D-14

14 - 27

classmate
Date
Page 16

1. Introduction
2. using the console
3. what is a variable?
4. Data Types in JS
5. Number in js
6. Operation in js
7. NaN in js
8. Operator precedence
9. Assignment Operators
10. unary operators
11. identifier rules
12. Boolean in js
13. what is Typescript
14. Storing in JS
15. String indices
16. null & undefine in JS

17. Let, Const, Var.

Target

14, 15, 16, 17, 18, 19, 20, 21
22, 23, 24, 25, 26, 27

{ VS-code multiple cursor Press 'Alt' then click where you have to write code in simultaneously. }

1 Using the Console

→ Uses REPL → Read-Evaluate-Point-Loop.

यानि जब console में हम code लिखते हैं, तो पहले console उसे Read करता है, फिर Evaluate यानि calculation फिर Point. और Loop means यह process फिर से दोहराता है।

Console

Eg:
> 1
< 1 — input + enter
> 1+2 — output
< 3 — input + enter
— output

यहाँ पहले 1+2 को read किया, फिर evaluate किया फिर point किया।

Console → clear → Ctrl + L

2 Variable

→ A variable is simply the name of a storage location.

age = 23
name = Tony Stark
↳ variable

Let x = Bigint("123")
>> x
<· 123n ← 'n' means it Bigint है।

[Bigint & Symbol rarely used]

Page 32

3 DATA Type in JS

Primitive Types

1 Number
2 Boolean
3 String
4 undefined → we dont know what in it is?
5 null → we know what is thus but it is empty.
6 Bigint
7 Symbol

Primitive यानि Basic या fundamental Data type.

typeof null = object

↳ object is non-primitive but we take null as primitive data Type.

4 Numbers in JS

→ Positive (14) & Negative (-14)

integers (45, -50)

Floating number with decimal (4.6, -8.9)

JS में कोई Data Type को declear करने का आवश्यता नहीं होता है। JS automatically declear कर लेता है। (like Pythone).

Notes

Data type का v memory limit होता है।

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

eg: a = 0.9999999999999999 (16 digit)

<· 1 - output

यानि 16 digit के बाद में इसे Nearest value count कर देता।

5) Operation in JS

a	a + b	Addition	major operation
b	a - b	Subtraction	
c	a * b	multiplication	
d	a / b	division	
e	a % b	modulo	
f	a ** b => a^b	Power	

Operator

1) Arithmetic + - * / ** %

Power

2) Assignment = , += , -= , *= , /= .

3) Comparison == , != , !== , a > b , a < b
===
a >= b , a <= b.

4) Logical && , || , !

5) Bitwise & , | , ^ , ~ , << , >>

6) Ternary age >= ? "Adult" : "Minor".

7) Typeof Typeof a

पूरा details में जानने के लिए Net में search करें।

6 NaN

→ The NaN global property is a value representing Not-A-Number.

Eg: 0/0
NaN-1
NaN+1
NaN-*1

V.V.I

TYPEOF NaN → Number

7 Operator Precendence

→ This is the general order of solving an expression.

First ( )
**
*, /, % → Same Level of Precedency
+, - → +,- also same level.

Left to Right

if we have multiple power operator then solve power in Right to left

2 ** 2 ** 2 ** 2 R → L

⇒ 2 ** 4 → $2^4 = 16$ Ans.

यदि बहुत बड़ा number आ जाए जो computer में दिखाना posible नहीं तो वह Infinity show होगा।।

2**2**2**4 R → L
  2^4

2**2**16
  2^16

2**65536

2^{65536}

∴ Infinity.

8 Let, Const, Var

→ Java Script एक dynamically Type language है।
ये Automatic variable का data Type को detech करता है।

variables Rules

a) case Sensitive.
b) Only letter, digit, underscore & $ is allowed.
c) only Letter, digit or $ Shoud be First Character.
d) Reserved woord can't be variable name.

Suggestion [use cammel case to declear variable]

fullName → Cammel case ✓
fullname → Small letter all X
full_name → Snake case X
full-name → cebab case X
FullName → Pascal case X

{ आप सारा ही use कर सकते पर cammel case is better }

dont declear variable name :-

```
a = 10;
name = "Nikhil";
cgpa = 7.62;
```

use let, const and var

1. Var : Variable can be re-declear and updated. A Global scope variable.

2. let : can't re-declear but can be updated. A Block scope-variable.

3. const : neighter re-declear nor updated. A Block-scope variable. eg const $\pi = 3.14$;
We can't chage value after declear.

Maximum cases में let और const का use होगा ।

eg: let name = "Nikhil Mahato" ;

```
{
    var name2 = "Nikhil Mahato" ;
}
```

यहाँ कोई error नहीं आएगा ।

But Var का use हम कभी नहीं करेंगे । क्योंकि var का use 2015 से पहले होता है क्योंकि उस Time let और const नहीं होता था ।

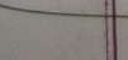
2015 में ES-6 यानि ECMAScript6 (यह JS का new standard) यही new New standard यानि New feature let & const को लाया ।

Why we use let not var?

```
var a = 10;
var a = 20; // Re declear
var a = 30; // Re declear
            // Update
```

```
let b = 100;
b = 200;  // यहाँ let b = 200;
          // नहीं लिख सकते
b = 300;  // Re declear नहीं होगा
          // पर b = 300 update होगा
```

Memory



यानि b नाम का एक ही variable बन रहा memory में ।
इसे update कर सकते हैं ।

Re declear नहीं ।

Nonprimitive
→ object
→ Array

classmate
Date
Page 18

const

→ we can't chang value.

const PI = 3.14;

यदि const लगा दिया तो हमें must value भी देना होगा नहीं तो वह error देगा :

const PI ;

[Syntax error : missing initializer in const declearation.]

But let a;
ये error नहीं देगा ये [undefined] में चला जायेगा।

Brief introduction Objet [Nonprimitive data Type]

Stoore value using [key: value]

const Student {
    name : "Nikhil",
    age : 20,
    cgpa : 7.6,
}

console.log (Student["name"]);
console.log (student.name);

Student["age"] = Student["age"] + 1;
console.log (student.age); [» 21 output]

[इसका है const को change नहीं कर सकते पर यदि वह object में use हो तो उसे हम change कर सकते।]

9 Assignment operators

```
{
let age = 20;          // To assign the value.
age = age + 1          // age += 1;
}
```

```
age -= 1
age *= 1
age /= 2
age %= 2
```
} Short syntax.

10 Unary operator

→ Single operands.

```
age = age + 1   ✓
age += 1        ✓
age++           ✓ unary ++
```

```
age = age - 1
age -= 1
age--   unary --
```

++age → Pre increment
age++ → post increment

11) identifier Rules

→ All JS variables must be identified with unique names (identifiers)

a) Name contains :- letter, number, '_', $.
b) Begin with letter, $, and '_'.
c) Case sensitive
d) Reserved word can't use.

12. Booleam

→ True and False

```
isAdult = true ;
isAdult = false ;
```

Note: in Javascript we can change datatype

Eg: let a = 5;
<. typeof a ⇒ number
a = true; //change type
<. typeof a ⇒ boolean //
;

{Note: CPP, Java में dataType change नहीं कर सकते पर Java script में कर सकते।}

13 Typescript (Design By Microsoft)

→ Typescript is Supperset of Javascript

a) Typescript

b) Static Typed

c) Strict Vision

d) एक बार datatype define होने के बाद we cannot change datatype

a) Javascript

b) Dynamic Typed

c) Normal

d) we can change data type any time

14) Storing in JS

→ Strings are text or sequence of characters.

We can use double qout or single qout

```
let name  = "Nikhil";
let title = 'Mahato';
let char  = 'a';
let num   = '23';
let empty = " ";
```

} All are storing

To print sentence with quot

» let poem = 'this is "apple" ' ✓ valid
let poem1 = " this is 'apple' " ✓ valid
let poem2 = " this is "apple" " ✗ invalid
let poem3 = 'this is 'apple' " ✗ invalid

15 String indices

→ let name = T O N Y _ S T A R K
0 1 2 3 4 5 6 7 8 9

name[0] → T
name[1] → O

16 null & undefine in JS

undefined

→ A variable that has not been assigned a value is of type undefined.

> let a ;
< undefined

null

→ The null value represents the intentional absence of any object value.

To be explicitly assigned.

eg :> let a = null ;
< undefined
> a
< null

Some Important if found