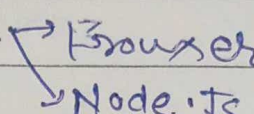


19/Nov/2018

JS.01 →

- 1) Environment :-  Browser
Node.js
 - 2) Javascript Engines :-
 - 3) Variables → Deep Dive → var, let, const
 - 4) Data Types → Number, String, undefined, Boolean, Null
Primitive Data Type
Array, function, Object → reference Data Type
 - 5) Strings :- methods → String manipulation
-

Browsers :-

- Js Introduction Index.html
- intro.js
- index.html.

HTML → Hyper-text Markup Language (make up)
(skeleton)

method

Now How to run JS in Browsers:-

①

<body>

<script src = 'intro.js'>

</script>

</body>

intro.js

↓
console.log('Hello from the browser')

method

②

<body>

<script>

console.log('Hello from the Browser through HTML');

</script>

</body>

JavaScript Engines

Chrome
↓
V8
Engine

Firefox
↓
Spider
monkey

Inter E.
↓
Chakra.

Node.js Royan Dan
(Build in V8 engine)

↓
V8 → local system
↓
Node.js.

Intro.js

console.log('Hello from Node');

↳ Terminal node intro.js

"Variable" declaration with let var and const

var a; // declare first the variable is initialize with undefined.

console.log(a) → %m undefined.

a=20;

console.log(a) → 20.

// JavaScript is a Synchronous Language

↳ means

- Run line by line

- Single threaded language.

Page No. _____

Date _____

JavaScript is a Dynamic Language

↳ means

var a;

a = 20

a = 'Hello'

} Dynamic behaviour.

(Changing any data type)

// SemiColon not Compulsory.

Problems in var keyword

Problem

① First problem with var

- Redeclaration

var b = 2

console.log(b)

b = "I am a string"

console.log(b)

Solution

let b = 2

console.log(b)

let b = 'I am a string'

console.log(b) → (error)

(Identifier 'b' has already been declared)

// let keyword does not allow us to redeclare a variable.

// Loops and if else

let flag = true;

var num = 13

for (let i = 2; i <= num; i++) {

if (num % i == 0) {

flag = false

break;

}

}

if (flag) {

console.log('num is prime', num)

} else {

console.log('num is not prime')

}

Problem 2:- 2nd Problem with var - Scoping

- 1- function scoping
- 2- block scoping

if (10%2 == 0)

{

var c = 2

console.log(c)

}

console.log(c)

Both
works

(It become
more
dangerous)

(var is
function
scope)

if (10%2 == 0) {

let c = 2

console.log(c)

}

console.log(c)

Now
error

(let is
block scope)

Var keyword kit use krni hai kit
only use for Interview preparation.

Const `const c = 2` `c = 'Lillo'` `let a = 2` `a = 'hello'` (Block Scope)
 (Not allowed) String `// Type error.` `// not reassign.`

String

{ `let a = 2`
 `a = 3` // allowed reassignment
 `let a = 2`
 `let a = 3` // not allowed
 } Redeclaring

`let str = 'pepcoders'`
`console.log(str)`

String methods :-

① // .length
`console.log(str.length)`

② Extracting a part of string (from starting to ending -1)
 Ⓐ → slice (start, end) → return sliced array
 Ⓑ → substr

Ⓐ `let sliceStr = str.slice(1, 7)` (start, end)
 `= str.slice(3, 7)` // last index is excluded
`console.log(sliceStr)` → code

Ⓑ Substr method

`substr(start, length)`
`let subString = str.substr(3, 4)` o/p Code
`console.log(subString)`

It expect starting and the length of the word.

③ Replace method

```
let sayHello = 'Hello Mukesh'
```

```
console.log(sayHello) → o/p → Hello Mukesh
```

```
let sayBye = sayHello.replace('Hello', 'Bye')
```

```
console.log(sayBye) → o/p → Bye Mukesh
```

expect the word to be replaced as first argument and the replacement argument as second argument.

④ toUpperCase and toLowerCase

```
let text1 = 'Hello World'
```

```
let text2 = text1.toUpperCase()
```

```
console.log(text2) → HELLO WORLD
```

```
let text3 = text1.toLowerCase()
```

```
console.log(text3) → helloworld.
```

⑤

// Concatenation method

```
let firstStr = 'Hello'
```

```
let secondStr = 'World'
```

```
let ConcatenatedStr = firstStr.concat(secondStr)
```

```
console.log(ConcatenatedStr) → Helloworld
```

Hello_world

⑤ Trim Method

Page No.			
Date			

← (remove space)

let text = ' Hello World '

let trimmedText = text.trim()

console.log(trimmedText) → o/p → Hello World