

Day - 12

① // variables

```
var age = 24;
```

```
age = 35;
```

```
age = 'Pijush'
```

② `console.log('Age is', age);`

```
var age = 24;
```

```
age = 35
```

→ This is like a bad habit
~~top~~ before the
declaration you are
using variable

Not var - deprecated

let

→ var like

→ sometimes

→ use

let age = 6

12 let ~~Roisted~~

Yes

PD2 → Temporal

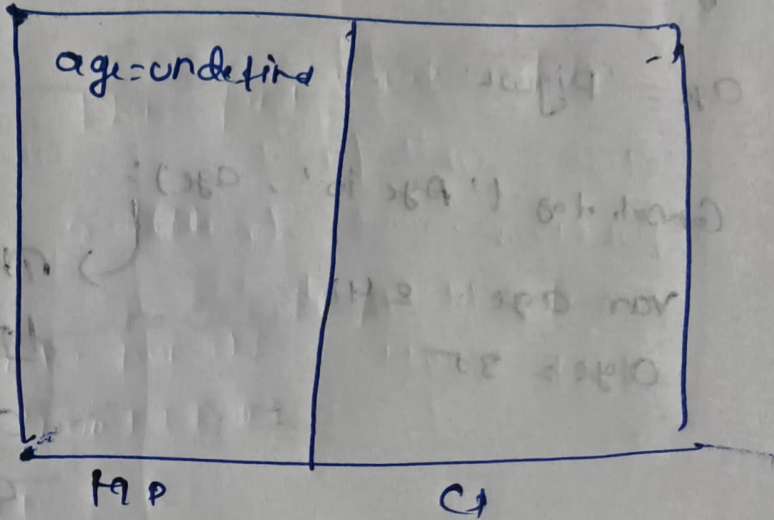
age = 65
console.log(age);

let age = 24;

console.log(age);

age = 45;

console.log(age);



Surrounding : you can not access before initialization

Temporal Dead Zone

↳ it is behavior between let and const variable exist in a scope but cannot be accessed before the declaration

JavaScript

Access Change

Const

- No re-assignment
- इन्फेक्स में const में

day 2 JS

11 What is a function

11 Function - A reusable set of instruction

11 A block of code designed to perform a specific task

function sayHello() {

console.log('Bye')

}

sayHello();

Note

function का नाम से कुछ ले सका है जिनसे
parameter होते हैं

function greet(x, y, z) {

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

}

console.log('Hey', {x: 1, y: 2, z: 3})
↓
String
Interpolation

function add (num1, num2) {

 console.log('Result is ' + num1 + num2);

}

add(2.5);

→ not a problem because the first value

is

so need to refresh

function add (num1, num2) {

 const result = num1 + num2;

 return result;

}

const v = add(2.5)

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

 console.log('value of i is ' + i);

}

Return keyword is used to return the value

Example

function cartoon () {

 return 'string to return karte hai';

}

const anime = cartoon();

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

```
function Cartoon () {  
  function CartoonInsideCartoon () {  
    return 'Naruto';  
  }  
  return 'CartoonInsideCartoon()';  
}
```

```
Get anime = Cartoon();  
anime()
```

↳ give error (anime not a function)

— here we already

Call that why Cartoon ~~and~~

function return a string
not a function

Function Creation method

```
let Cartoon = function () {  
  console.log('Anime');  
}
```

```
Cartoon();
```


variable कि declaration के बाद ही initialization.

var age = 18

Age = undefined

Function की पूरी body के साथ ही const होती है,

```
const isAllowedToVote = function (age) {  
    return age >= 18;  
}
```

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

```
const isAllowedToVote = (age) => {  
    return age >= 18;  
}
```

or

```
const isAllowedToVote = (age) => { return age >= 18; }
```

or

```
const isAllowedToVote = (age) => { age >= 18; }
```

Const isuperAllowedOpenBottleAccount = (age, minBalance)

Age >= 18 && minBalance >= 5000;

Const log (isuperAllowedOpenBottleAccount (23, 6000))

// Data Structure

↳ memory में data को एक particular structure
में store करना

1. Subjectivity ना हो तो list badiya hai

Const fruits = ['apple', 'cherry', 'orange', 'apple', 1,
1, 'banana', ' ', 'lemon']

Const list (fruits[4])

fruits.push('kiwi')

Const list (fruits)

fruits.includes('apple')

2

fruits.splice(1)

Const ~~fruits~~ = fruits

Fruits. shift ()

↳ remove 1st element from Array

Cont firstElement = fruits. splice (2, 5);

↳ return a copy of

element of Definition.

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

Consolidated fruits);

↳ add '1', '2', '3' at

the first index of Array

fruits.indexOf (^{elem.} ~~value~~);

↳ this return index of element

and if the element not exist

this return -1:

↳ -1 means element
not exist

High order function - Jo kisi bhi function ko parameter ke roop mein le sakti hai

```
function mergeFunction(num1) {  
  return 2+3+num1;  
}
```

```
mergeFunction(5);
```

→ Matlab ki function ko a parameter ke roop mein le sakti hai Matlab ke

function ko le sakti hai.

Jo ki jab kisi function ko a parameter

function ~~and return~~ ~~function~~ ~~function~~ ~~function~~

ke, High Order function kehti hai;

```
function mergeFunction(udharFunction) {  
  return udharFunction() + 40;  
}
```

```
function Greet () {  
  return 10;  
}
```

```
Greet() (mergeFunction(Greet));
```

High Order function

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

But JS Obj Map or Array use forEach

forEach

① forEach is function parameter H
 Meant to call the function for each
 element to give each value Call Hoga H

```
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 (e => e * 2);

console.log (result);

function map (fn) {

 let result = [];

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

 let currentElement = nums[i];

 let num = fn(currentElement);

 result.push(num);

 }
 return result;
}