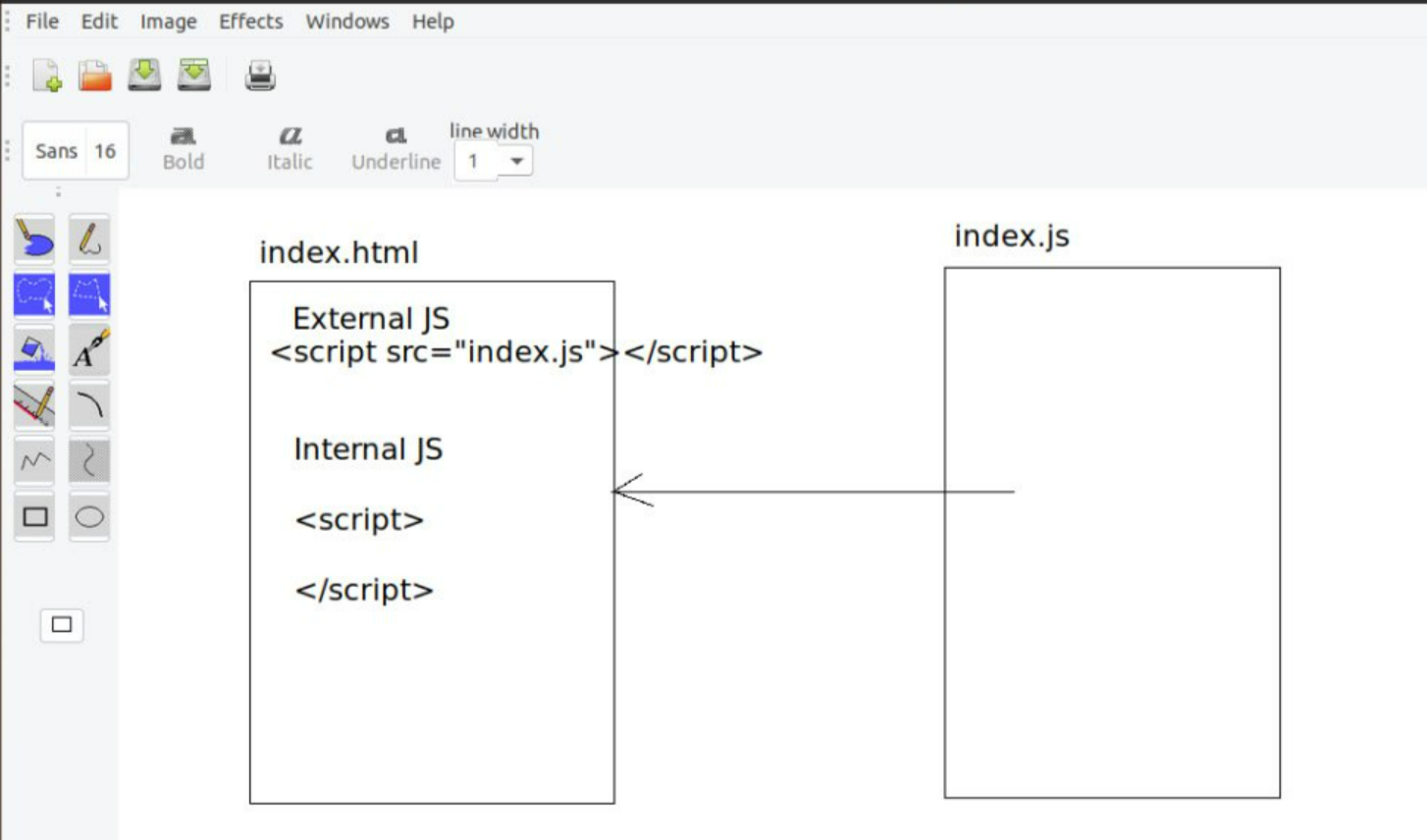
Day-9 java script

To include js file to html we have to use script tag

Eg: script



Execute js in 3 ways

1. Browser – interpreter
2. Nodejs- node filename.js

**Demo.html**

<!DOCTYPE html>

<html>

    <head>

        <title>JS BASCIS</title>

    </head>

    <body>

**<script>**

**//to display on webpage**

**document.write("welcome to klu","<br>");**

**//to display of output device**

**console.log("hello");**

**// to display data in form of table**

**console.table([10,20,30,40]);**

        </script>

    </body>

</html>

Output : liveserver

Rightclick browser – inspect- console- ur output will be displayed.

TypeCoersion && NAN

<!DOCTYPE html>

<html>

    <head>

        <title>JS BASCIS</title>

    </head>

    <body>

        <script>

            //to display on webpage

            document.write("welcome to klu","<br>");

            //to display of output device

            console.log("hello");

            // to display data in form of table

            console.table([10,20,30,40]);

**document.write(10+"10","<br>")**

**// to convert string to number**

**document.write(10+ +"10","<br>");**

**document.write("1"+"2"+"10","<br>")**

**// -, /,\*,% converts string to number**

**//conversion as typecoersion**

**document.write(10-"10","<br>");**

**document.write(10\*"10", "<br>");**

**document.write(10/"10","<br>");**

**document.write(10/"0","<br>"); //infinity**

**document.write(10/"A","<br>"); //NAN not a number**

**</script>**

**</body>**

**</html>**

**//=(assignment)**

**//== comparision ignores datatype**

**//===(strict comparision) both values and type**

  document.write(10 == "10" , "<br>");// will compare only values

  document.write(10 === "10" , "<br>");

 document.write(0.1 +0.2 , "<br>");

 document.write(0.1 +0.2  == 0.3 , "<br>");

document.write(0.1 +0.2  === 0.3 , "<br>");

output :

0.30000000000000004  
false  
false

**//true--1**

**//false--0**

**document.write(true+1 , "<br>");//2**

**document.write(true/false , "<br>");//infinity**

**document.write(1+"true" , "<br>");//1true**

**document.write(true+1 - "1", "<br>");//1**

**document.write(3>2>1,"<br>"); //false**

**document.write(1<2<3,"<br>"); //true**

**//bodmas**

**//b-bracker**

**//o-of**

**//d-division**

**//m-multiplication**

**//a-addition**

**//s-subtration**

**document.write(10+5 / 10-5 , "<br>"); //**

**document.write((10+5) / (10-5) , "<br>"); //3**

**//ternary operator**

**//condition ? block1 : block2**

**9>8>7 ? document.write("jfs") : document.write("jfsd");**

**Variables:**

* **Store the data**
* **1. Var**
* **2. Let**
* **3. Const**
* **4. Without any keyword**

**Let & const 🡪 ES6**

**To declare variables some rules**

1. **a-z, A-Z, 0-9, $ and \_(underscore)**
2. **should not start with numbers**

**ex: varibles.html**

<!DOCTYPE html>

<html>

    <head>

        <title>JS BASCIS</title>

    </head>

    <body>

        <script>

            var x=100;

            document.write(x);

            // as per developer point of view multiple declaraions should not be done

            var x=100;

            var x=200;

            document.write(x);

           // let x=100;

            //let x=200;

          //  document.write(x);// error x is already declared

        </script>

    </body>

</html>

**//global variable**

**let x=100;**

**// document.write(x);**

**{**

**//local variable**

**let x=200;**

**// document.write(x);**

**}**

**document.write(x);//100**

**varible declaration**

**variable initialization**

**variable assign**

**ex:**

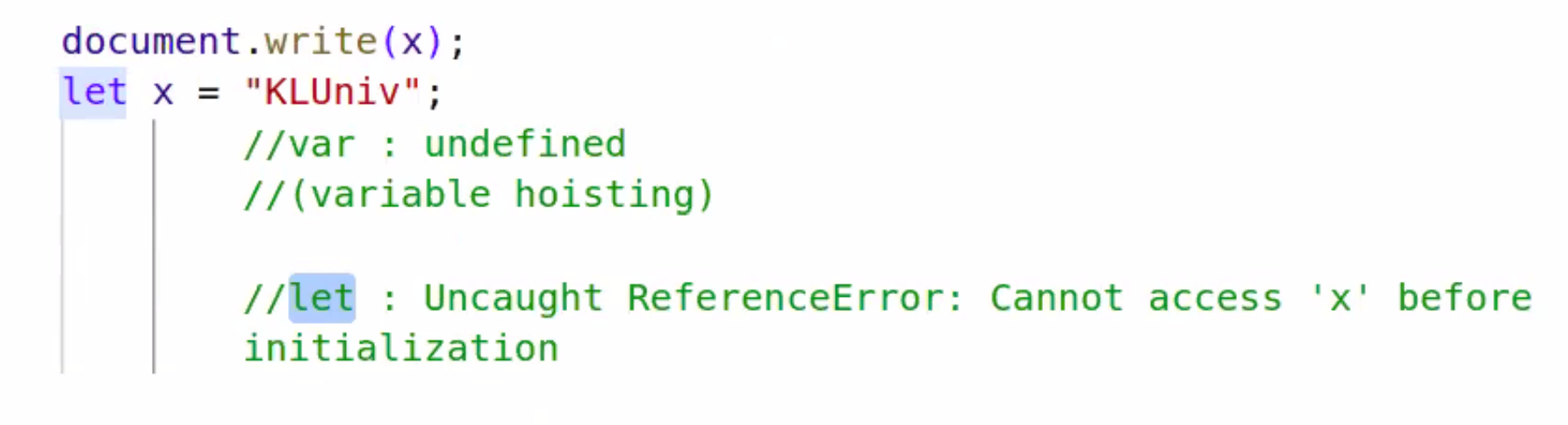
**var x="klu";**

**document.write(x);**

**ex:** o/p undefined will be the output

**document.write(x);**

**var x="klu";**



Functions:

function klu(){

                var x=100;

                document.write(x);

                {

                    let y=200;

                }                 document.write(y);

            }

            klu();

 function klu(){

                var x=100;

                document.write(x);

                {

                    let y=200;

                }

                document.write(x);

                document.write(y);

            }

            klu();

correct interpreter is correct till x value and y is error

second case:

 function klu(){

                var x=100;

                document.write(x);

                {

                    let y=200;

                }

                document.write(y);

document.write(x);

            }

            klu();

**interpreter will stop at y**

**and x will also stop**

String:

1. “ “
2. ‘ ‘
3. ` ` (backtick)
   1. Backtick- template literal
   2. Introduced in ES6
   3. Easily we can concat the strings
   4. Multiline string

Ex:

 let msg= `devops`;

           let wish=`welcome to ${msg}`;

            document.write(wish);

//let msg= `FSD with devops`;

           //let wish=`welcome to ${msg}`;

           // document.write(wish.substr(0,15));

**For large data type- bigint**

**Range of bigint is 2power 53-1 to -2power53-1**

Big int suffix with “n”

**let x=123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789123456789n;**

            document.write(x);

**var x=100; //varibal einitlization**

**var is keyword**

**x- variable name- identifier**

**100-value or literal**

**I want to hide the identifier**

To achieve security we have one datatype

**Symbol datatype**

**Var x=”klu”;**

**let x= symbol("klu");**

**document.write(x);**

**very important**

**declare var;**

**don’t assign any values**

**we call undefined**

**undefined with any arithmetic operations result will be** NAN

**var x;**

**document.write(x,"<br>");//undefined**

**document.write(x+10,"<br>");**

**//null represents empty or blank values**

**//while performing any arthmetic operation null value sconverts to zero**

**//null represents empty or blank values**

**//while performing any arthmetic operation null value sconverts to zero**

**var x = null;**

**document.write(x,"<br>");**

**document.write(x+10,"<br>");**

**I want to know datatype**

**We have operator**

**//typeof-operator**

**//typeof**

**document.write(typeof "klu","<br>");**

**document.write(typeof 100,"<br>");**

**document.write(typeof true,"<br>");**

**document.write(typeof undefined,"<br>");**

**document.write(typeof null,"<br>");**

**document.write(typeof 100n,"<br>");**

**document.write(typeof Symbol(100),"<br>");**

**document.write(typeof [],"<br>");**

**document.write(typeof {},"<br>");//json object**

**document.write(typeof function klu(){},"<br>");**

**data will be staored in the form of bits**

**first 3 bits represents type of data**

**0 1 1-🡪 number**

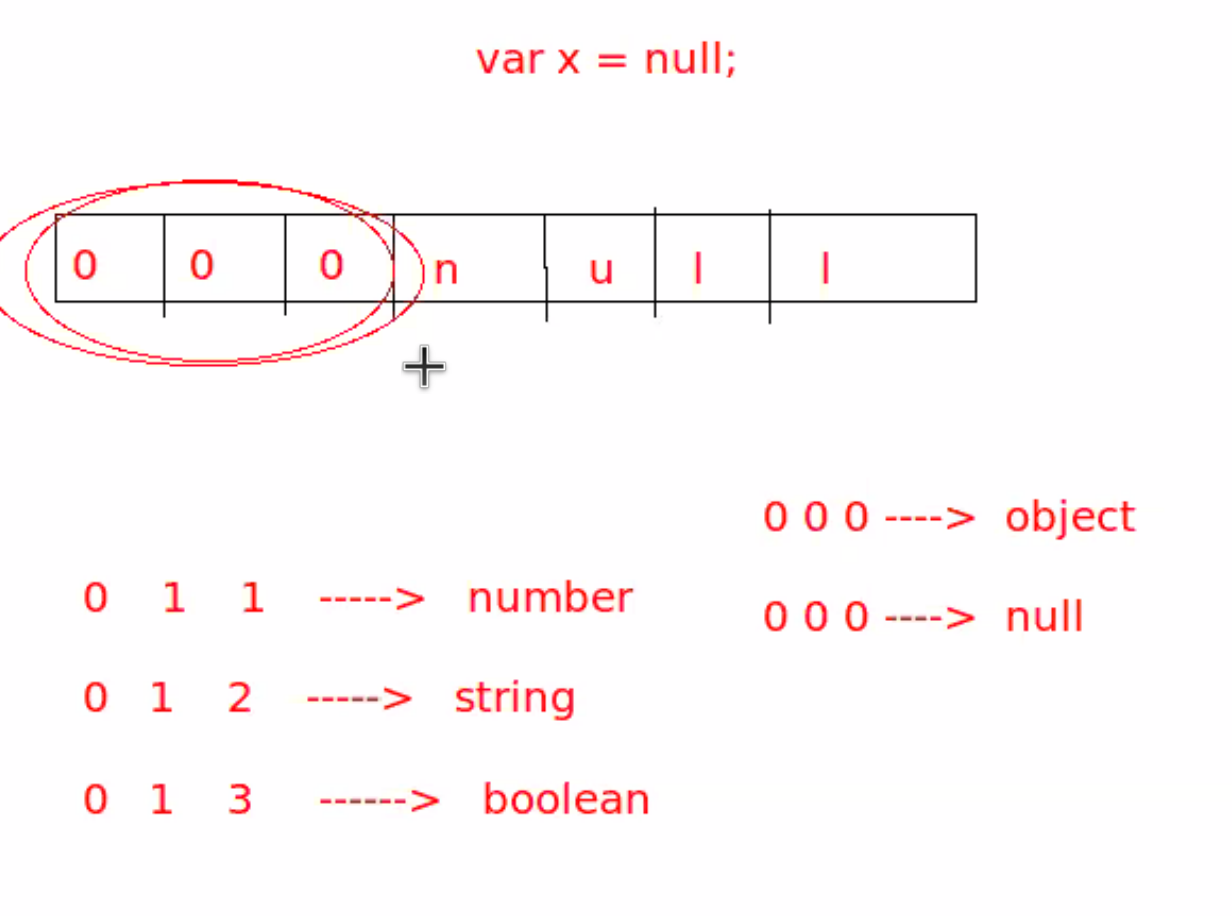
**0 1 2 -🡪 string**

**0 1 3-🡪 Boolean**

**0 0 0 🡪 object**

**0 0 0 -> null**

**Mistake by nescape company**

****