expression ← la
operation

Arithmetic Operators

i) + → addition
ii) - → subtraction
iii) * → multiplication
iv) / → division
v) % → modulus / remainder
vi) Exponentation;

UNARY OPERATORS: (single operant

- Increment
  - → Post Increment (a++)
  - → Pre Increment (++a)
- Decrement
  - → Post decrement(
  - → Pre decrement(--

Comparison Operators (boolean result)

== equal to (values)
=== (equal to (value + same data types)
!== (not equal to & data types also)
!= (not equal to)

```
a = a % 4;
a = a * 4;
```

Assignment Operator

i) `+=` → `a = a + 4`

ii) `-=` → `a = a - 4`

iii) `*=` → `a = a * 4`

iv) `%=` → `a = a % 4`

v) `**=` → `a = a ** 4`

vi) `=` → `a = b`

Comparison Operator (boolean result)

`>=` greater than and equal to

`<=` less than and equal to

Logical Operator

Logical	AND	&&	both conditions must be true
Logical	OR	||	anyone of them true
Logical	NOT	!	Result Inverted

(works on 2 condition)

(works on 1 condition)

Conditional Operator

i) If statement	Terenary Operator
2) If-else	condition : true or false
3) if-else-if	output.

If statement, checks conditional

→ syntax:

```
if (condition)
{
    block of code;
}
```

i) true
ii) → if condition will b it execute
iii) block of code. otherwise ign
it

If - else:

→ syntax:

```
if (condition)
{
    block of code;
}
else
{
    block of code
}
```

else - if statement:

```
if (condition)
 {
   block of code;
 }
space
else if(condition)
{
 block of code;
}
space
else if (condition)
{
 block of code;
}
else
  {
  }
```

It checks all the condition which condition will be true it executes the block of code associated with that condition

→ Ternary Operator:

* is also logical operator
* can check condition in one line.

Syntax

*) condition? "true value": "false value";

msg which will be print when condition is true

Class: 07

SWITCH STATEMENT:

```
let variable = prompt("_____");

switch (variable)
    case "yes";                    → condition
    document.write("_____");
    break;
    case "_____";
    document.write("_____");
    break;
    .
    .
    .
    default:
    document.write("_____");}
```

STRING

→ String is a sequence of characters used to represent a text
→ It is primitive data type
→ We can create string by using template literals and in single and double quotations.

String Creation & Manupulat

→ let variable = "_____"

→ let variable = '_____'

→ let variable = `_____`; use to embedded expression by using backtick string interpolation

→ let variable = `_____ ${variable}_____ ${variable}_____`;

placeholders values calculated at run time and this process is called string interpolation

Tempelate Literal 'Use'

Tempelate literals are the new feature of Js introduced with ES6. They give more flexible and easier way of working with string in Js

* works on console.

How to use Tempelate literals

→ \n → for new line

→ \t → for 1 tab space

→ \\ → for single slash

→ \" → for 1 side quotation

→ length of string;
- l= string name.length;
- It also count spaces.
- for specific Indexes (str[0]); (Index ↑)
→ Index numbers starts with "0";

Some Properties & METHODS:

→ To join String

```
document.write(str1+" — "+str2);
document.write( str1+str2);
```

→ by concatination (برابر لکھنے + کے بجائے)

1st string ↑ with second string ↑

```
let new = str.concat(str2);
```

new ↓ variable name

str ↑ 1st string

To join ↓

```
let new = str.concat(str2, str3);
```

str2 ↑ 2nd

str3 ↑ 3rd.

TRIM METHODS

(does not change original string)

→ (string.trim()) : removes space from start & end.

→ trimStart() : removes space from start

→ trimEnd() : removes space from

↔ str.toUpperCase(); to capatil each letter

→ .toLowerCase(), to small each letter

→ .replace("1st word", "2nd word") to change
↓ case sensitive ↓ one word
to change with word

→ replaceAll("____"); // replace all the matching val

→ .includes("word that has to be find");
↓ results in boolean.

→ .slice(5, 10) : It bring words from
↓
1st index number to second index number. (used in calculator) (last index is excluded)

→ const cars = ["____", "____", "____"];
cars[0] = "____"; // change the first item to "____"

→ .charAt(index number); // to search characters of index

Class: 08

Loop :-

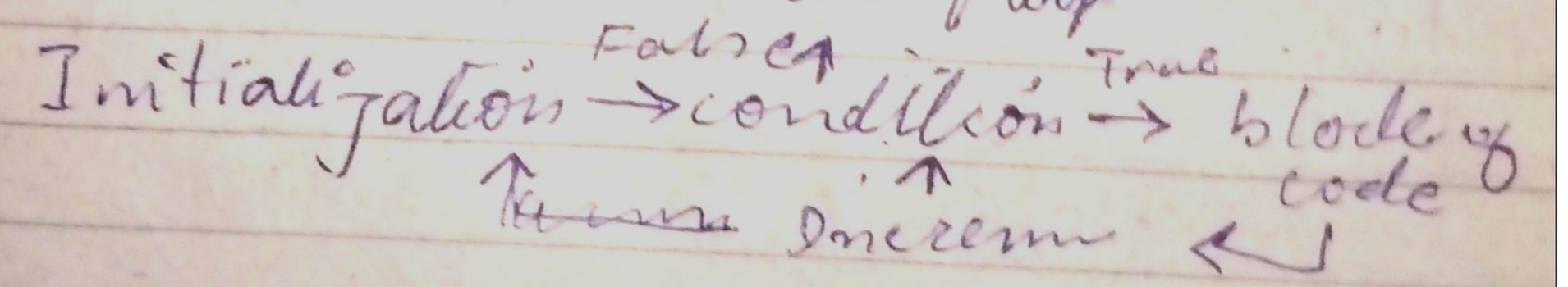
→ to execute a piece of code again and again
→ finite loop and infinite loop
→ end ℰ does not end
→ infinite harmful computer

1) For Loop :

```
for (let i = 1; condition; Increment/decrement).
{
    block scope variable
    blocks of code
}
```

(1) (2) (4) (3)

- 1st step Initialization of variable
- 2nd step is condition testing then execution of block
- 3rd step is Increment and decrement out of loop



Array (non-primitive Data type)

→ stores multiple value in single variable
→ declared in square bracket.
→ values seperated by commas
→ each position of keys is called index. - [0p, 1p, 2p, 3p]
↓ keys

Syntax:-

=[5, "——", "——"];

=[]; you specify index

42:30

2) For Of Loop (Array, step)

for (let variable of declared variable)
{
block of code (variable);
}

Class #09

PROPERTIES & METHOD In ARRAY

let arr = ["book", "copy", "pencil", "Ruler"];

→ arr.length (it counts the length including whitespaces (results in numbers)

→ arr.push("item want to push"); // insert element in the end of array

→ arr.unshift("item to insert"); // adds item in the start of the array.

→ arr.shift("——"); remove item from start (0 position)

→ arr.pop(); // removes item from the end.

→ arr.toString(); // converts array into string

→ arr.concat(arr2); // join 2 array but without changing 1st array we have to take 3rd variable to print its result.

→ for two strings → str1 + str2 = x let.

→ arr.indexOf("item of array which you want to determine the position");

→ arr.slice(starting index, ending index);
// return array elements from starting given index to ending index given gives 1 number less than the ending number which is declared (works only on string.

→ arr.slice(starting index); // returns the value from ↓ till the end of the array.

→ arr.splice(starting ind, no of items to replace, "string from which you want to replace")

→ arr.splice(starting, remove);

(starting, 0, replace);

سب سے پہلے والے index پر جائے گا اور جتنے second number میں لکھیں گے اتنے items لاد دے گا۔

Class # 10

revision of class #09

For Splice

جب splice کو use کریں گے تو سب سے پہلے ہم لکھیں گے
document.write (array.splice(…,…,…)); جس میں سب سے پہلے number اس
کے بعد ہو گا جس پر comp کو جانا ہے پھر اس کے لئے
جتنے Items کو delet کرنا ہے پھر اس کے لئے جس
Items سے اسکو replace کرنا ہے- replace کرنے والے
1 Items سے زیادہ بھی ہو سکتے ہیں- پھر وہ Items کو
show کرے گا جو delete ہو گئے اسکے بعد جب آپ
again اس array کو print کروائیں گے تو وہ
actual جو ہو گا اس کی بجائے جو بنا ہو گا
addition اسکو print کر دے گا۔

for example

```
let array=[0,1,2,3,4,5,6];
document.write (array.splice(1,4,0,6,3));
document.write (array);
```

Items: OUTPUT:-

~~0-6~~ 1,2,3,4 // deleted Items
0,0,6,3. // current array.

جہاں کچھ نہ کروانا ہو وہاں 0 لکھیں گے

OBJECT:

Store multiple values in a single variable
values written in curly brackets {} in pairs with keys.

Syntax:

let/const variable name = {

key: 1,
key: 2,
key: 3,

}

→ variableName.key // for access specific key.

For In Loop (for object, Arrays)

→ for (let Key in student)
↓ variable — declared variable (already) array.

{

→ document.write(Key); // for just Key not their values

→ document.write(student[key]); // for their values

→ variable = (declared / array)

→ variable.key → for specific key

→ even k liye for of use karenge
or odd k liye for in loop

→ returns keys not value

While Loop

let variable declaration (i) = "——" / Num;

while (i <= 10)

{
block of code (~~variable~~) "wants to print text")
i++; → increment / decrement
}

Do WHILE LOOP :-

let sum=0 ; let i=1;

do
{
document.write(——); | block of code
sum= sum+i;
i++;
}
while(condition); // i<=10.

ESCAPE CHARACTERS :-

\n backslash "n" → new line
\t one tab space
\" for single time double quotes →
\\ for single backslash.

They are written as two character in a string but count as only one at the time of calculating length They ignored by printin and works only on console first two

Practice Question :-

Get ElementBy Id ("id name"). inner HTML = "Hello Javascript";

use the correct array method to add "kiwi" into array.

const fruits = ["—", "——", "——"];
fruits.push("Kiwis");

→ arrayname.sort( ); → to arrange array alphabetically.

→ to stop loop "break"; statement is used.

→ old javascript :- <script type = "text/javascript

display PROPERTIES :-

→ inner HTML
→ count.