Day17(string methods, min/max_int,===&== ,boolean)

JS

Numbers → man tve no?
                    -ve no?

✓ man   999 9999 -- 1000000

man   99 999 --   infinity

                    Store X

                    ✓ man limit

                    ✓ min limit
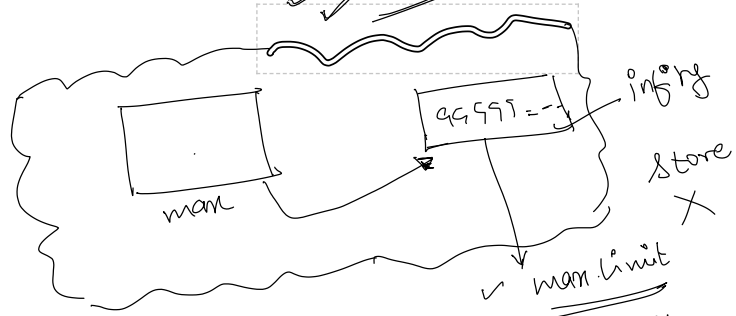
9876
-9875

man →   Number . MAX_SAFE_INTEGER ✓

min →   Number . MIN_SAFE_INTEGER ✓

✓ html → index. html → JS own ?

                    YES

    < Script > ✓   Str[5]

                    1 2   56
                    ↓↓   ↓↓
Let Str = " Hi_i_am_your_friend " ;
                    ↑ ↑↑
                    0 3 4

                    ⓐ

① \1 file pile up X

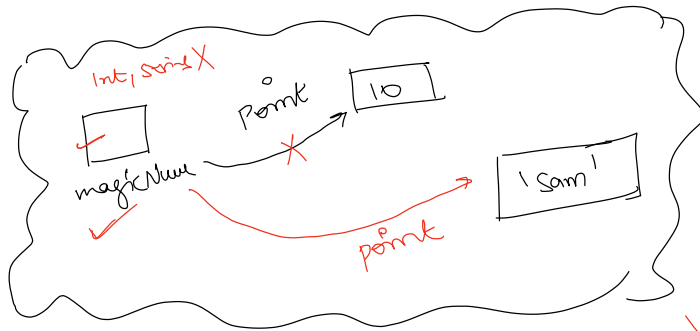   tion or concern

② Separ......   code

efficient

- HTML →
- CSS →
- JS →
- Boot →

Diff. files

easily readable
easily accessible

bug (Kis Jagah)

Same variable → (Let)
cannot be reinitialised.

① Let magic

② magic Num
(Valid)

Memory

Int, String ✗

Point ✗ → [10]

magicNum → point → ['Sam']

variable type ✗ (Zaroorat)   (JS)

variable

datatype ✗

weakly typed PL

CPP

variable
datatype ✓

int buck1 = 10;

buck1 = 'Sam';

Weakly typed language:
It is a programming language
in which variables are not
bound to a specific data type.

Strongly typed language:
It is a programming language
where variables are bound to
a specific data type and
cannot be changed at runtime.

Let variable_name = (10)

variable_name = 'Sam'

(allowed)

(Number) ✗

→ Let var_name = 10;

✗ Let var_name = 100; ✗ error

$\boxed{var\_name} = \underline{100};$ ✓

✓ $var\_name = 'String';$ ✓

$= True;$ ✓

$\underline{any\ datatype}$ (change)

$\underline{PL}$ 1 declare
              kuch bhi
              stored

① Let naam = 10 ; → Number ⎫ weakly
② naam = true; → Boolean
③ console.log(naam); → true //ans
④ naam = 'Samanth'; → String
⑤ console.log(naam); → Samanth //ans

——→ X end X ——

naam   kis type ka tha
 ↳
   (run) X

   (run)   variable type
   ↓
   time

Statically typed languages:
A language is statically typed
if the type of a variable is
known at compile time.

Dynamically typed languages:
A language is dynamically
typed if the type of a variable
is known at run time.

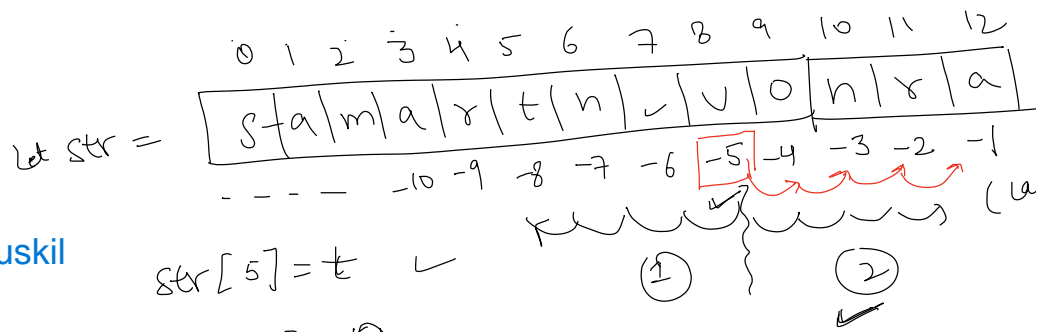$\underline{Dynamically\ typed\ PL}$

$\underline{Cpp}$ (strongly typed & statically typed)

X   (int) naam = 10;
       (naam) = 100;
       ↓   = 1500 /2000;
      DT

?

JS → interpreted → weakly → Dynamically → [____]
typed typed

```
     0  1  2  3  4  5  6  7  8  9  10 11 12
Let Str = | s | t | a | m | a | r | t | n |  _ | v | o | n | r | a |
          ____ -10 -9 -8 -7 -6 -5 -4 -3 -2 -1          ( la
```
① ②

Str [5] = t
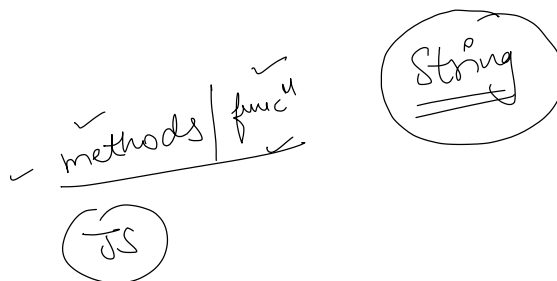Str [7] = O
Str [17] =   X

Javascript m error lana muskil hai ?

1. Agar hum string m aise index ko access karenge jo ki string ki length ki range se bahar hai toh hame error nahi milega balki undefined milega.
2. Hum explicitly out of index ki value ko value provide nahi kar sakte.

2 => explicitly (Str [17] → valueX (assign)
     (khudse)

Str [17] = undefined   JS automatically **put**
                                   undefined
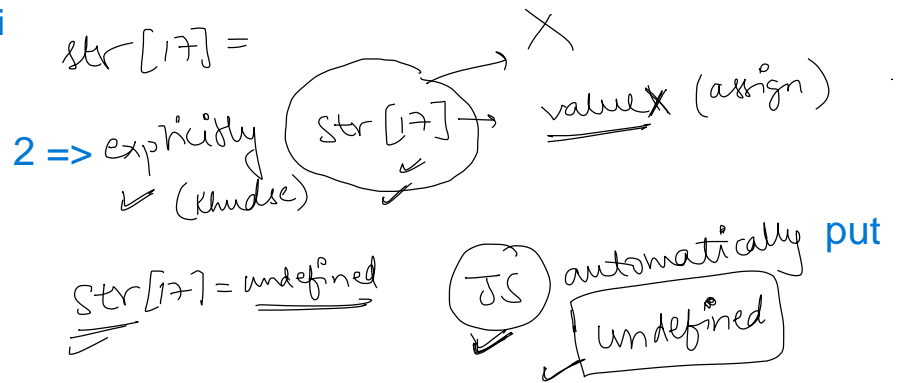
1 => Str [17] → undefined → undefined

methods | funcⁿ   String

JS

String methods

① substr 🗑
② substring (replacement)   ⑦ toUpper(
③ indexOf ✓   ⑧ toLower(
④ replace   ⑨ trim(
⑤ replaceAll

⑥ repeat

① substr →
0 1 2
'Samanth Vohra'

str . substr (2)
(start pointing) → sp

str . substr (2, 5) ;
(sp) (length)
count

marth

both negative index will always
return "" (empty string).

str . substr (-5, 5)   return => 'Vohra'
sp   length

str . substr (-5, -5) => return " "

② substring

0 1 2 3 4 5 6 7 8 9 1
S a m a i r t h V b

tra
ending point

substring (2, 5) length X
sp   ep
sp

mart

ep → is not included
X

substring (5, 2)
sp ep

wherein   sp > ep   swapping of values

→ wri-   ' $(5, 2) \rightarrow (2, 5)$

when start > end then swapping
takes place b/w start and end
indices.
Eg : str.substring(5, 2) => (2, 5)

substring $\begin{array}{c}(-5, 2)\\ \uparrow_{sp} \quad \uparrow_{ep}\end{array}$

→ (−ve) values are encountered they are
alloted to 0 .

$(-5, 2) \rightarrow (0, 2)$ $\begin{cases} -5 \rightarrow 0 \\ -100 \rightarrow 0 \\ -500 \rightarrow 0 \end{cases}$

→ substring $(-5, -2);$ ✗

$(-5, -2) \rightarrow (0, 0)$

③ indexOf   Samauth _Vohra _ Wels

dlg $\left(str . indexOf('a'); \right)$ → number

str = | Samauth Vohra is a good boy |

('a', 2)
  ↑ ✗

3rd a   (2nd & 3rd) a
2nd a
5th a

('a'

-- and boy"

Str = "Samarth is a goo... ..."
      0 1 2 3 4 5 6 ---- - - -

str.indexOf('a') → 1st encounter ✓ → ①

str.indexOf('a')+1 → ②

2nd encounter → str.indexOf('a', str.indexOf('a')+1) ;
                                              2
                        ⇓
                        ③ + 1 = 4

3rd encou

str.indexOf('a', str.indexOf('a', str.indexOf('a')+1)+1);

Boolean

== (general equality) →

=== (strict equality) →