

Day - 12

①

// variables

var age = 24;

age = 35;

age = 'Pigus'

② console.log('Age is', age);

var age = 24;

age = 35

This is like a bad habit
to do before the declaration you are using variable

Not

let

→ var like

→ design

→ use

(age = 6

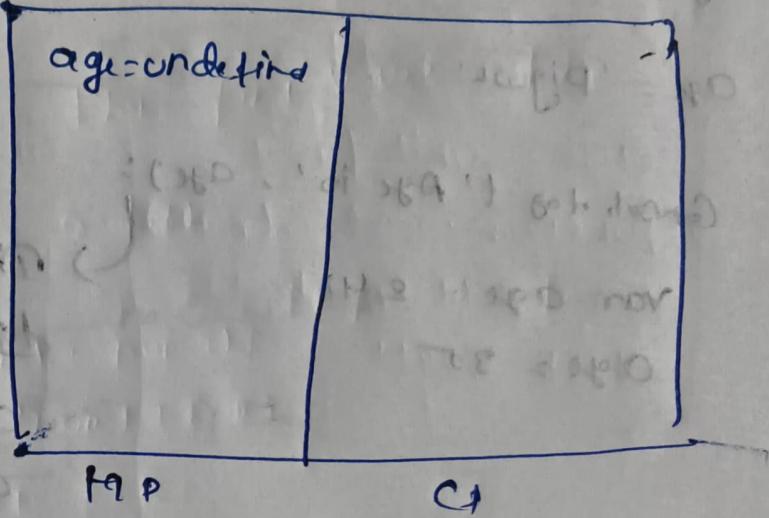
let - hoisted

Yes

Age = 65
Console.log (age);
let age = 24;
Console.log (age);

Age = 45;

Console.log (age);



surgeon : you can not access before initialisation

Temporal Dead zone

↳ diff b - behavior between let and const

variable exist in a scope → but const

→ be aware before the declaration

JavaScript

Object Class

Const

→ No re-assignment → TD2 गति
इसके नीचे को हित नहीं विभाजन करें

Object

II What is a function

II Function - A reusable set of instruction

II A block of code designed to perform
a specific task

function sayHello() {

 console.log('Bye')

}

sayHello();

Note

function का बहुलये रूपों का हो सकता है परन्तु
Parameter नहीं है

function greet(x,y,z) {

 console.log('Hey', x,y,z);

}

 console.log(`Hey \${x} \${y} \${z}`)

↳

String

Interpolation

function add (num1, num2) {

 console.log(`Result is \${num1 + num2}`);

}

(add(2, 5))

→ not anyone use the first value

, so need to return

function add (num1, num2) {

 const result = `num1 + num2`;

 return result;

}

Get it & = add(2, 5)

for (let i = 0; i < 8; i++) {

 console.log(`Value of i is \${i}`);

,

Return keyword is going usually to ->

return

function Cartoon() {
 return 'string' // return **Cartoon Av.**

,

Get arr = Cartoon();

The datatype of anime will be Cartoon function
return value datatype.

function Cartoon () {

 function CartoonInsideCartoon () {

 return 'Naruto'

}

 return 'CartoonInsideCartoon'

}

Cart anime = Cartoon();
anime()

↳ give error (Cartoon not a function)

→ here we already

call that why Cartoon ~~is~~

function return a string

not a function

Function Creation method

let Cartoon = function () {

 console.log('Anime')

}

Cartoon();

variable विक्षिप्ति declaration @ hoist होता है नहीं

initialization.

var age = 18

Age = undefined

Function @ पुरी - body के बाहर - hoist

होती है,

Const isAllowedVote = function (age) {

return age >= 18;

↳ This is variable which ~~has~~ value in function.

Const isAllowedVote = (age) => {

return age >= 18;

3

or

Const isAllowedVote = (age) => return age >= 18;

or

Const isAllowedVote = (age) => Age >= 18;

Const IsUserAllowedToOpenBankAccount = (age >= minAge) &
(minBalance >= 5000);

age >= 18 & minBalance >= 5000;

Console.log('Is user Allowed To Open Bank Account (23, Good)');

II Data Structure

↳ memory of data of the particular structure

me store [const]

↳ Subjectivity now we can take [const] badiya log

Const fruits = ['apple', 'Cherry', 'orange', 'apple',
'orange', 'kiwi', 'apple', 'Mango']

Console.log(fruits)

fruits.push('kiwi')

Console.log(fruits)

fruits.includes('apple')

fruits.shift()

Const fruit = ~~kiwi, orange~~

~~fruits. shift ()~~ - ~~return index of element from array~~

~~return first element from array~~

~~return a copy of~~

~~element of direction.~~

fruits. unshift ('1', '2', '3');

Concatenated fruits ↗ add '1', '2', '3' at
the first index of array

fruits. indexof (elem);

↙ this return index of element
and if that element not exist
then return $\textcircled{1}$:
↳ -1 means element
not exist

• High Order Function - je kota lga het

→ function mera yarafuction (num1) f
return 2+3+num1;

3
meray na function (5);

→ ~~stata~~ as function as a parameter

→ ~~ta~~ it shaiti ke ~~stata~~ ke

function ta ke ~~stata~~;

→ ~~ta~~ ke ~~stata~~ function as a parameter

function ~~as~~ ~~stata~~ ~~function~~ ~~pt 2~~

function ~~as~~ ~~stata~~ ~~function~~ ~~pt 2~~

3, High Order function bechali;

→ function mera yarafunction (ultra ka function) f

return ultra ka function (1)+40

3
function Cartoon (1) f

return 10;

3
Cartoon (1) = Huaan 10;

Cartoon (1) = Huaan 10;

→ High Order function

for (let i=0; i<fruits.length; i++) {
 console.log(fruits[i])

But JS Obj Copy or Array use Pointer

~~# ForEach~~

④ forEach is function Parameter H

ऐसा ही अॉरिजनल फंक्शन के समान
element के लिए ही बार कोल होगा है

fruits.forEach(element) => console.log(element))

const nums = [1, 2, 3, 4, 5, 6];

const result = [];

for (let i=0; i<nums.length; i++) {

result.push(nums[i]*2);

}

or

const result = nums.map(el => el*2);

console.log(result);

```
function map (fn) {  
    const result = [];  
    for (let i=0; i<nums.length; i++) {  
        const currentElement = nums[i];  
        const num = fn(currentElement);  
        result.push(num);  
    }  
    return result;  
}
```