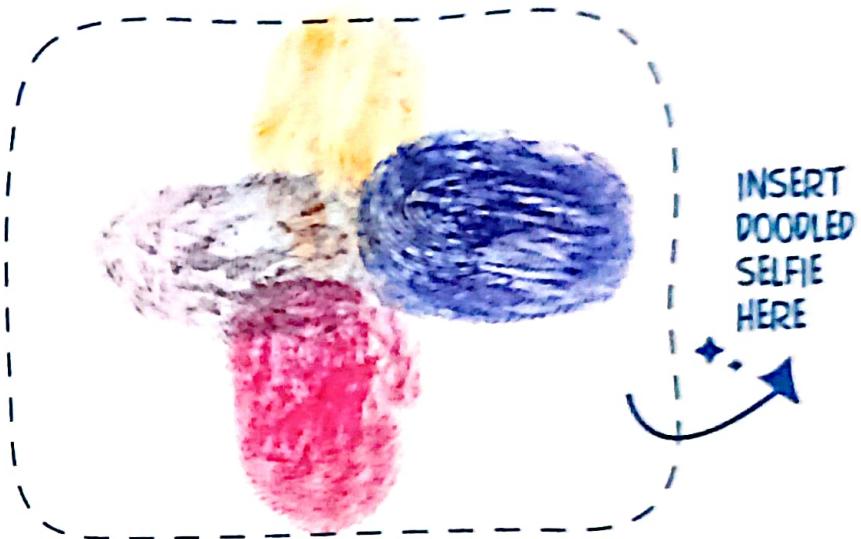


9:AM 2:30PM
32 HUQ

10:30
11:30



NAME SahonP Ashish Anit

ADDRESS

MOBILE 9136282407, [Call me]

EMAIL ashishshahani1-6-@gmail.com.

EMERGENCY CONTACT #SAY CODING...
SAY Web Development
Development

#

①

㊀

②

③

7/5/22

9/4/22

Console Type

int, value, Anything, Any

Console.log (_);

Console.log ('Hello world');

Console.log ([34, 2, 1, 2]); \rightarrow This is an array

Console.log ({harry: 'this', marks: 34});

\hookrightarrow In Harry Brackets This is a [Object].

Console.table ({NAME: Ashish, marks: 50});

\hookrightarrow print in table form

Console.time ('^{Writ a} identifier');

\hookrightarrow Do How many time require to write code.

Console.timeEnd ('my code')

\hookrightarrow Always End by Console.timeEnd & Start by Console.time in this between your code.

Condition? : boolean, message?

(4) console.assert ('556 > 189, 'Age is not possible');

(5) console.error ('Warning');

(6) console.clear();

#

In Javascript THREE TYPE OF variables
let, const, var.

① // Rules for creating JavaScript variables. (var)

- ① Cannot start with numbers. underscore.
- ② Can start with letter, -, \$
- ③ Case sensitive.

② // The value of a const (constant) can't be changed through reassignment.

① ~~for~~ var ma value assign kar sakte ha
aur value ko change kar ke second
value de sakte sakte hai

② But, In const, ek bar value daal diya
to, humara value ko change nahi
hoga kar sakte hai.

Eg :-

const marks = 60;

Change \Rightarrow Marks = 50; // This will give an error

Change \Rightarrow Marks = Marks + 40; // == =

③ // Let variables is defined as

Block Scope; local, not global

Eg:

```
let x = 1;  
if (x === 1) {  
    let x = 2;  
    console.log(x);  
    // expected output: 2  
}
```

```
console.log(x); // expected out: 1
```

// Case types //

① Camel Case \Rightarrow Second word first letter capital.

② Kebab Case \Rightarrow give a hyphen

③ Snake _ Case \Rightarrow Underscore

④ Pascal Case \Rightarrow First letter are capital.

#

Data types in JS

8/5/22

Base
data
type(Primitive data)
type(Reference data)
Object,
directive.

- String → "Ashish"
- numbers → 35
- Boolean → true, false
- Null →
- undefined → undefined value
- Symbol → ES6

- Arrays
- Object literals
- Function
- Date

* In Ma Saikisi Bhi value ka type
janana ha to. (typeof)

Eg:-

```
let Name = ('Ashish');
console.log ("Data type is "+(typeof Name));
```

| Output | // String Data type is String

* Non= not a number

10/5/22

* Type Conversion and type Coercion.

⇒ The process of converting data of one type to another.

eg ① Converting String to Number, ~~"34"~~

② Converting Number to strings, "34"

③ Converting Dates to numbers, ~~Number()~~ ~~String()~~

④ Converting Number to Dates., ~~getTime()~~

⑤ Converting Booleans to Number., ~~Number()~~

⑥ Converting Number to Boolean.

// type of conversion and type coercion

* Type of Conversion

① Koi Bhi number ko String Convert.

↳ `String();`

Eg:-

`var = String(34);`

`console.log(var, (typeof var));`

② Koi Bhi Boolean ko String Convert

↳ `String(true);`

③ Koi Bhi Dates ko String Convert

↳ `String(new Date());`

④ Koi Bhi arrays ko String Convert.

↳ `String([1, 2, 3, 4, 5, 6]);`

↳ You can find the length.
`length.`

Imp

{ Arrays ki length ka matlab, How many elements
String ki length ka matlab, How many characters.

* Definition

↳ toString() returns a number as a String.

Eg:- let tostri = 75;
console.log(tostri.toString());

Number Function

(5) Convert into string to number.

↳ Number();

Eg:- let Stri = Number("345-4");
console.log(Stri, (typeof Stri));

Also you can't convert "String" like "letter" and errors, ~~you~~

* Use of parseInt / parseFloat Function

① parseInt :- The parseInt method takes a value as a String and returns the first integer? Not returns decimal point.

Eg:- $34.56 \Rightarrow 34$.
 $60.50 \Rightarrow 60$

② parseFloat :- Converting into floating point

Eg. $34.60 \Rightarrow 34.60$

* toFixed() function.

This value is decide how many decimal you want.

toFixed() :- to fixed is by default 0 decimal you can put the value & add how many decimal point you want.

Eg:-

```
let number = parseFloat('34.09');  
console.log(number.toFixed(), (typeof number));
```

{
 ↗ by default, is zero.
 app jitana value put kar
 ga utana decimal increment
 logo.

* Type of Coercion / Concatenation.

- =) Type coercion is the automatic or implicit conversion of values from data type to another. (such as string to number).

Eg:-

```
(const value1 = '5');
```

```
(const value2 = 9);
```

```
let sum = value1 + value2;
```

```
console.log(sum);
```

Output // 59,

10/15/22

* Strings; properties, Methods & Template Literals in Js. *

Note :- JavaScript counts position from zero. 0 is the first position in a String, 1 is the second, & so no.

- ① * • `charAt(x)` :- It returns the character at the "x" position within the string.

↳ `charAt(1)`, write value

- ② • `concat(v1, v2...)` :-
⇒ Combines one or more string and returns the concatenated string. Remember that the original string is not modified.

Eg :- // `concat(v1, v2, ...)`

var message = "ASHISH";
var final = message.concat(" is
a good boy", " learn a JS.");

Output:- alert(final).

ASHISH is a good boy learn a JS.

9.

=)

(3)

• CharCodeAt(x) :-

=) This function will return the Unicode value of the character at position 'x' within the string.

Eg:- //charAt (position)

var message = "Jondholo";

Eg:-

console.log(message.charCodeAt(0));

[Output [1]]

5

Eg:-

From char code (c₁, c₂)
Function will

.indexof (substr, [start_from]) :-

This method searches and returns the index number of the searched character within the string. If not found, it will return -1. "start_from" is an optional argument specifying the position within String to begin the search. Default is string length -1.

Var Str1 = "Hi, my name is Ashish";

Var Str2 = Str1.indexof ("locate");

// Output : -1

.lastindexof (substr, [start_from]) :-

Similar to indexof difference is count the last indexof string.

Var Str1 = "Javascript 20x";

Var Str2 = Str1.lastindexof ('r');

// Output : 11

⑤ .endswith()

=> This function check whether a string ends with specified string or characters. return Boolean value.

Ex:-

//endswith()

var mystr = "List of javascript function";

var n = mystr.endsWith("Javascript");

//output : True

⑥ .includes()

=> It is use to check whether a string contains the specified or character, return Boolean value

Ex:-

//include()

var str1 = "Hello, welcome to my page";

var str2 = str1.includes("page");

//output: True

also, It is return space '' True.

→ $\text{substring}(0, -4) \Rightarrow$ this value valid but lost four
value is not return. (-4)

(7)

→

`substring (from, [to]) :-`

It returns the character in a string between "from" and "to" index, NOT including "to" itself. "To" is optional, and if omitted, up to the end of the string is assumed.

Eg:-

```
// Substring (from, to)  
var mystri = "Javascript is a good";  
var mystr2 = mystri.substring(0, 7);  
// Output : Javascript is a go  
                                ↴ Count a space also.
```

(8)

→

`.slice (start, [end]) :-`

This function return a substring of the string. The slice function is similar to substring but substring -value is not return, .slice include a -value.

Eg:-

```
// Slice (start, end)  
var text = "excellent";  
text.slice(0, 4)    // exc  output  
text.slice(-4)     // lent  output.
```

⑨

split (delimiter, [limit])

- This will split a string into many according to the specified delimiter, and returns an array containing each element. The optional "limit" is an integer that lets you specify the maximum number of elements to return.

Eg:-

```
var txt = "a,b,c,d,e";
```

```
txt.split(","); // Split on commas
```

```
txt.split(" "); // Split on spaces.
```

(ii)

Created in errors forms.

⑩

replace (regexp / substr, replacetext)

- Searches and replaces the regular expression (or sub string) portion (match) with the replaced text instead.

It will replace only first occurrence

11

Eg:- // replace()

```
var mystring = "999 javascript codes";
console.log(mystring.replace("999", "45"));
// Output :- 45 javascript codes.
```

(11)

toLowerCase() :-

This will return the string with all of its characters converted to lowercase.

Eg:-

// toLowerCase()

```
var myString = "JAVASCRIPT Box";
myString = myString.toLowerCase();
console.log(myString)
// Outp : javascript box
```

(12)

toUpperCase () :-

similar to toLowerCase, convert all character to uppercase.