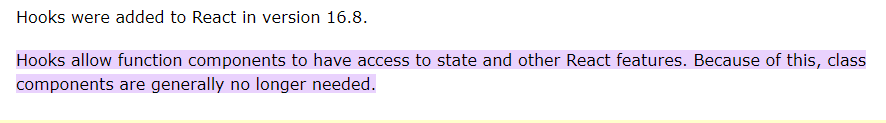
29. How to pass data from one component to another in react js 

30. State and props which is immutable 

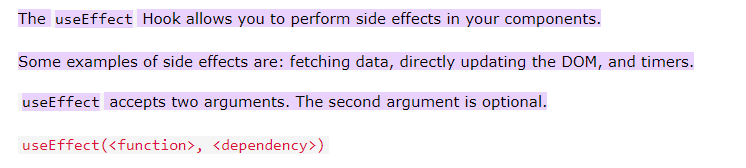
31. Difference between usestate and conteext api



32. What are react hooks

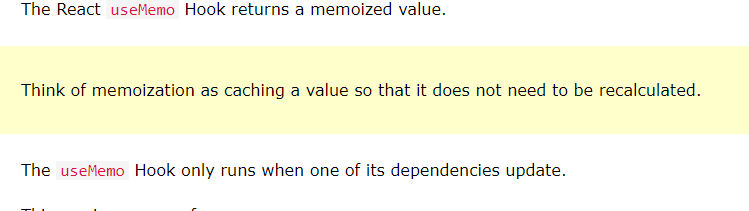


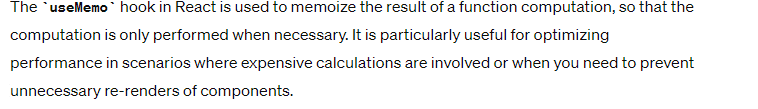
33. What is use of useeffect in react

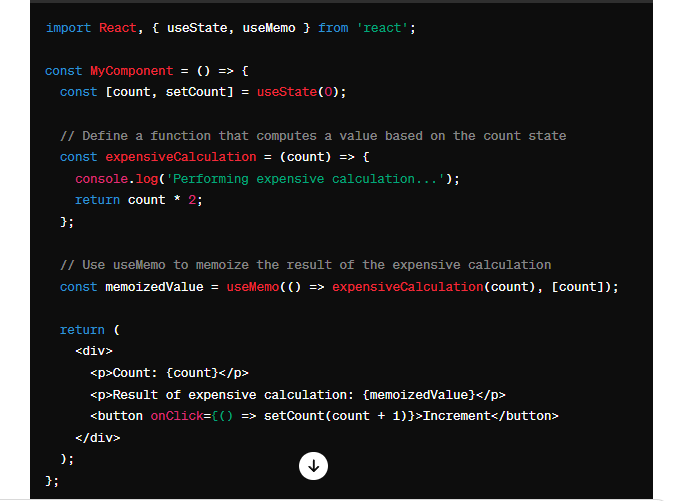


we put side effects into componentDidMount and componentDidUpdate.

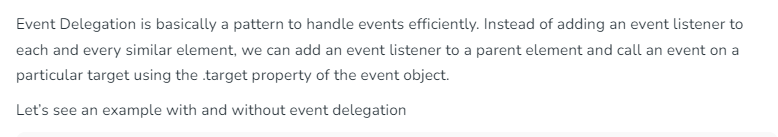
34. Usememo in react js





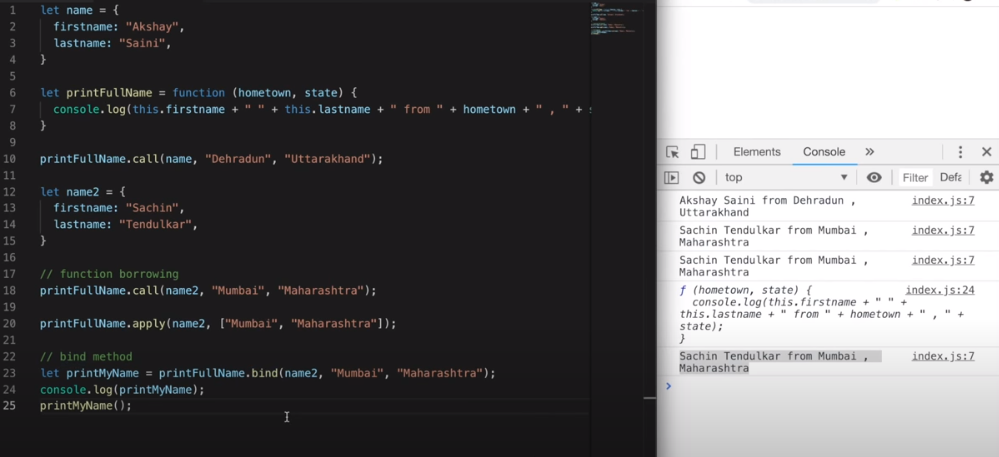


35. What is Event delegation

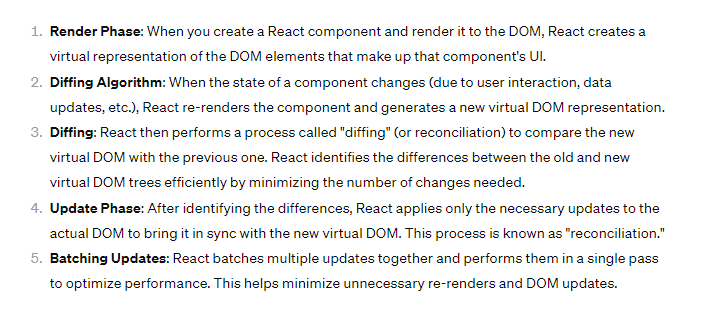


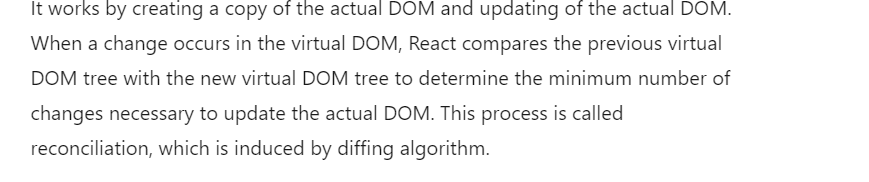


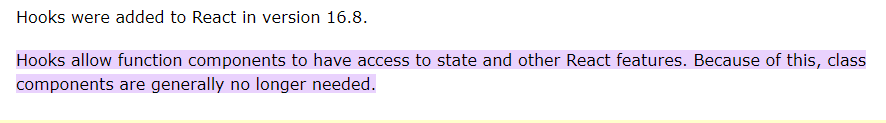
36. Difference between call apply bind in javascript



37. Explain virtual dom in react



38. What are react hooks

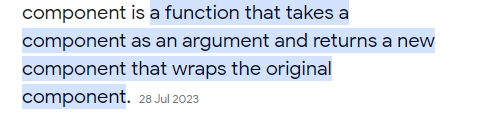


39. difference between usememo and usecallback

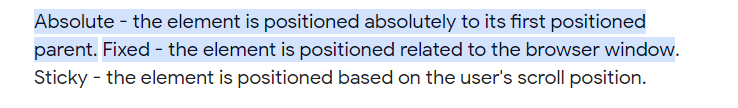




40. Higher order componenets



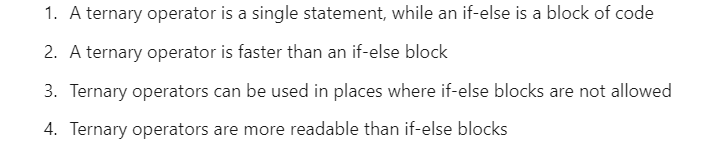
41. Difference between position absolute and fixed 



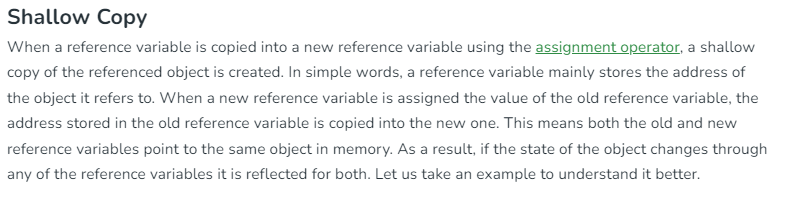
42. Difference between if else and ternary operator in javascript

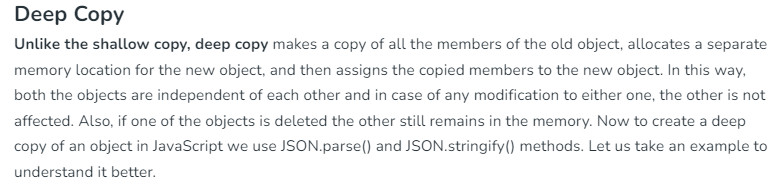
Const statement= a > b? “hi” : “no”

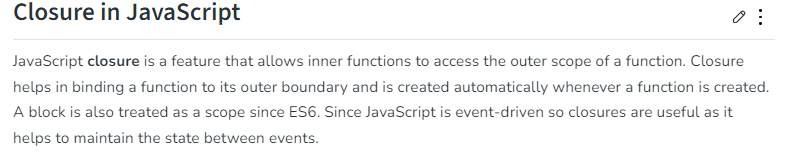
* + - statements can be stored but expression cannot

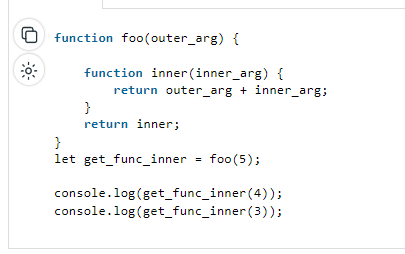


43. Shallow copy and deep copy in javascript

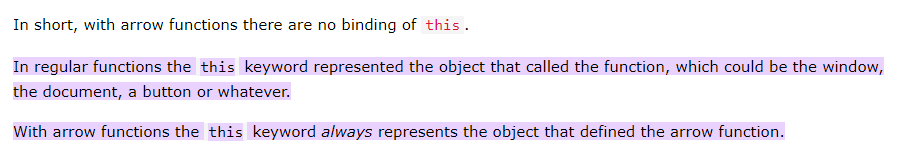


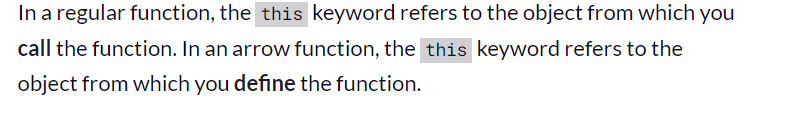


44. What is closure in javascript 



45. This keyword in arrow function





46.

Background blur property

Background Filter property

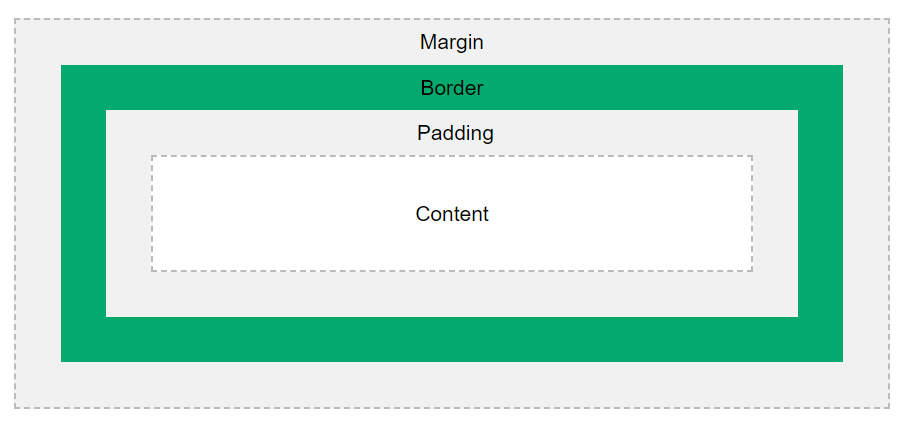
Transform translate

Perspective

Clip path

Position fixed relative to its parent

47. What is box model in css

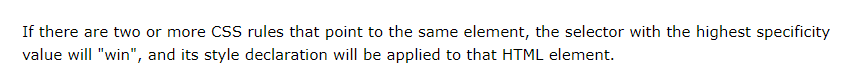


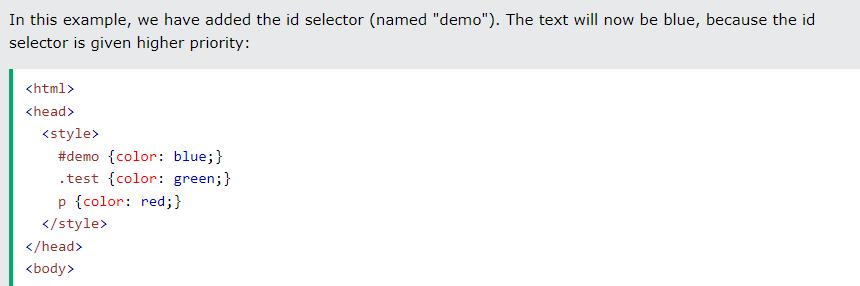
48. What is box-sizing

The box-sizing property allows us to include the padding and border in an element's total width and height.

* **border-box:**In this value, not only width and height properties are included but you will find padding and border inside of the box for example
* **content-box:**This is the default value of box-sizing. The dimension of element only includes ‘height’ and ‘width’ and does not include ‘border’ and ‘padding’ given to element. Padding and Border take space outside the element.

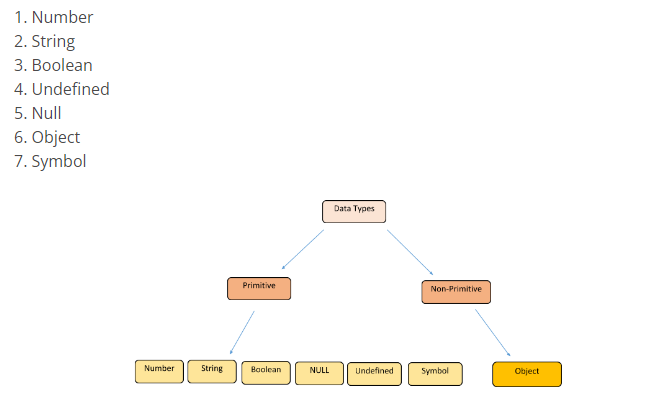
49. Specifycity in css



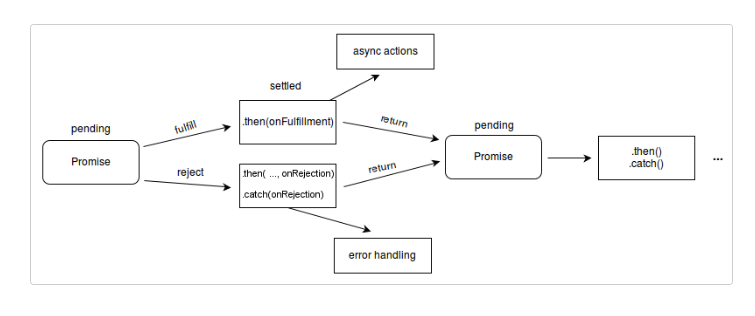


50. Difference between div and sapan

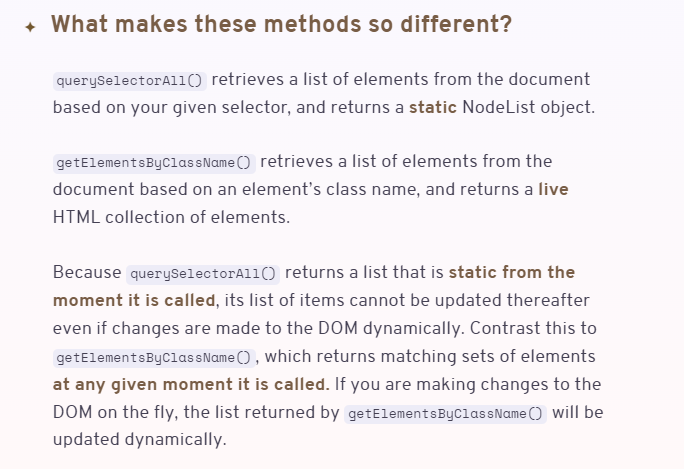


51. Data types in javascript 

52. Promises in javascript



53. Difference between getelementsbyclassname vs queryselector

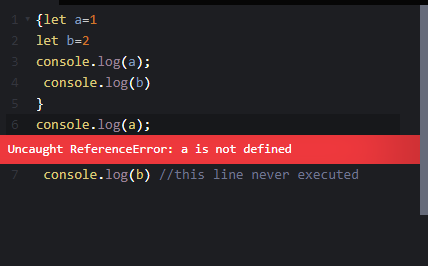


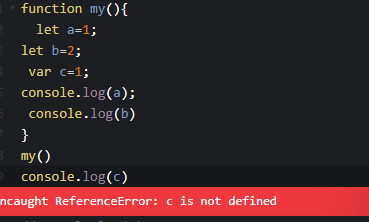
54. Why we need react

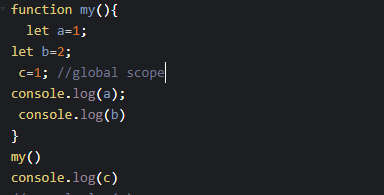
Js- imperative

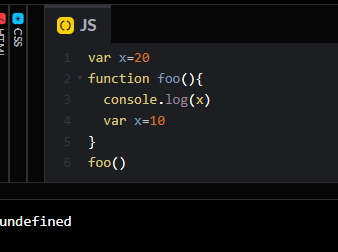
React- declarative

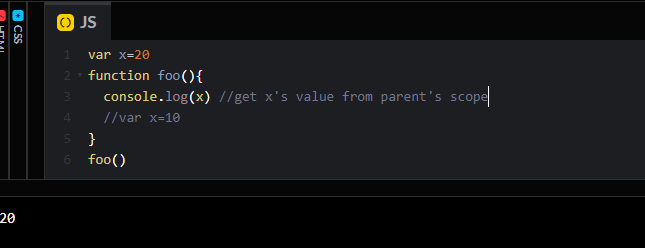
55.ouput

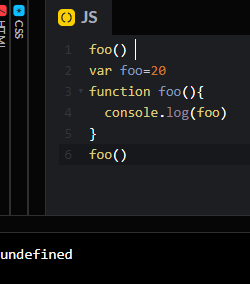








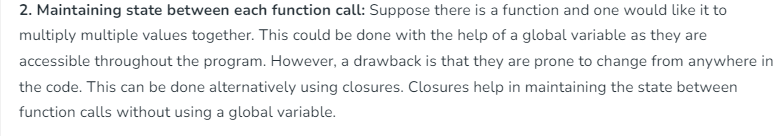


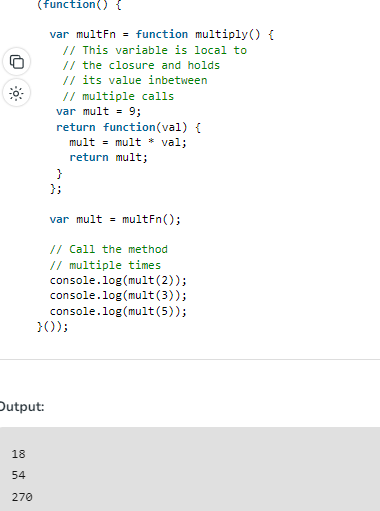
 

56. Closure

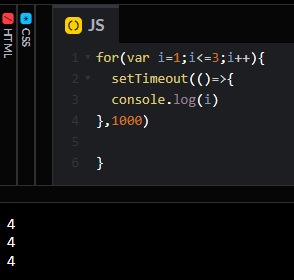
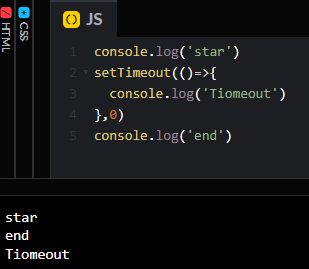


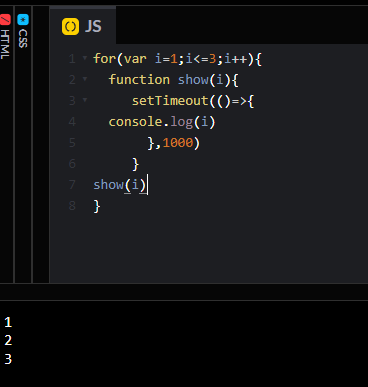


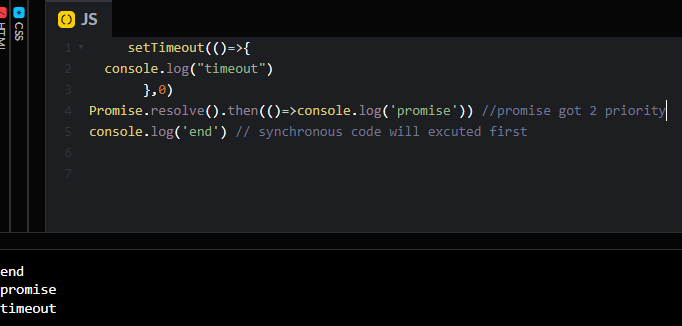




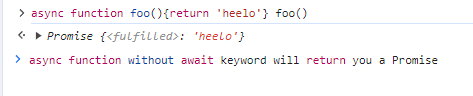
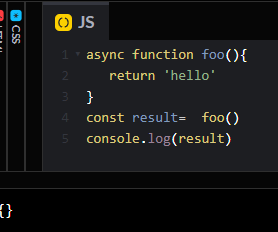
57. Timeout question

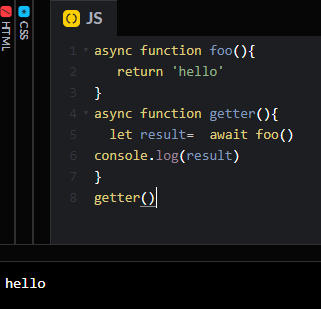


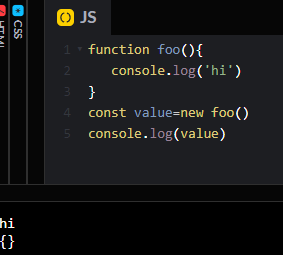


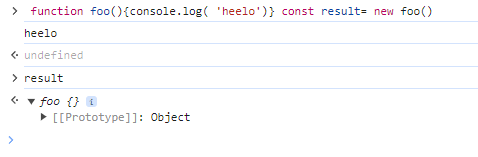


58. Async

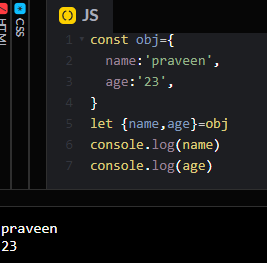


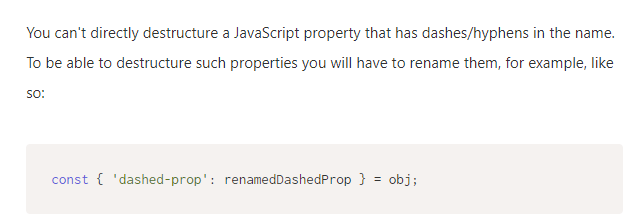


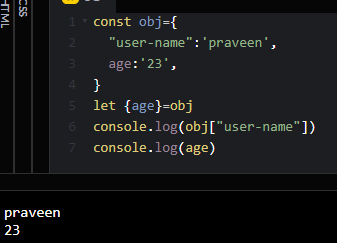
59. Creating function’s object 



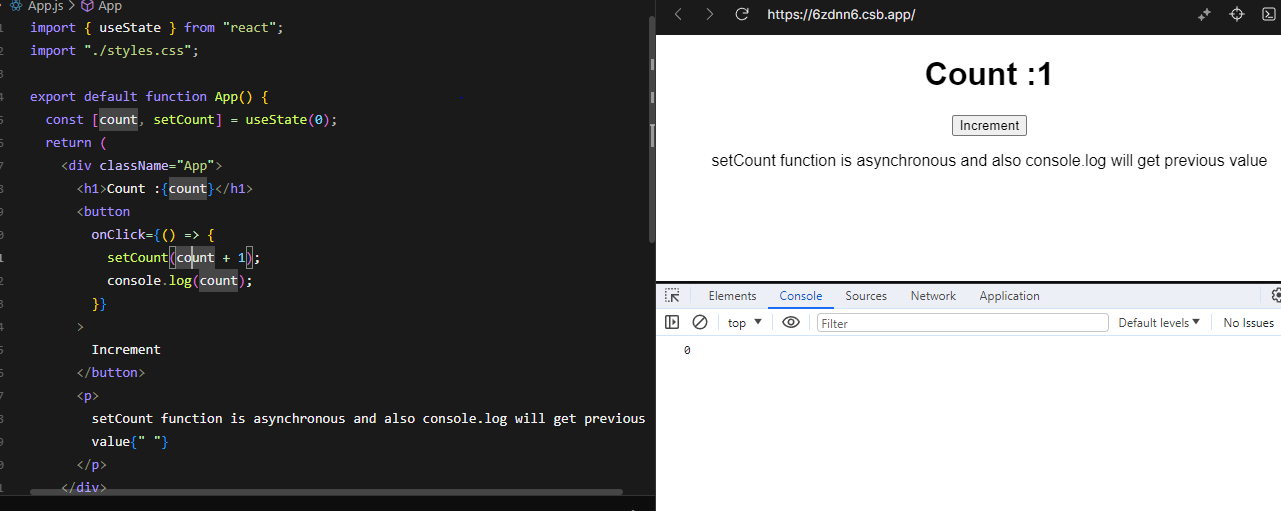
60. Destucturing object



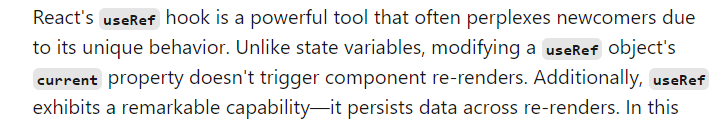


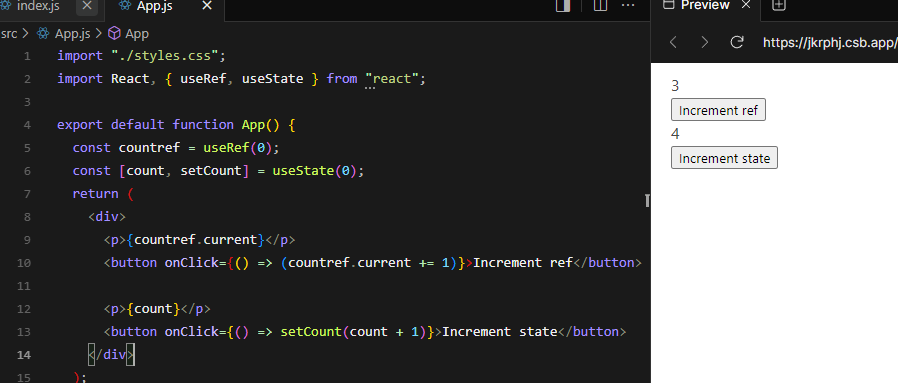


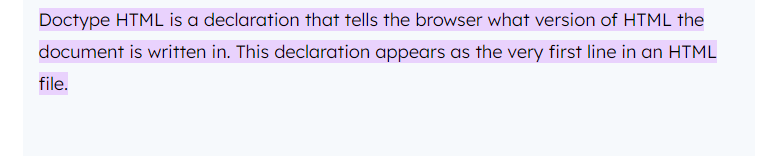
61. React setCount() vs consle.log()



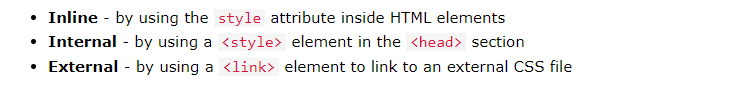
62. useRef not re-render

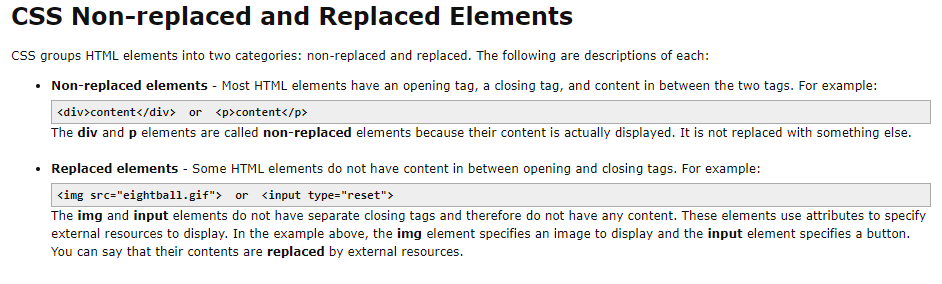




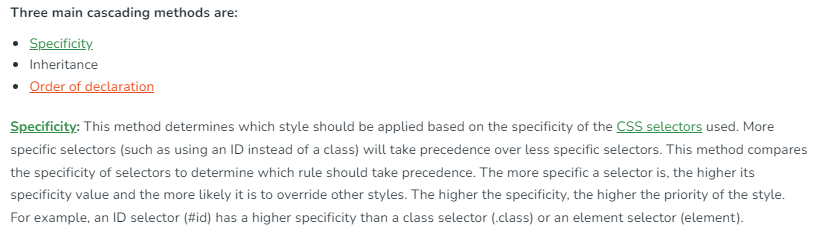
63. Why doctype html is used 

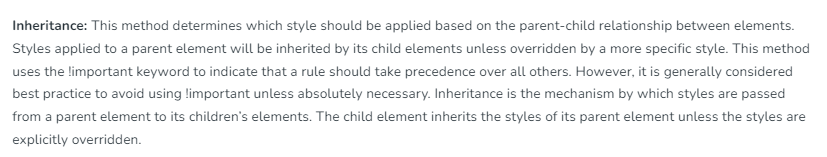
64. Ways of adding css to html

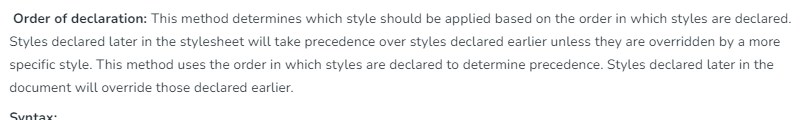


65. Replaced inline element non replaced inline elements in html 

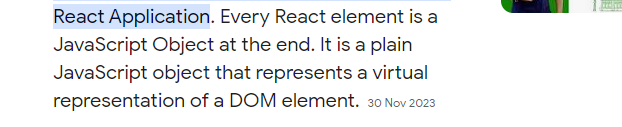
**Inline replaced elements can have height and width attribute**

66. cascade rule in css 

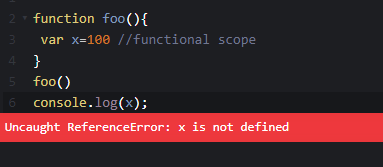


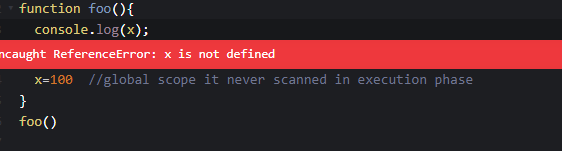


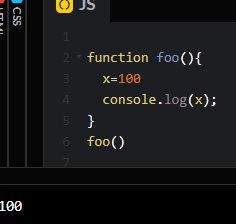
67. What is react elements



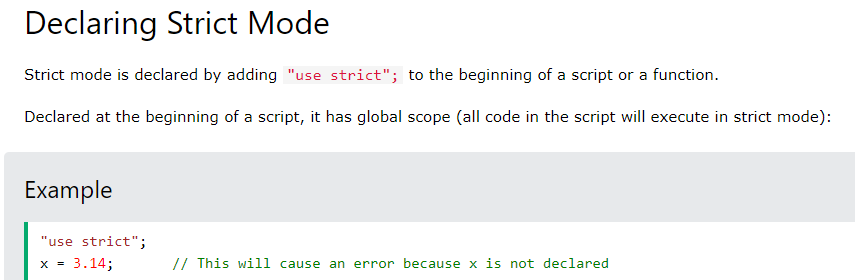
68. Funtional scope vs block scope

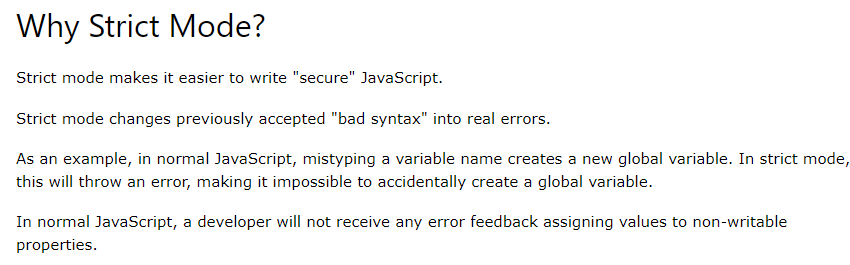




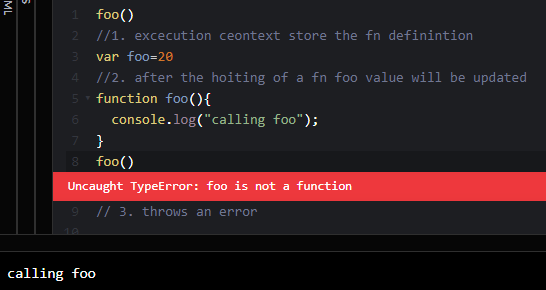


69. Strict mode

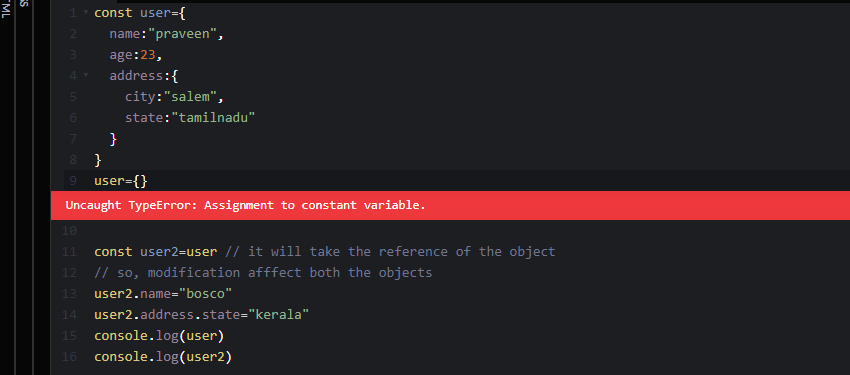




70. Same name for function and variable



71. Object reference 





72. Check box

[**check box**](%3ciframe%20src=%22https:/codesandbox.io/embed/7x7x9l?view=Editor+%2B+Preview&module=%2Fsrc%2FApp.js%22)

import "./styles.css";

import React, { useState } from "react";

export default function App() {

  const [checked, setChecked] = useState({

    check1: false,

    check1: false,

    check2: false,

    check3: false,

  });

  return (

    <div className="App">

      <div>

        <input

          type="checkbox"

          checked={checked.check1 && checked.check2 && checked.check3}

          onChange={() => {

            setChecked({

              ...checked,

              check1: !checked.check1,

              check2: !checked.check2,

              check3: !checked.check3,

            });

          }}

        />

        <label>Check all </label>

      </div>

      <div>

        <input

          type="checkbox"

          checked={checked.check1}

          onChange={() => {

            setChecked({

              ...checked,

              check1: !checked.check1,

            });

          }}

        />

        <label>1 </label>

      </div>

      <div>

        <input

          type="checkbox"

          checked={checked.check2}

          onChange={() => {

            setChecked({

              ...checked,

              check2: !checked.check2,

            });

          }}

        />

        <label>2 </label>

      </div>

      <div>

        <input

          type="checkbox"

          checked={checked.check3}

          onChange={() => {

            setChecked({

              ...checked,

              check3: !checked.check3,

            });

          }}

        />

        <label>3 </label>

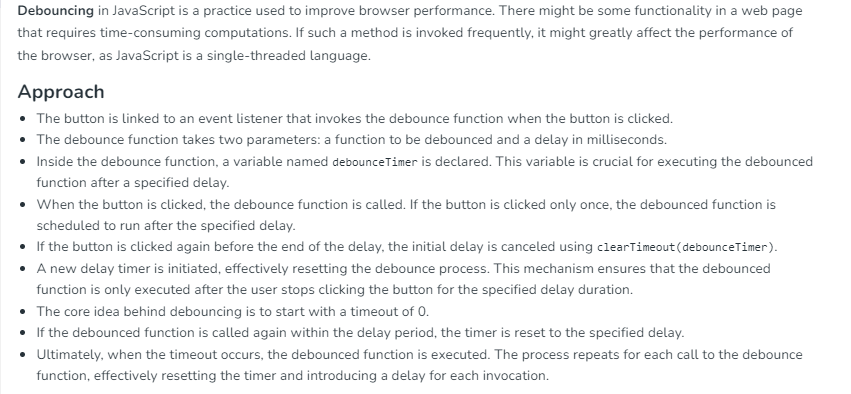
      </div>

    </div>

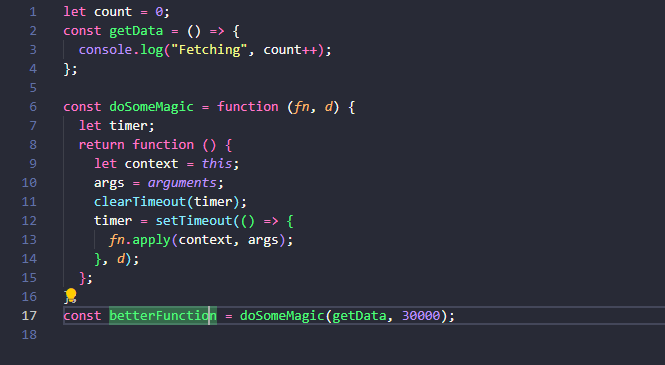
  );

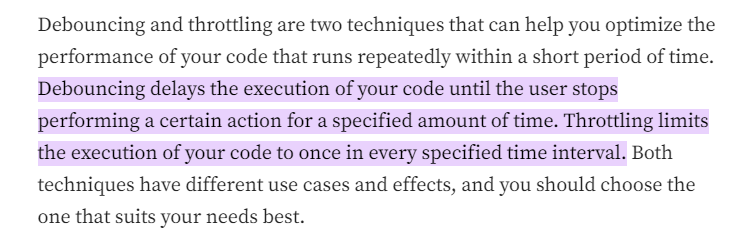
}

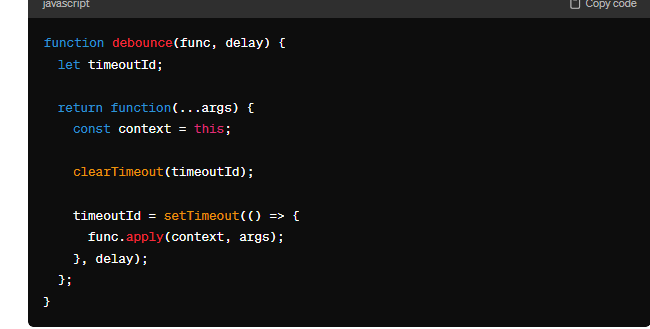
73. Debounce in javascript

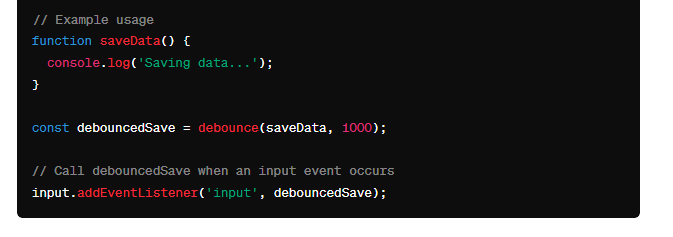


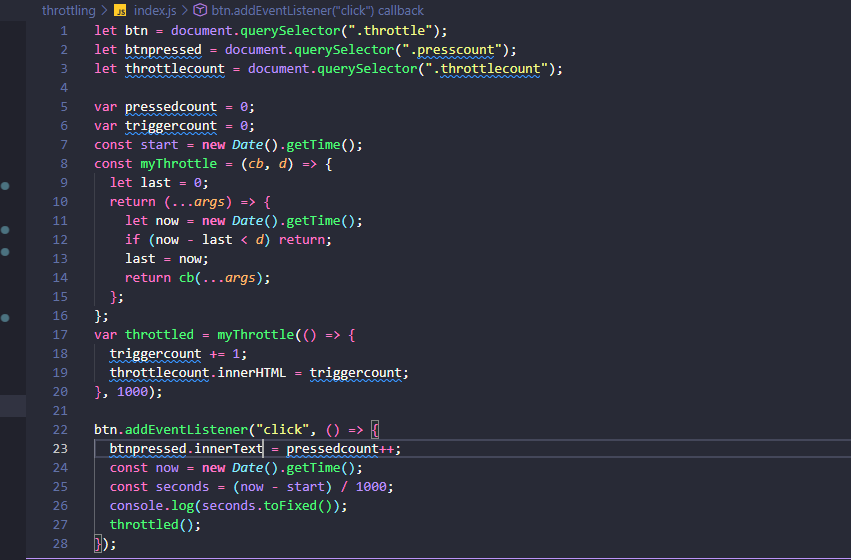
**­**

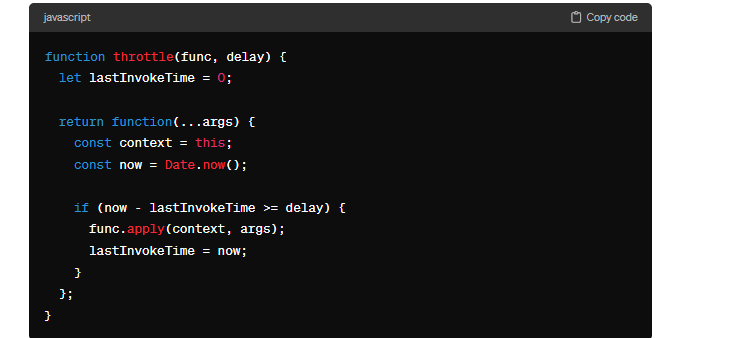


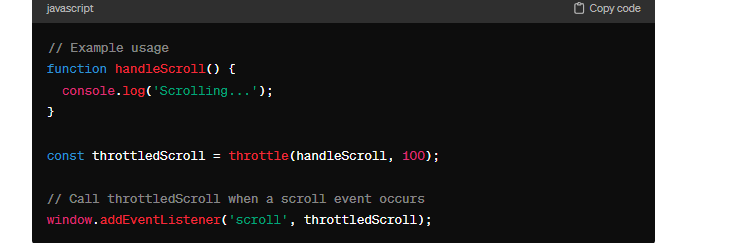




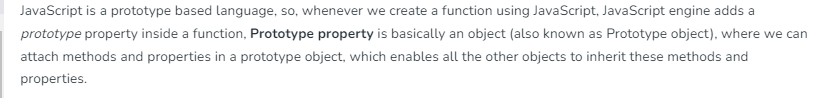


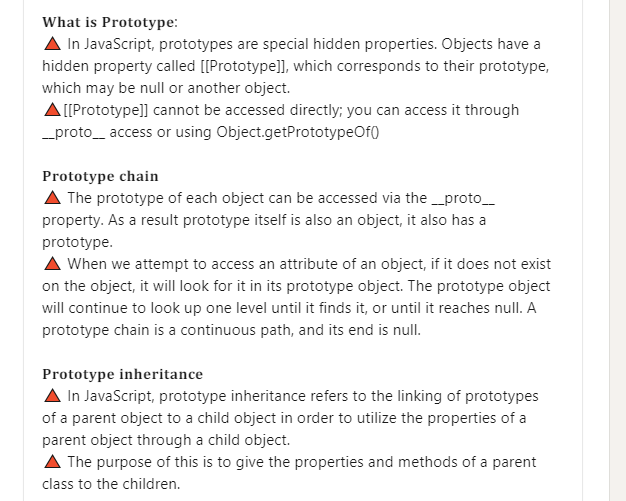
74. Throttling in javascript 

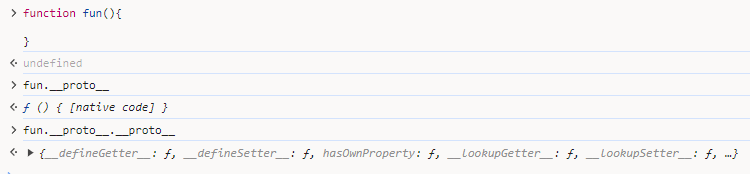


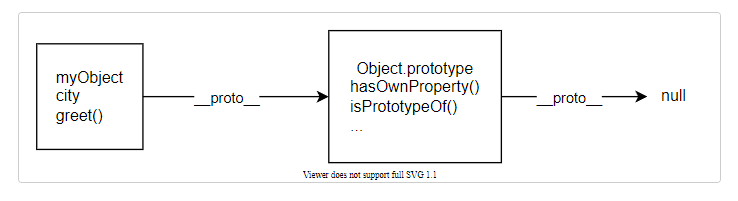


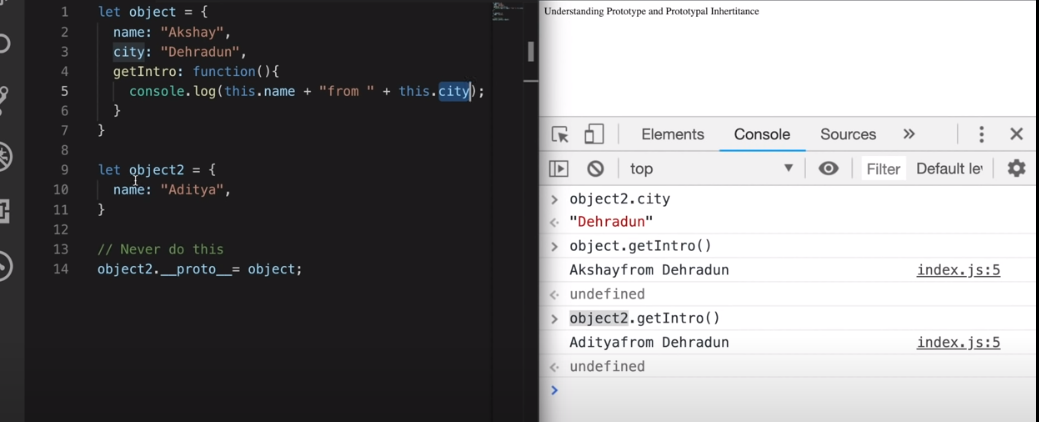
75. Prototype in javascript



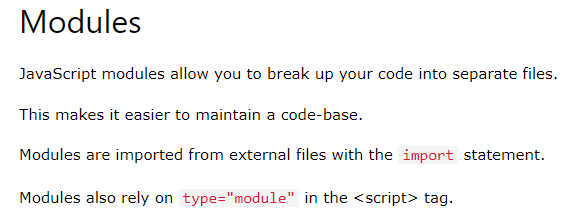


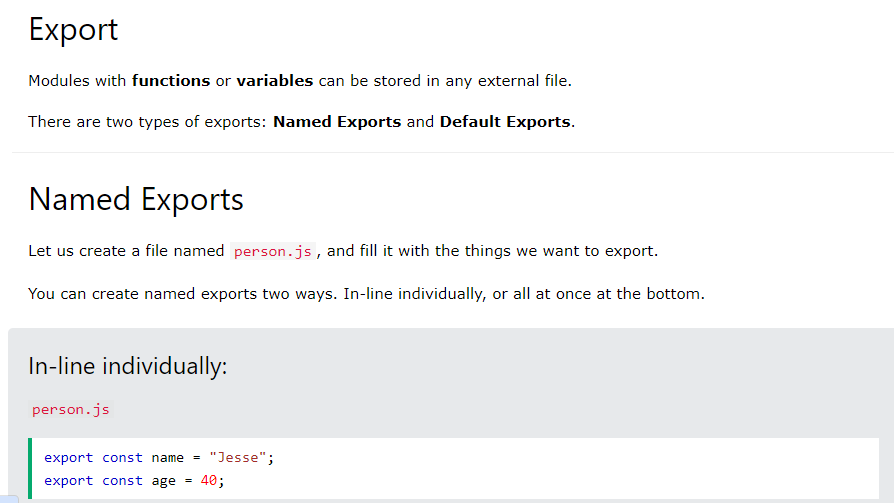


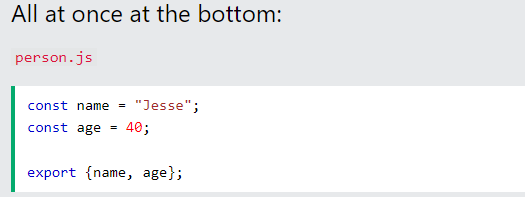




76. Javascript modules import and export

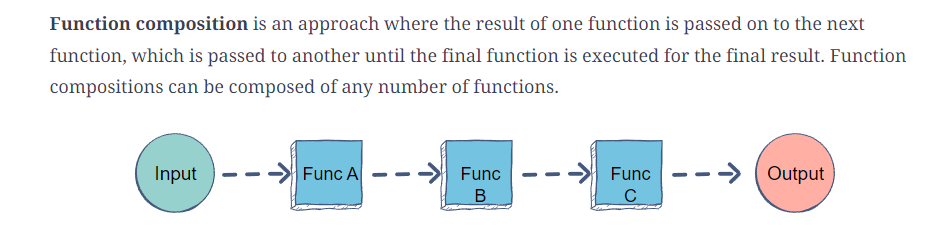


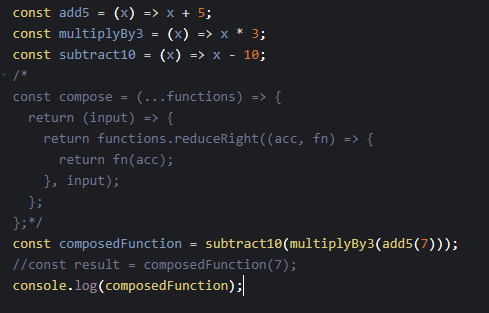


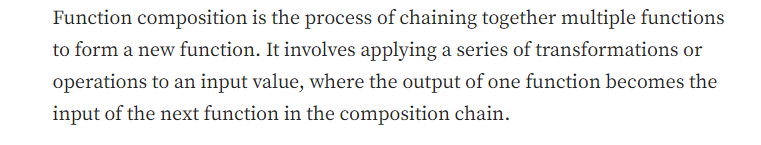


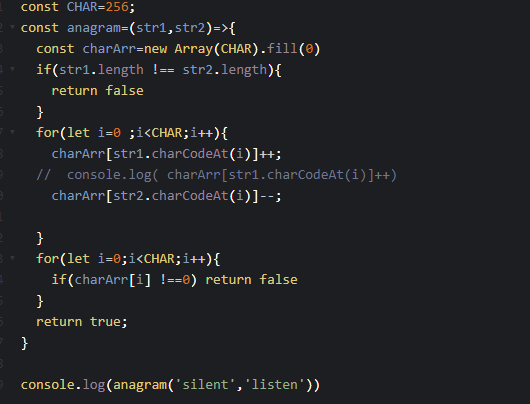


77. Function composition

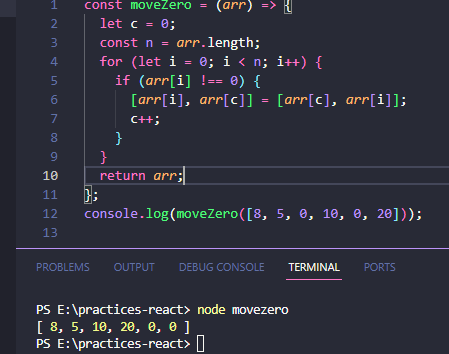


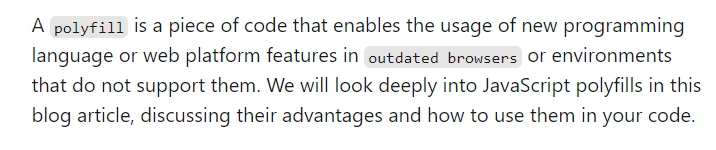


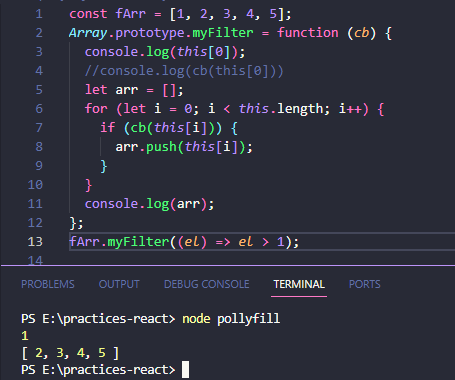


78. Anagram 

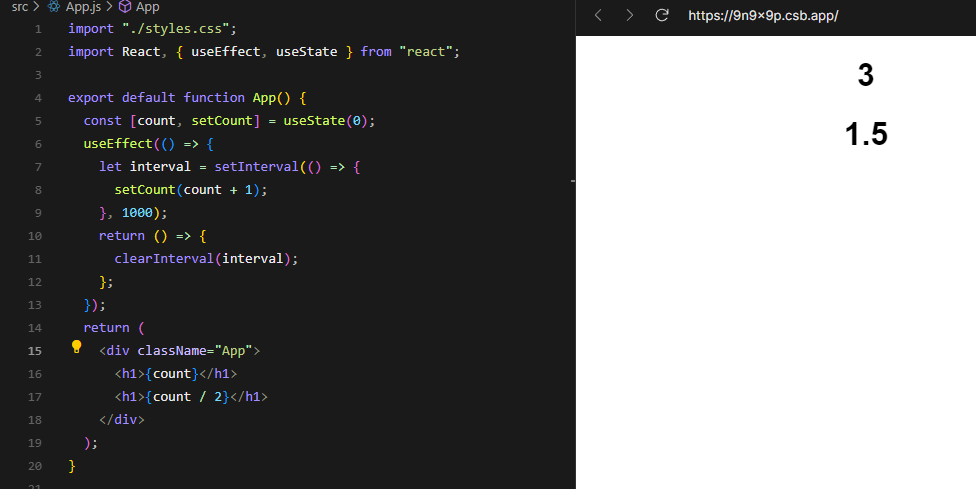
79. Movezero



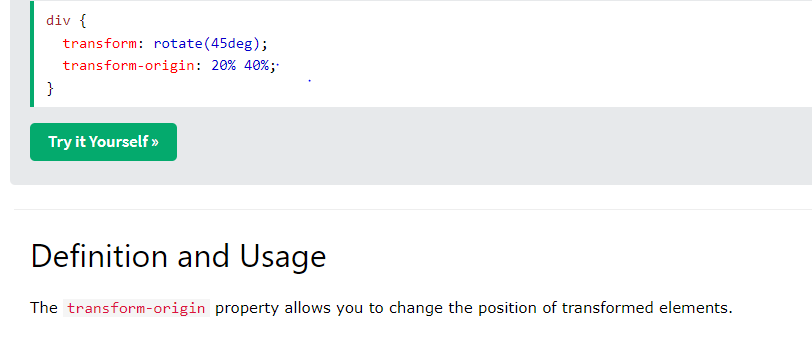
80. Pollyfill in javascript 

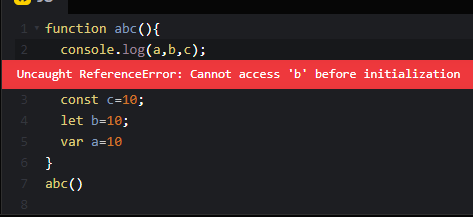


81. Setinterval count 1 and 0.5 with react



82. Transform-origin

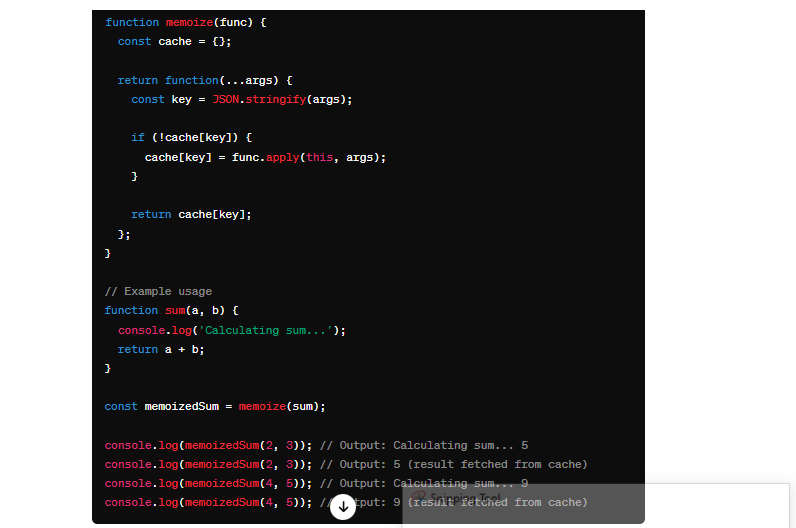


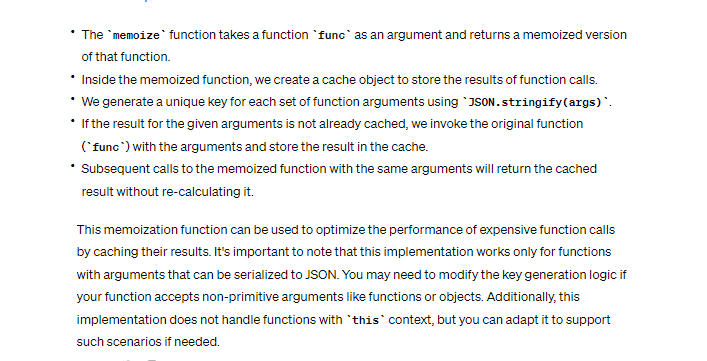
83.hoisting 

84. Implicit binding and explicit binding

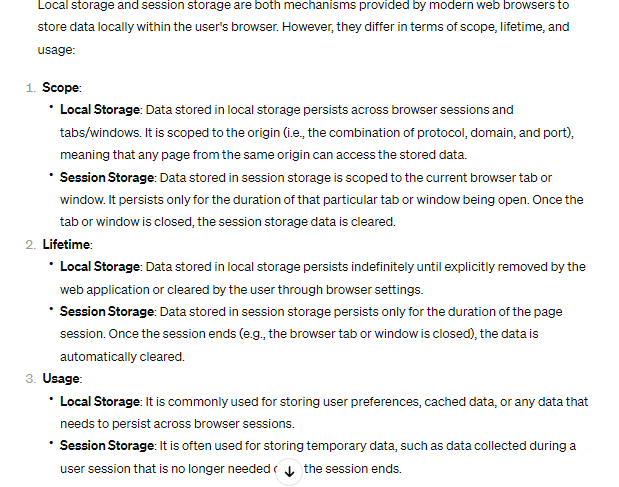


85. Caching/memorize function

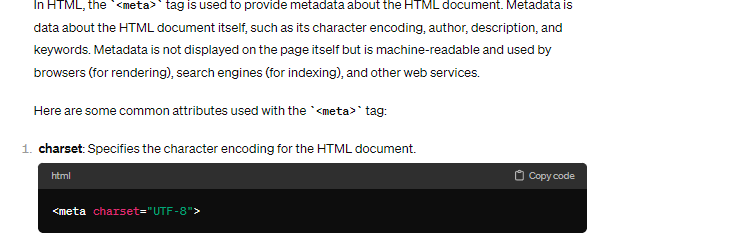




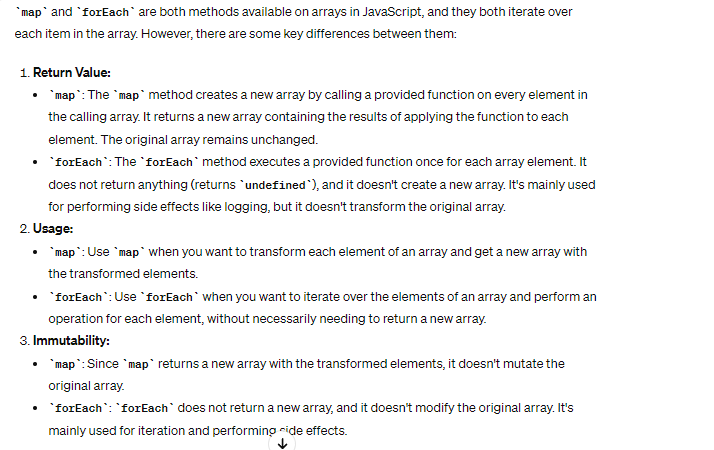
86. Session storage vs Local storage

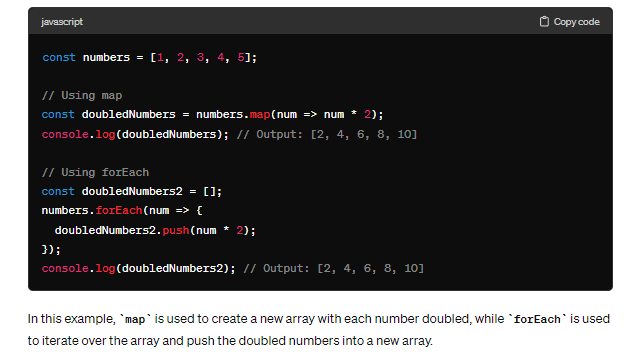


87. Meta tag

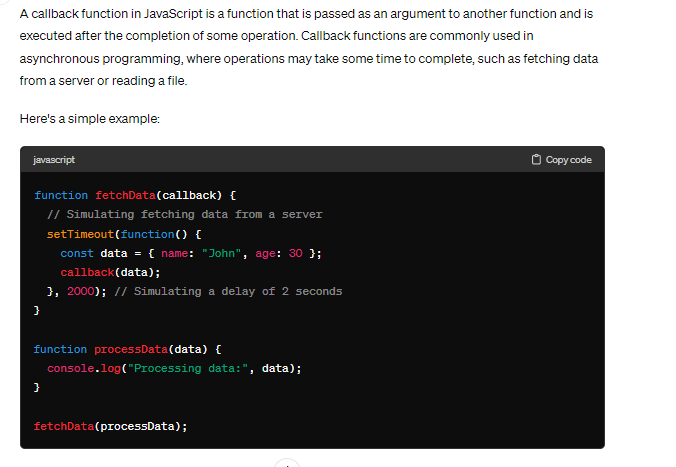


88. Map vs foreach

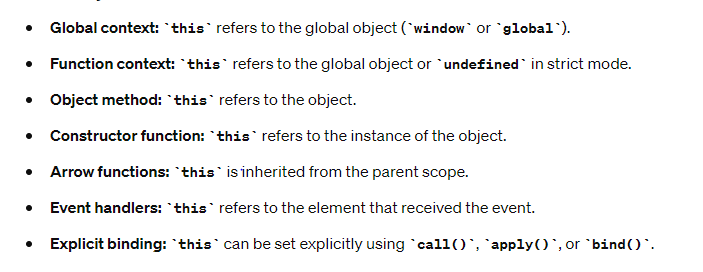




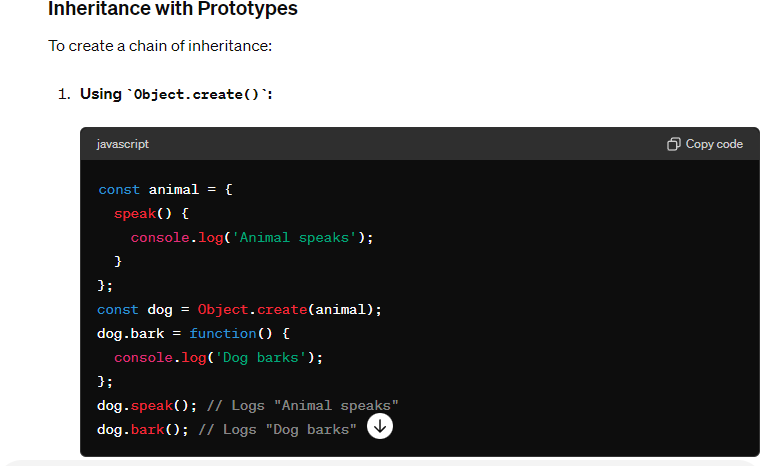
89. Callback function



90. This keyword



91. How prototypal inheritance work



92. Generator functions

