**TASK 1**

1) <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width, initial-scale=1.0">

<title>task1</title>

</head>

<body>

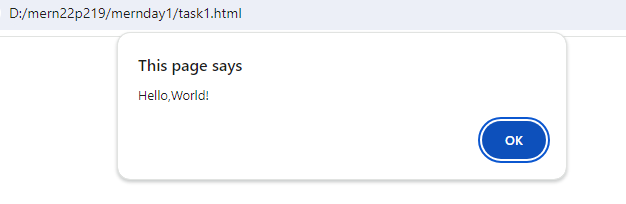
<script>

alert("Hello,World!");

</script>

</body>

</html>



2) <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width, initial-scale=1.0">

<title>task1</title>

</head>

<body>

<script>

let num=10;

let bol=true;

let str="char";

let nulval=null;

let undef;

console.log(num);

console.log(bol);

console.log(str);

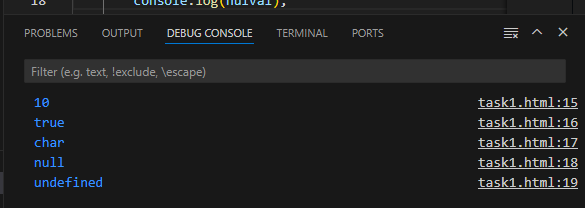
console.log(nulval);

console.log(undef);

</script>

</body>

</html>



3) <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width, initial-scale=1.0">

<title>task1</title>

</head>

<body>

<script>

let num=5;

let num2=36;

console.log("sum:",num+num2);

console.log("sub:",num-num2);

console.log("mul:",num\*num2);

console.log("div:",num/num2);

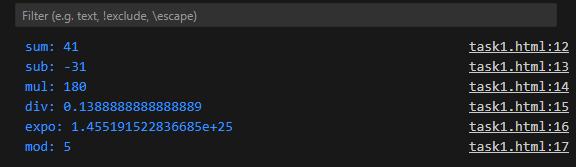
console.log("expo:",num\*\*num2);

console.log("mod:",num%num2);

</script>

</body>

</html>



4) <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width, initial-scale=1.0">

<title>task1</title>

</head>

<body>

<script>

var s1="indu";

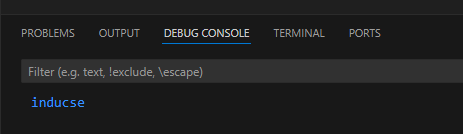
var s2="cse";

console.log(s1+s2)

</script>

</body>

</html>



5) <!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width, initial-scale=1.0">

<title>task1</title>

</head>

<body>

<script>

let num=10;

let bol=true;

let str="char";

let nulval=null;

let undef;

console.log(typeof num);

console.log(typeof bol);

console.log(typeof str);

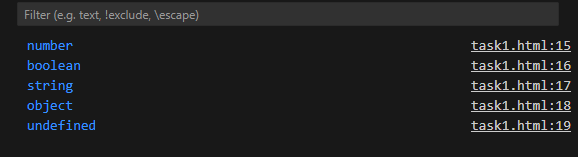
console.log(typeof nulval);

console.log(typeof undef);

</script>

</body>

</html>



2)

6)

<script>

let num=10;

let undef;

/\*This is multiline

comment created using slash and asterisk and it is ignored by jsinterpreter\*/

console.log(typeof 0);

//This is single line comment!!!!!;

</script>

7)

<script>

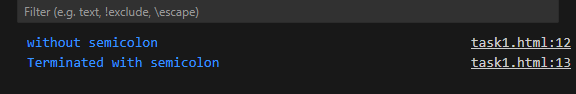
let num=10;

let undef;

console.log("without semicolon")

console.log("Terminated with semicolon");

</script>



8)

<script>

for(let i=0;i<n;i++){

for(let j=0;j<n;j++){

console.log(j);

}

}

</script>  
9)

<script>

let user,pswd;

user="indu";

pswd="cse";

</script>

10)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width, initial-scale=1.0">

<title>task1</title>

<script>

console.log("Hello world!");

</script>

</head>

<body>

<script>

let user,pswd;

user="indu";

pswd="cse";

console.log("credentials:",user+pswd);

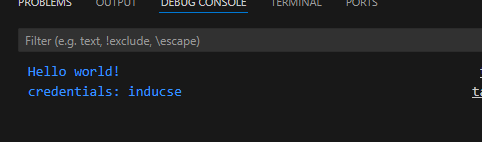
</script>

</body>

</html>

There is difference when script is placed in head tag or at the beginning of body it is excuted before the document is fully parsed.

It is best practice to write script at end of body.



**2)”use strict”**

**1**1)<body>

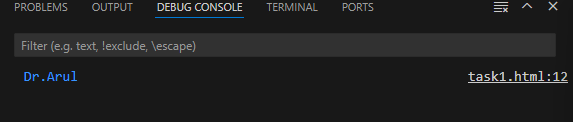
<script>

admin="Dr.Arul";

console.log(admin)

</script>

</body>



12) <body>

<script>

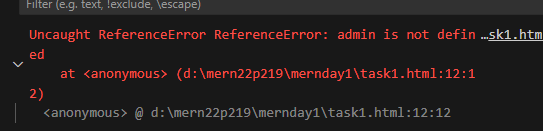
"use strict";

admin="Dr.Arul";

console.log(admin)

</script>

</body



13)

<script>

"use strict";

let admin="indu";

delete admin;

function showmsg(name){

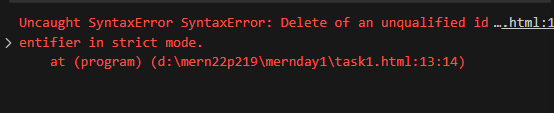
delete name;

}

delete showmsg;

</script>

In strict mode we cant delete a variable,function or function parameter.



14)

In stricty mode a variable cant be assigned before declaration it shows referenceerror .But in non strict mode it can be assigned ro a variable before its declaration.

15)

<body>

<script>

"use strict";

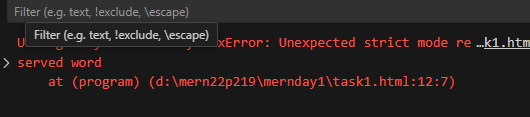
let break="its lunch at 1.25pm";

var implements=100;

</script>

</body>

In strict mode we cant use reserved keywords as variable name.



16)

<body>

<script>

let a=10;

var age=20;

var age=18;

const CLS=10;

</script>

</body>

When we don’t want a value to be changed we can use const keyword.And when we need to access the variable from anywhere and allowed to redeclare we use var kwyword.Let is used when variable is to be accessed only within local scope and to avoid redeclaration of variable.

17) <body>

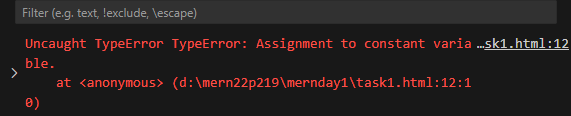
<script>

const CLS=10;

CLS=12;

</script>

</body>



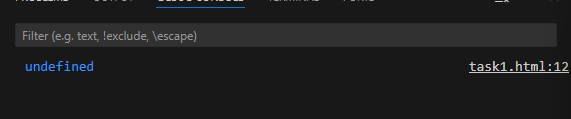
18)

<script>

let name;

console.log(name);

</script>



19)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width, initial-scale=1.0">

<title>task1</title>

</head>

<body>

<script>

let num=10;

let bol=true;

let str="char";

let nulval=null;

var undef;

console.log(typeof num);

console.log(typeof bol);

console.log(typeof str);

console.log(typeof nulval);

console.log(typeof undef);

</script>

</body>

</html>



20)

<script>

let a=10;

let b=a;

delete a;

console.log(a,b);

</script>



21) <script>

let num=10.0;

let bol=true;

let str="char";

let nulval=null;

var undef;

var obj={}

console.log(num);

console.log(bol);

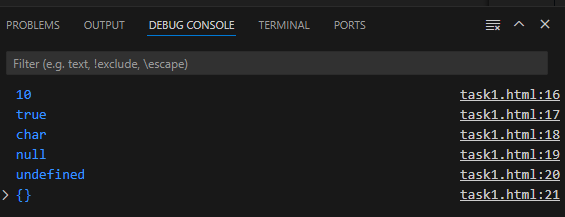
console.log(str);

console.log(nulval);

console.log(undef);

console.log(obj);

</script>



22) <script>

let num=10.0;

let bol=true;

let str="char";

let nulval=null;

var undef;

var obj={};

console.log(typeof num);

console.log(typeof bol);

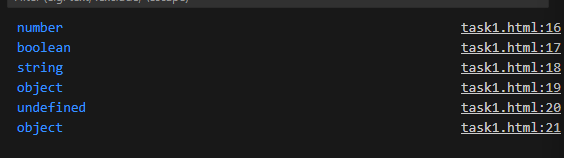
console.log(typeof str);

console.log(typeof nulval);

console.log(typeof undef);

console.log(typeof obj);

</script>



23)

<script>

var obj={};

let sym=Symbol(name);

console.log(typeof sym);

</script>

  
24)

<body>

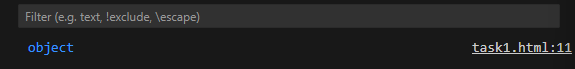
<script>

var nullval=null;

console.log(typeof nullval);

</script>

</body>



25)

While declaring a variable using let it has only block level scope.while variable declared using var has global scope that is it can be accessed anywhere.

26) <script>

let str="232";

console.log(str\*1);

console.log(Number(str));

</script>



27) <script>

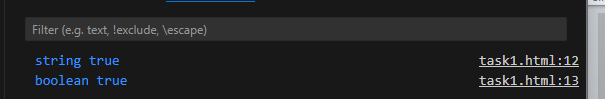
let flag=true;

let flstr=String(flag)

console.log(typeof String(flag),String(flag));

console.log(typeof Boolean(flstr),Boolean(flstr));

</script>



28)

<script>

let num=52;

let num2=30;

console.log("sum:",num+num2);

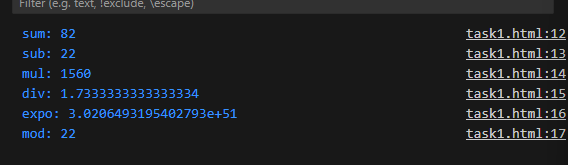
console.log("sub:",num-num2);

console.log("mul:",num\*num2);

console.log("div:",num/num2);

console.log("expo:",num\*\*num2);

console.log("mod:",num%num2);

</script>

29)

<script>

var a=10;

console.log(a++);

console.log(++a);

console.log(a--);

console.log(--a);

</script>



30)

<script>

var a=10,b=20,c=32,d=233,e=89;

console.log(10+10/5\*20+45-45);

</script>

