* Console. log () :

To write (log) a message on the console.

```
Console. log (" Apna College ");
Console. log (123);
Console. log (" Ro ", " HI ", " T ");
```

* Linking JS File :-

1] `<Script src = "app.js" > </script>`

2] Inline JS:

```
<button onclick = "alert ('Hey there! greeting From codedamn')"
    > Welcome Message </button>
```

3]

```
<Script>
    alert ( );
</Script>
```

* Templete Literal:

They $\overset{re}{a\cancel{dd}}$ used to add embedded expression in a string.

```
e.g let a = 5,    b = 5;
    Console. log (`sum is ${a+b}`);
```

$\underbrace{\$\{a+b\}}$

① Calculate

② embeded in string

* Conditional Statement :

~~else-if statement~~

if - else

nested if-else

switch

1] if statement :

```
if ( condition)
{
    // Do something
}
```

2] else-if statement

```
if( condition)
{
    //Do something
}
else if (condition2)
{
    // Do something else
}
~~else {~~                              (optional)
    //Do something else
}
```

3) if - else

```
if ( condition)
{
    // Do something
}
else {
    //Do something else
}
```

4] Nested if-else

- nesting is writing if-else inside if-else statement.
- in can many level

```
if( )
{
    if( )
    {

    }
    else{
    }
}
~~else~~
{
```

Page No.
Date

* Truthy & Falsy Value :

Falsy : false , 0 , -0 , 0n (Bigint value), " ", null , undefined
NaN

True : Everything else

* Switch statement :

Used when we have some Fixed values that we need to compare to.

```
e.g let col = "red";
    switch (col)
    {
      case "red":
              console.log("red");
              break;
      case "green"
              console.log("green").
              break
        :
        :
      default :
            console.log ("Broken light");
    }
```

* Alert :

Alert display an alert message on the page.

alert (" message");

* Prompt :

Prompt display a dialog box that asks user for some I/P.

Prompt (" please enter your roll no");

Console. error ();

Console. worn ();

* String Method *

* String is ~~col~~ sequence of character.

* Method :- Actions that can be performed on object.

* format : StringName. method ();

* Trim method :-

trim () remore whitespace from both end string & return new one.

e.g let s= " ROHIT ";

S.trim() → O/P : "ROHIT";

Page No.
Date

* String are Immutable in JS :

+ No change can be made to string
+ Whenever we do try to make a change, a new string is created & old one remain Same.

* ToUpperCase() & ToLowerCase() :-

e.g 1] let str1 = Rohit $\xrightarrow{\text{str1.ToUpperCase()}}$ O/P : ROHIT
e.g 2] let str2 = ~~rohit~~ ROHIT $\xrightarrow{\text{str2.ToLowerCase()}}$ O/P : rohit

* String method with argument:

Argument is some value that we pass to the method.

format:

```
str.method($\overset{arg}{value}$);
```

1) ~~Idex~~ indexOf() :
Return the First index of occurance of some value in string. or give -1 if not Found.

e.g
```
let str = "ILoveCODING";
str.indexOf("I");      //0
```

Method chaining:
Using One method after another.
Order of execution will be left to right.

~~str.toUpper case~~ `str.ToUpperCase().trim();`

ii) Slice () :
Return a part of the original string as a new string.

end index is non-exclusive

```
let str = "ILovecoding";
    str.slice (5);     // Coding
    str.sice (1,2);    // love
```

iii) Replace () :
Search a value in the string & return a new string with the value replaced.

```
let a = Rohit
    a. replace ("R","T");
    o/p : TOHIT
```

iv) Repeat () :
Return a string with a number of copies of a string.

```
let a = fruit
    a. repeat (3);
    o/p : fruit-1,-11
```

* Array : Collection of item of ~~same~~ different type.

```
let student = ["Rohit", "Ram", "Arjun"];
```

Access through index:

```
student [0];
student. length ~~()~~;
```

Array is mutable

Array Method :-

$$\begin{tikzcd} & & \text{add to start} & \\ \text{Push ()} \arrow[d] & \text{, pop()} \arrow[d] & \text{, Unshift ()} \arrow[u] & \text{, shiF()} \arrow[d] \\ \text{add at end} & \text{remove from end} & & \text{delete from end \& return it} \end{tikzcd}$$

i) indexOF (): Return index of something

```
arr.indexOF(1);
```

ii) includes(): ~~ret~~ search for value

```
arr.includes(1);
```

iii) Concat : merge 2 array

```
arr1.concat(arr2);
```

iv) reverse : reverse an array

```
arr.reverse();
```

v) Slice () : Copies a portion of an array.

```
arr.slice(5);
arr.slice(2,3);
```

vi) Splice () : remove / replace / add element in place.
Splice (start, deletecount , itemo... itemn)

e.g :

```
Color = ["red", "Yellow", "blue", "Orange", "Pink", "white"];
Color.splice(4);              // Pink , white
Color.splice(0,1);            // red
Color.splice(0,1, "black", "grey");   // yellow.
```

Page No.
Date

number
↓
string

vii) Sort : sort an array.

```
num = [1, 4, -1, 2, ]
num.sort();      [-1, 1, 2, 4]
```

* Array Reference :

```
[1] == [1]   -> false
[1] === [1]  -> False
```

reference variable represent not a value.

* Const Arrays:

```
const arr = [1,2,3];
```

copy not allowed

* Nested Array :

```
let num = [[1,2], [2,3], [3,4]];
```