

Javascript starting year
is 1995

es fall from is ECMAScript)

ES1 : 1997

ES2 :

ES3 :

ES4 :

ES5 : 2009 (with New features)

→ ES6 : 2015 (biggest update in history)

↓
New

Name is

ES-2015

↓
(ModernJS)

(members of TC39)

→ ES7 : ES 2016

→ ES8 : ES 2017

* Javascript is backward compatible.

↳ यद्यपि ये नए फीचर्स ES1 से लेकर तो
सभी ES8 तक तभी तभी सभी सभी सभी.

* Javascript is not forward compatible

↳ यद्यपि ये नए फीचर्स ब्राउज़र वा एंड्रॉयड ब्राउज़र
update करने की जरूरत नहीं है तो भी अभी ये नए
फीचर्स काम नहीं करते.

Comment on javascript :-

Note:-

* single-line comment :

// this is comment

* multi-line comment

/* this
is multi line
comment */

// Java script Hello word print //



console.log("Hello word");

// Java script has three type of Variables //

1. Var

2. let

3. const

} New Variables

1. Var : Old Variable &?

→ Var is global level variable &!

Var is old variable & is global level variable &
declaration is old variable & is global level variable &
its value is update & old variable &

Ex:-

{

first → Var x = 20;

second → console.log(x); ← ①

x = 21;

.log(x); ← ②

Var x = 30;

second time → console.log(x); ← ③

Output

① - 20

② - 21

③ - 30



તો અને var ની એકાં બધાં આ declar ને update
કરી શકીએ છું. તે (બધાં secure નથી. તો
તો નો use નો ખાલ છે!)

૨ let :-

Js ની let નો use સભ્યાની શાખામાં ESG (2015)
ની પેટી હતી.

→ let એ Block level Variable એ.

→ કિમી let એ આ declare કરીયા એ તો
Value કર્યા ~~update~~ કર્યું હશે હતી. તો

{ જેણું let એ Js ને Secure બનાવું છ'

Ex:-

{

first time → let x = 20;

console.log(x); - ①

x = 25;

console.log(x); - ②

3

Output

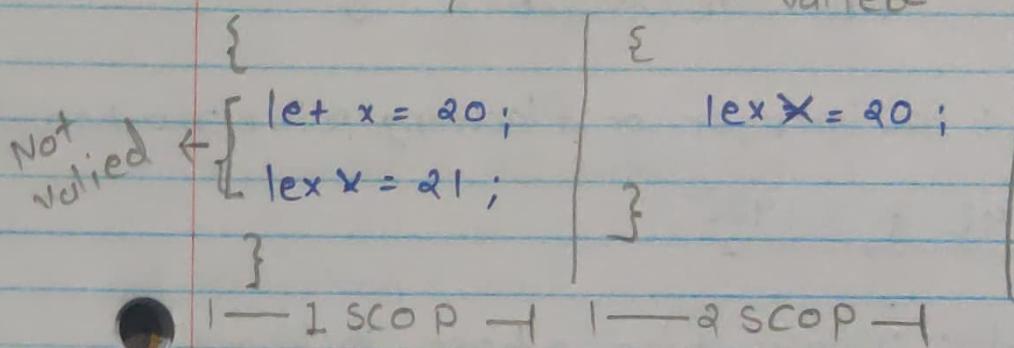
① - 20

② - 25

let ને જોકિ scop ની અંતર્ગત સમાવેશ થાયા હોય ત્થી declare કરી શકતો નથી તો તો Error આપકા

{ એવી scop ની યારા કે એવી new scop ની }
 { અંતર્ગત કુલી declare કરી શકતો કે? }

Ex:- *new scop* valued



3. Const

Const નાના Block level variable એ. નિ એવી નિયમ હોય કે declare કરી શકતું એ. const ને let ની રીતે update એવી કોઈ માટો નથી.

Ex:-

{
 Const x = "guyan";
 console.log(x);
 }

Output

Jayam

Const Variable

ની Declare સારી

માટે તીવ્યા value

ન્યૂની દેખાયા રહ્યું એ

Ex:-

const x = 4;

x = 20;

not work

Output

Error

not work
update x = 21;

console.log(x); }

const x = 20;

work

Data types

* Primitives and objects

7 primitives Data types.

1. Null :- A special keyword denoting a null value.

{ Null and null both are not same
because JS is case-sensitive.
null is equals 0 }

2. number : An integer or floating point number

Ex: 42 & 3.1435

3. Symbol : a data type whose instances are unique and immutable.

4. String :- A sequence of characters that represent a text value.

Ex:-
"Hello"

5. Boolean. is return only

true or false value
(1) (0)

6. BigInt :- A integer with arbitrary precision

Ex:-
→ 984682768455n...

→ 3.141592653589793 (value of pi)

Type of Variables :-

```
var myInteger = 12; // 32 bit number  
                    (from : -2,147,483, to 2,147,483);  
  
" mylong = 9310141419489; // 64 bit number  
                    (from : -9,223,036,854,775,808  
                     to 9,223,272,036,854,775);  
  
" myFloat = 5.5; // 32bit floating-Point number  
                  (decimal)  
  
" myDouble = 9310141419489.99; // 64 bit floating-point  
                                 number  
  
" myBoolean = true;  
" myBoolean2 = false;           } " 1 bit // 0 bit  
  
" my NotANumber = NaN;  
  
" My ·NaN-Example = 0 / 0;  
                    // NaN : Division by zero is not possible  
  
" notDefined; // undefined we didn't define  
               it to anything yet  
Ex:  
x = 20;  
window.alert(x); // undefined  
  
" myNull = null; // null
```

Create a tag in HTML using Javascript

js એવીએ હતી તગ બનાડું માર્ગ
createElement મધ્ય ની ઓપરેટર કરવામાં આપ એ

Ex:- (variable name)

Var h1tag = document.createElement("h1");

~~h1tag.text~~ h1tag.textContent = "Hello Guy";

document.body.appendChild(h1tag); tag ની વિશે એ

CreateElement એ

HTML એ body ની

અને એ tag બનાડું શકે એ.

section સ div ની

લિફ્ટ કરું એ.

ગુણીય create

ફારીની નીચે ની tag એ એવી

Javascript Reserved Keyword — 11

var

function

if

else

do

while

for

switch

break

continue

return

try

catch

finally

debugger

case

class

this

default

false

true

in

instanceof

typeof

new

null

throw

void

width

delete

1. What is the result?

$$\rightarrow 1. \frac{1 + 2 \times 3}{1 + 6}$$

$$= \boxed{7}$$

$$\rightarrow 2. \frac{1 + "2" \times 3}{1 + 6}$$

$$= \boxed{7}$$

$$\rightarrow 3. "1" + \frac{"3" \times 3}{1 + 6}$$

$$= \boxed{16}$$

2. What is the result?

$$\rightarrow 1. 1 + 2^{**} \frac{3}{1} / 2 - 1$$

$$1 + 2^{**} \frac{3}{2} / 2 - 1$$

$$1 + \cancel{2}^{**} \frac{3}{2} / 2 - 1 = 5 \cancel{+} 1$$

$$= \boxed{2} / \boxed{1} = \boxed{4}$$

$$\rightarrow 2) \cdot 9 / 2^3$$

$$= 4.5 * 3$$

$$= \boxed{13.5}$$

$$\begin{array}{r} 4.5 \\ \times 2 \\ \hline 9.0 \end{array}$$

$$\begin{array}{r} 4.5 \\ \times 3 \\ \hline 13.5 \end{array}$$

$$\rightarrow 3) \cdot 9 / (9^3)$$

$$= 9/6$$

$$= \boxed{1.5}$$

$$\begin{array}{r} 15 \\ \times 6 \\ \hline 30 \\ 30 \\ \hline 0 \end{array}$$

3 what is the result

$\rightarrow 1) x = 1; \frac{x++}{2}; \text{alert}(x);$
 $x \leftarrow \text{result.}$

$\rightarrow 2) x = 1; \frac{\text{alert}(++x)}{2};$
 $x \leftarrow \text{result}$

$\rightarrow 3) \frac{x = 1}{T}; \text{alert}(x++);$
 $x \leftarrow \text{result}$

4. What is the result?

→ $1 == 01$: false

→ $1 === 01$: false

→ " 1 " == "01" : false

→ " 1 " == "0" + "1" : true

→ " 1 " == "0" + " 1 " : false

→ " 1 " == "0" * "1" : false

→ " 1 " == "0" * " 1 " : false

5. What is the result?

→ true && false ; false

→ true && false || true : true

→ true && false || !true : false
false false

→ !(true && false) : true

6. What will be the output of the js code

A

```
var string1 = "123";
```

```
var intvalue = 123;
```

```
alert(string1 + intvalue);
```

→ 123123

Error ?

f. Guess the ternary operator used in the following example.

marks = 33

```
marks = (marks < 35) ? "full" : "pass";
```

→ if else

g. What will be the output of the following script

```
<script>
```

```
var x = 5;
```

```
document.write(x == "5");
```

```
</script> - False
```

9. what will be the output of the following JS code?

→ Function height()
{

Var height = 123.56;



Var type = height >= 190 ? "tall" : "short"

return type;

y → short

10 what type of value gets printed if we add following two variables

Var a = "1"; → string

Var b = 5; → number

20/01/2024

// String - with - quotes //

let x1 = "He ' llo " ;

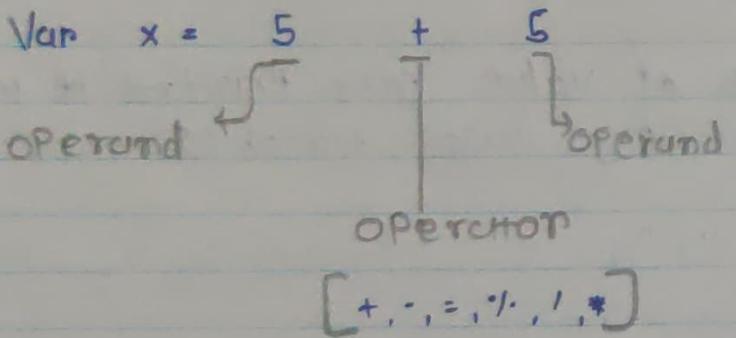
let x2 = "He ' llo " ;

let x3 = ' He " llo ' ;

let x4 = ` He " llo " ;

let x5 = ` He It In llo ' ;

// EXPRESSIONS and OPERATORS //



6 TYPE OF OPERATORS

- 1 Assignment Operators :
- 2 Arithmetic Operators :
- 3 Comparison Operators :
- 4 Logical Operators :
- 5 String Operators :
- 6 Conditional (ternary) Operators :

OPERATORS

1. Assignment operators

1. + (Addition)
2. - (Subtraction)
3. * (Multiplication)
4. / (Division)
5. % (Modulus)
6. ++ (Increment)
7. -- (Decrement)
8. ** (Exponentiation) (New)

→ 8 Ex:-

(**) ഒരു OPERATOR എന്നീ ഗവി യാകു ഒരുപയാക്കിയ ഒരു തിരി തിരി ചാർഗ്ഗ യാകു ഒരു മന്ത്രം ഫോം അഥവാ ലൈറ്റ് ഫോം ആണ് ഇതും ഒരു ഫോം ആണ്.

Ex:-

(മന്ത്രം നിരൂപിക്കുന്നത്
(work 8))

let x = 2**3; } 8
Console.log(x);

$$\begin{array}{c|c} 2^{**}3 & 2^{**}4 \\ \boxed{2 \times 2 \times 2} & \boxed{2 \times 2 \times 2 \times 2} \end{array}$$

* and Assignment

1. +=
2. -=
3. *=
4. /=
5. %=
6. **=

Ex:-

$$1. \quad A = A + B \rightarrow .$$

1. $+=$ (short)

$A += B; \rightarrow A = A + B$

short

long

2. $-=$

$A -= B; \rightarrow A = A - B;$

3. $*=$

$A *= B \rightarrow A = A * B;$

4. $/=$

$C /= B \rightarrow C = C / B;$

5. $\% =$

$A \% = B \rightarrow A = A \% = B;$

6. $**=$

$A ** = B \rightarrow A = A ** B;$

Q. Comparison Operators:

→ $==$ (Equal)

check if the value of two operands are equal or not .if yes then the condition becomes true.

→ $!=$ (Not Equal)

check if the value of two operand are equal or not .If the value are not equal ,then the condition becomes true.

- $==$ (Identical equal and same type)
The strict equality ($==$) operator check whether its two operands are equal. returning a Boolean result unlike the equality operator, the strict equality operator always considers operands of different types to be different.
- $!=$ (not equal value or not equal type)
check if the value of two operands are equal or not if the values are not equal, then the condition becomes ~~true~~ true.
 - $(0 != 1) \rightarrow \text{true}$
 - $(0 != 0) \rightarrow \text{false}$
- $>$ (Greater than)
- $<$ (less than)
- \geq (greater than or equal to)
- \leq (less than or equal to)

3. Logical operators

→ `&&` (Logical AND)

`(0 && 0) = false`

`(0 && 1) = false`

`(1 && 1) = true`

→ `||` (Logical OR)

`(0 || 0) = false`

`(1 || 1) = true`

`(0 || 1) = true`

→ `!` (Logical NOT)

`!(0 && 0) => true`

// document.write();

→ document.write(); ari function of API of JS on aier aapno HTML file hi JS g output hui rahte hain. aur console.log(); function on aise work ki ed.

Ex:- function

→ document.write();

→ document.open();

→ document.close();

→ document.writeln();

// .write() & .writeln()

* write() does not add new line (CR) after statement.

Ex:-

{

document.write("Hello");
document.write("word");

}

OUTPUT

⇒

Hello word!

writeln does add new line CR after statement.

Ex:-

{

document.writeln("Hello");
document.writeln("word");

}

Hello
word:

* Prompt();

→ Prompt function ની મદ્દ કરી આપણી JS ને અંદે યુઝર પાટી કી value ને Enter કરવી શકાડું છોડ્યું.

-: Syntax :-

```
let x = prompt();
```

આમ જ્યારે web page ~~is~~ Reload હતું રહ્યું તોના પર એક
POP UP શોડ હતું કેમા આપણી x ની value entre
કી શકાડું છે.

* alert();

Alert method નું કથું એવા prompt ની કેમણ
work કરી શકે.

Note:-

Alert ની user આપણી input ને વિસ્તાર નાલ. //
નિયમ POP કરી message કી શકે છે

{ syntax: { alert("Hello"); } }

↑ print message

* Confirm();

Confirm method દ્વારા user પાટી કી Allow અને cancel
ની permission આપણા માટે POP બોલાવી અને
HTML Page ની એવું output કરું શકાડું રહ્યું
શકે.

→ કોઈ true અને false કા કા કા work કરી શકે

Syntax

```
• confirm(" ");
```

↑
print message || Pass Value

* if - else.js

-: Syntax :-

```
if (expression)
{
```

} Statement(s) to be executed if expression is true

```
else {
```

} Statement to be execute else expression is false

* if()...else if()...else.js

-: Syntax :-

```
if (main expression)
{
```

} Statement to be executed if expression main is true

```
else if (expression 1)
{
```

} Statement to be executed if expression 1 is true

```
else if (expression 2)
{
```

} Statement to be executed if expression 2 is true

```
else {
```

} Statement to be executed if none of expression are true.

* Switch case :-

-:- Syntax :-

```
let input = "1";
switch (input) {
```

case 1:

```
    doc.write("this is case 1");
    break;
```

case 2:

```
    doc.write("this is case 2");
    break;
```

case 4:

```
    doc.write("this is case 4");
    break;
```

default:

```
    doc.write("this is default or case");
    break;
```

21/01
Date
Hollyday

23/01/2024

-:- Loop's :- in JS :-

-:- Syntax :-

run type

first → second → third

1. while [condition executed if condition wrong than loop infinite]
→ check
while (condition)
{
 code to be executed / increment / decrement
 }
Part.

Ex:- let x = 0;

OUT-PUT

while (x <= 5)

0 1 2 3 4 5

{

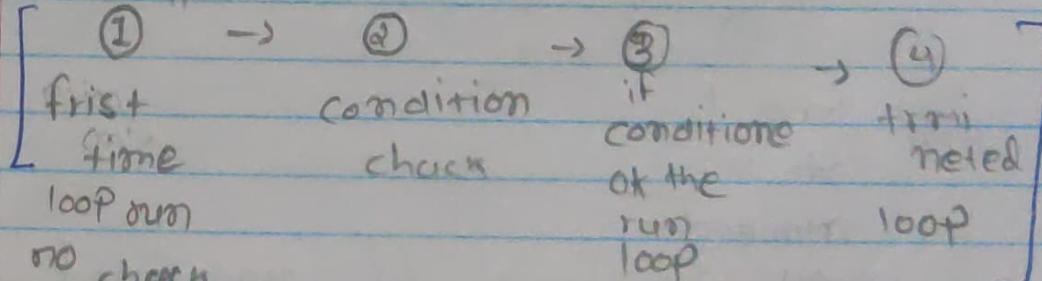
doc.write (x);

x++;

}

2. do while

run type



Condition

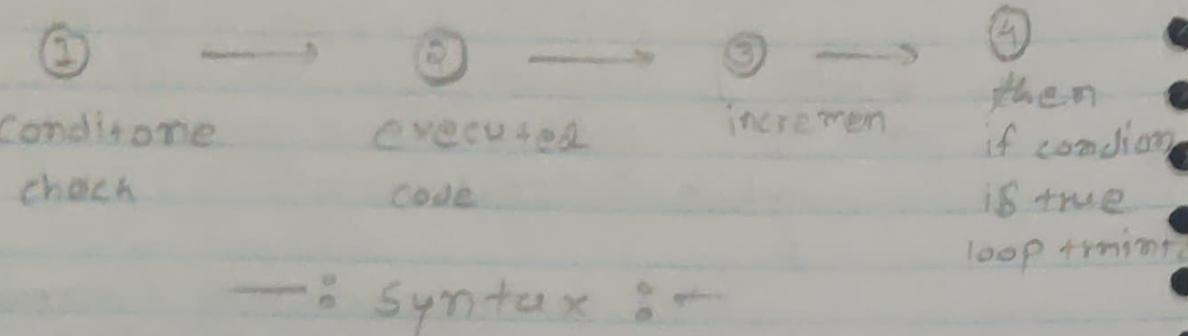
Syntax :-

do {

 code to be executed / or increment / decrease
} while (condition);

3. for

run type



—: Syntax :-

for (initialization ; Condition ; Update Express)

 {
 Code to be Executed
 }

// Undefined //

→ undefined console all ran show करते हुए वाले जैसे ऐसी variable create करते हैं तो उनका value assign करते हुए तभी output करते हैं तभी undefined show करते हैं।

Ex:-

```
let nume;  
Console.log(nume); // undefined.
```

Not { map like set
method use in
function. }

25/01/99
— v —

11 - New Map (); - 10

Basic Syntax of map

[start map]

{ Const Map1 = new Map ([])
Variable Income declaration method

[" option-1 ", 20], ←
[" option-2 ", 30]

↑ ↑
Key name Value

for coll value

]); ← [End map]

* Some method it's use in map

1. set();
2. get();
3. delete();
4. has();
5. entries();
6. size

key()
values()
forEach()

1 Set();

syntax of use

const map1 = new Map ([]); // end map

map1.set("option-3", 40);

↑ ↑
Key Value.

→ Set ની use map મુજા પથ્ય એણી કરવા હો આપણે તું કરી શકતું હો છો કે map ની ટોપ વિધાને Key and Value add કરવા માટે use ચાપ એ!

{ Set ની key value ને update કરાવી શકતું હો ! }

2. get()

Syntax of use

key name

→ `Console.log (map1.get("option-1")); // 30`

આદ્ય map ની હોય કોઈ specific value show કરવા માટે map ની name પણ get બાબો અને જીબી હોય key name નિયમાની આવ છે. તો તું તું કોઈ Key Value Show કરવા માટે!

→ `Console.log (map2);`

આદ્ય `Console.log` ની હોય અને map name નિયમાની હોય full map ની key and value બાબો show કરી એ!

3. delete()

Syntax of use

→ `Console.log (map1.delete("option-2")); // 30`

આદ્ય map1 ની હોય option-2 name દરાવાની key Remove કરી. `.delete` method ની દીને હો key name નિયમાની આવ ગે delete કરી અથડા value કાઢી.

4. has()

Syntax of use

```
Console.log ( map1.has("option-1")); // 30
```

has ની અને એક key name નવામિ આપ એ ની
map ની અને એ ની ની has એ boolean
Value એટાં કે true અથવા false Return
શકે છે!

⇒ એ key map ની ફિય ની = true Return
શકે.

⇒ એ key map ની ન ફિય ની = false Return.

5. entries()

Syntax of use

```
Console.log ( map1.entries );
```

→ entries method જીએ map ની કોઈ entries એ
ની console ની બેચે કાણાય તો તે માત્રા console ની
web ની અને show શકે છે!

7. Foreach()

foreach method જીએ

map ની simpl txt ની print શકાય શકે

ની HTML file ન
એસ્ટ્રીન્ઝ ફંક્ષન ની
show શકે

ની આંદોલિન્ઝ foreach ની એસ્ટ્રીન્ઝ Anonymous function
નો use કરાવાની આપ એ એસ્ટ્રીન્ઝ
String ની અને એફ store કરાવાની આપ એ.

6 size

syntax of use

Console.log(map1.size); // only number

→ size method will map of outer both key
Ex. if there is 1st key in map of size contains

//newSet //

→

Const xset = new Set([1, 2, 3]);

variable

name

method name

start

set

end

Console.log(xset);

Set in which set hi
each value is key
Add 3 values
2nd & 3rd

→ method of use with set

1. add

syntax of use

xset.add("Hello");

→ ~~get~~ add add in set in ~~value~~ value in update
2nd 3rd 4th 5th 6th

2. delete

syntax of use

xset.delete("Hello");

→ delete का set में एकी value of delete gaurd
होती है।

3. has

Syntax of use

Xset.has("3");

→ has का मean का set में ~~में~~ value of check करता है।
यहाँ में जो वाली boolean value return होती है।

4. Values.

Syntax of use

XSet.values:

→ Values जैसे Set में एकी values console web में
दिये जाते हैं।

5. Size

Syntax. 4

Xset.size;

→ size का set में size show करता है।

29/01/24

II Javascript - Converting II

1. type of typeof(); OPERATOR

typeof operator 2111 variable ñi type check
sau 2111 प्राप्ति अ! console211i type shoco 211i 211

a. number();

number operator ñi variable ñi check 211
211 iñ number 211 iñ output 211 a number
show 211 iñ "NaN" show 211.

number() method 211 is 211 variable ñi 211 type
change 211 211 using 211 value 211 change
211 211-

Ex:- \Rightarrow string typeof convert in to \rightarrow number typeof

11 Converting //

→ Converting Variable to number.

1 → number() method

number method hi એ કોઈ variable ની જીએ એ કોઈ single number નું double inverted comma ("") સિગ single comma (') અને માણની ડાયરેન્ડ ફરી અને રાખને હિન્દી convert માણની ડાયરેન્ડ ની એ Variable નું Value આપે અને type એટા Number ડાયરેન્ડ થાય એ?

Ex :- `(num = "10.5")` = this convert true.

`num1 = "10";` } એવી એવી જીએ નું number એ કોઈ convert +
`num2 = "201";` } એવી એવી જીએ નું arithmetic operation એ કોઈ કરી શકે છે? (+, -, *, /, %)
`num3 = '200';`

But :-

એ કોઈ Variable ની જીએ કોઈ word માણા એ નું number એવું જોવા (space) જીએ માણની ડાયરેન્ડ ફરી ની હિન્દી Number Method કરતું તારી જીએ typeof number ડાયરેન્ડ but output એ નાન શોઓ પાય એ?

Ex :-

`num4 = "10 10";` } એવી એવી જીએ નું variable ની એવું
`num5 = "10/2";` } typeof એ નું number show કરો
`num6 = "10,2";` } નેટ્વર્ક console ની જોવો Null
`num7 = "Hello";` } print કરો.

* boolean :- // boolean in to number

જોવાની boolean true માણા false ની `Number(true)` / `Number(false)` એવી એવી Convert ગાયાની એવી જોવાની true = 1 અને false = 0 output એ જોવો ગાય એ!

boolean typeof => number એવી એવી એ!

=> Parsefloat() method // Point and value print sehu

Parsefloat એ floating point number ની print કરી શકતું હોય પરંતુ આજીથી એવી એવી એવી input દ્વારા આપે જાએ કે એ અને Parsefloat એ નંબર આપી શકતું નથી એ?

Note:-

Parsefloat એ Variable ની પણાનીકી number type એવી convert કરી શકતું નથી.

Ex:-

```
let num1 = "10.33";
Console.log(Parsefloat(num1)); // 10.33
```

```
num2 = "20 30 40";
Console.log(Parsefloat(num2)); // 20
// this Parsefloat print only First Value
```

```
num3 = "24 Year";
Console.log(Parsefloat(num3)); // 24
```

```
num4 = "Year 24"
```

This time in Parsefloat show NuN in console

→ Number.isFinite()

Aqui method 2121 Integer number or point number ની check કરી શકતું એ! એ value floating એ નિ true

// or

એ value number એવી હોય એવી એવી String એવી નિ false output કરશકી.

→ ParseInt() method

→ ParseInt අු න්‍යා තී Point හිත් මෙම න්‍යා න්‍යා පිළි
only Integer number print සේ ඇ? මෙම න්‍යා input
හිස් යන් Integer value or number නැත!

Note:

ParseInt අු number Variable තී number නි convert සෑ
මාය ඇ? automatically.

Note :- ParseInt අු ParseFloat අු වර්ක සෑ ඇ but
defens is ParseInt only tuk Integer
value.

Ex:-

```
num1 = ParseInt ("10.33"); // 10
num2 = ParseInt ("-105"); // -10
num3 = ParseInt ("10 year"); // 10
num4 = ParseInt ("year 10"); // NaN
```

→ * Number.isInteger();

→ Number.isInteger අු මෙයි මායි input තී සිදු
value මායි න්‍යා number මායි තී check සේ ඇ
න්‍යා Integer ඇ ඇ න්‍යා මායි method මෙම පෙ
ඩිස් ඇ!

Note :- //

Number.isInteger method ඇ only
true or false & Return සෑ ඇ //

සි Value or number මායි Integer සේ to // true
or if Value ඇ සිදු word or infinity සේ
සි // false.

⇒ Number.isSafeInteger() // method

→ isSafeInteger method in will pass the variable
as it's value is a integer or not. if it's safe
then it's integer will save else it will not
then this will return the value of check result
use this method so!

Ex:-

let x = "203040";

let y = 20104050608090100200.0010987;

Console.log (Number.isSafeInteger(x));

↳ This Return True; for x variable //safe

Console.log (Number.isSafeInteger(y));

↳ This Return false; for y variable //Not
safe

// Maths object //

- The JavaScript ·Math· object allows you to perform mathematical tasks on numbers.

1/ Math.PI

1) Note :-

- The ·Math· object is static.
- All methods and properties can be used without creating a math object first.
- The syntax for any Math property is :
`Math.Property`
- JavaScript provides 8 mathematical Constants that can be accessed as math Properties.

I. Math.PI

// Returns PI Value.

```
let x = Math.PI ;
```

```
Console.log ( X ); // 3.14
```

2. Math.E

// Return Euler's number.

```
let x = Math.E ;
```

($E=mc^2$)

```
Console.log ( X ); // 2.71
```

3. Math.round()

It returns the nearest integer Value.

Syntax :-

let M10 = Math.round(21.5);
→ output
22

let M10 = 21..5
let M = Math.round(M10);
output
21

→ round method on other positive number ਅਤੇ Point ਜਾਂਗ ਦਿਵਾ ਹੀ ਕਿ ਜੇ ਹੋਰੀ ਫਿਲ ਨੂੰ ਨਹੀਂ ਪਿਸ਼ੀ base Value ਹੀ plus one ਕਰਾ ਭਾਵੇਂ Point ਵਿਖੇ ਵਾਲੀ ਦਿਵਾ ਨੂੰ ਦਿਵਾ ਹੋਰੀ base Value ਦੀ ਸਮਾਂ return ਕਰਾ ਵੱਧ

→ Negative Value or Number.

let M12 = Math.round(-22.5);
→ output
-22

let M12 = Math.round(-22.6);
→ output
-23

Negative number ਨੂੰ other round method ਹੀ 5 ਵੇਂ ਹੋਰੀ Point Value ਦੀਂਦੇ ਹੋਰੀ base Value ਦੀ ਸਮਾਂ return ਕਰਾ ਵੱਧ

But

ਅਤੇ ਹੀ ਕਿ ਹੋਰੀ ਦੀਂਦੇ ਹੋਰੀ base Value ਦੀ ਸਮਾਂ plus 1 ਕਰਾ ਵੱਧ
Return ਕਰਾ ਵੱਧ

4. Math.ceil

// returns the value of x rounded up to its nearest integer.

Syntax :-

Let $M20 = \text{Math.ceil}(54.5); // 55$

$M20 = \text{Math.ceil}(-54.5); // -54$

$M20 = \text{Math.ceil}(-54.6); // -55$

Note:- Math.ceil method on positive and negative number gives round method on it will work safe.

5. Math.floor();

// math.floor(); return the value of x rounded down to this nearest integer.

let $x = \text{Math.floor}(20.5); \quad | \quad x = \text{Math.floor}(20.1);$
→ $x = \text{Math.floor}(20.6); \quad | \quad 20$
Output → 20 / 20

Positive { Math.floor method at point value 1 to 9 at it will return the base value & Return 8 if it is 1 plus than 0. (in positive number only)

* Negative

let $y = \text{Math.floor}(-20.1); \quad | \quad y = \text{Math.floor}(-20.6);$
→ -21 / -21

Negative { floor method at Negative value at point 1 to 9 at it will return the base value & Return 8 if it is 1 plus than 0. (in negative number only)

5. Math.trunc(); // point value is not counts

// Math.trunc(x) returns the integer part of x.

let x = Math.trunc(10); | x = Math.trunc(10.98);
output 10 | 10

* Negative

let y = Math.trunc(10); (-10.1); | y = Math.trunc(-10.8);
→ -10 | -10

→ trunc method एक only base Value ने &
यात्रा में शाफ्टी के Return करते हैं नियम पॉइन्ट वल्यू
के Count साथ जैसे एक Negative or positive
number छोड़ते हैं।

(only return base Value)

6. Math.Sign(); // Point value is not ~~Count~~

// return + if x is negative ; null or positive

→ Sign method एक base Value ने यात्रा में हेड
output द्याते हैं जैसे base Value positive हो
तो "1" Return करते हैं एवं base Value नु
जिस Negative होती है तो "-1" Return
करते हैं।

→ यह Value जैसे null or 0 होती है "0"
Return

7. Math.Pow()

→ Math.Pow() power એટાનું કરી જાય જે જો નિયમ માટે જો નિયમ એ?

Ex:-

2 * 2 * 2 * 2 * 2

let x = Math.Pow(2, 5); // 32

let y = Math.Pow(-2, 5); // -32

-2 * -2 * -2 * -2 * -2

અન્ય ફિસ્ટ વલ્યુ એન્ને નેગેટિવ એટિ રીતે નેગેટિવ ઓપ્ટિયન કરી.

But:-

અન્યથી:- x2 = Math.Pow(-2, -5); // -0.03125

x3 = Math.Pow(2, -5); // 0.03125

→ એ એક મુજબ એન્ને વલ્યુ નેગેટિવ એટિ રીતે નેગેટિવ એન્યુન્ટ કરી

→ એ એ મુથોડ એન્ને સેકોન્ડ વલ્યુ નેગેટિવ એટિ રીતે નેગેટિવ એન્યુન્ટ કરી.

8. Math.Sqrt();

// return the square root of x.

Sqrt method એનું નંબર એન્ને સ્ક્વાર રોટ કરી જાય એ?

Ex:-

let x = Math.Sqrt(144); // 12

But Ex:- let x = Math.sqrt(-121); // Null

Sqrt method hi negative value aur aur hi aur
nhi fi output hi " Null " show kri root
nhi.

9. Math.abs()

→ abs ki thi Pass gya hi aur hi kriya.
Negative value ki positive value or
Return kriya.

Ex:-

	out put
let x = Math.abs(-10);	10
let y = Math.abs(0);	0
let z = Math.abs(-0);	0
let o = Math.abs(-10.21);	10.21

10. Math.min() // alli aayi values min lowest + find kriya
Math.max() // alli aayi values min highest find kriya

Ex:-

let x = Math.min(-1, -10, 20, 40, 5); // -10

let y = Math.max(-10, -1, 20, 45, 5); // 45

11. Math.random():

random method द्वारा "0" to "1" का एक संख्या
नंबर बनाती है जो अपने तरीके से

Ex:-

let x = Math.random(); // 0.12

// 0.1214214...
etc

→ If we multiply random by 10 or 100 or 1000 then
we have a integer value then we can use it.

Ex

Math.random() * 10 // 1, & to 10

" " * 100 // 11, 90, 99 & to 100

" " * 1000 // 999, & to 1000

// Task //

1. `Console.log("9" > "19");` ① {only count first number
`Console.log("09" > "19");` ② }
- me { ① false | in check out is true
 ② false | this fluse }

Because

in string ("9" > "19") → (9 > 1) → true
 | count only this
 (Numbers)

("09" > "19") → (0 > 1) → false
 | first | second

2. `Console.log(1 + 2 + '1');` // 31
`Console.log('1' + 1 + 2);` // 22

in second question is output is '1' + 1 count
then + 2 count // 11 22

3. `let x = 100;` first print then increment
`let y = x++;` $y = 101$
`let z = ++x;` $z = 102$ first increment then print

`let n = (x == y) ? z++ : ++z;`

me { $n = 102$

103

3. Var $x = 0;$

```

if ( $\frac{V}{(10 > 9)} \frac{V}{(9 > 8)}$ ) {
     $x + x = 1;$ 
}

```

Output is
false condition
 $x = 0$

Console.log(x); // 1

মাত্র একটি পরিস্থিতি দুটি first two number condition
is Right $(10 > 9)$ true Return ১; result
এবং $1 > 8$ কে $(1 > 8)$ চেক করো এটি
false এবং ফলী output false আসে।

Ex:- V

```

if ((10 > 9) = 1) } false | if((10 > 9) = 1) true
                (1 > 8) = 0 | not work like this

```

4. Const f = new Boolean(false): null

```

if (f) { console.log(1); }
else { console.log(2); }

```

Output
2 undefined

Console.log(typeof(f));

in Compiler

Output is

{ 2 }
object

check with

primitive value

null, undefined, = false

true
false

5 Var bar = function foo(){};

Console.log(bar == foo);

this go for flase or show error
because of foo is function here.

(bar) = Function

(bar) = f foo CD

foo() { body } →

{ bar == foo }