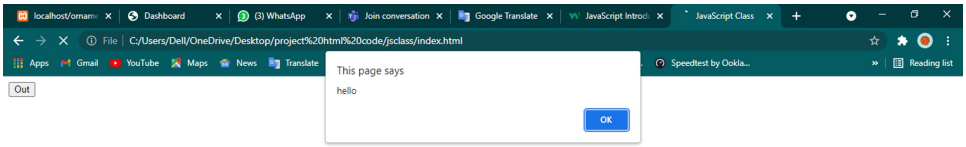
javascript all classes notes

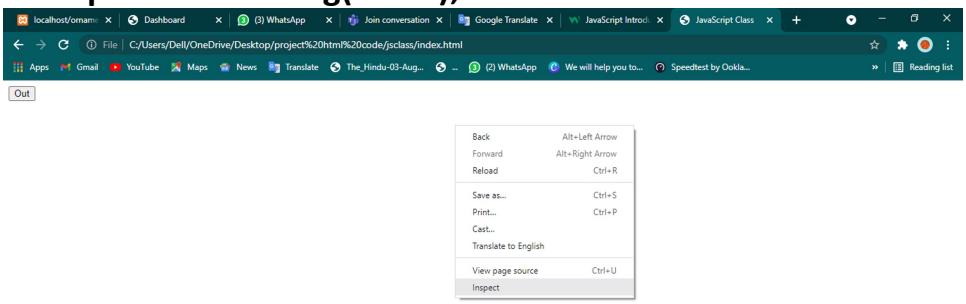
Tuesday, October 19, 2021 9:07 PM

4 types of functions to show javascript on browser

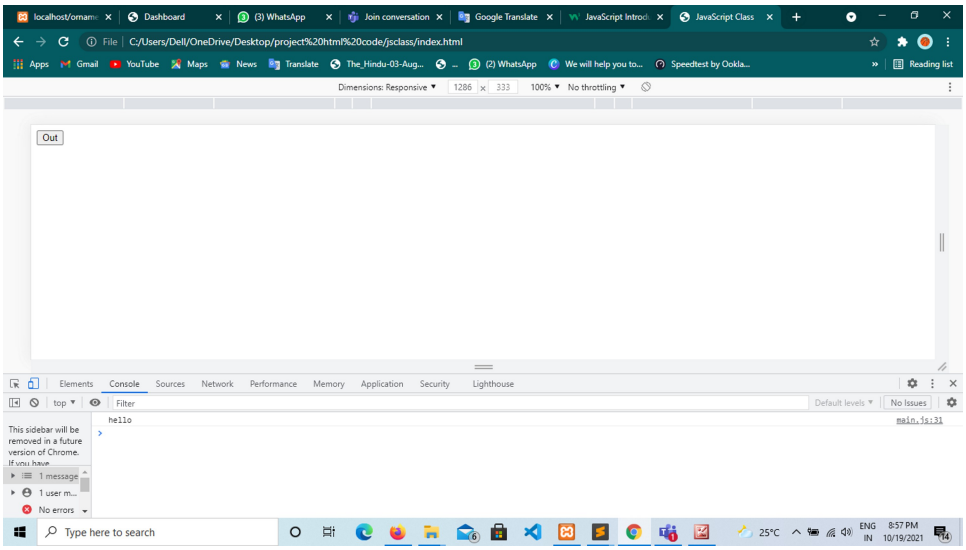
1) **alert('Hello')** :-**On writing in alert function script popup window on browser is Show**
Example:- alert('hello');



1) **Console.log('hello'):- When written in console.log() shows in console of all browsers.**
Example:- console.log('hello);

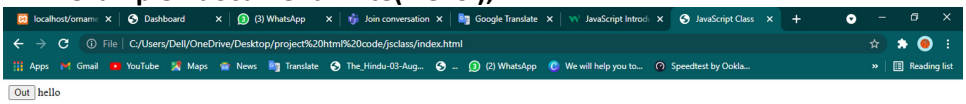


Note: This will show on the right click of the mouse, after that there is an option of bottom console on inspect click, it will show in the same.

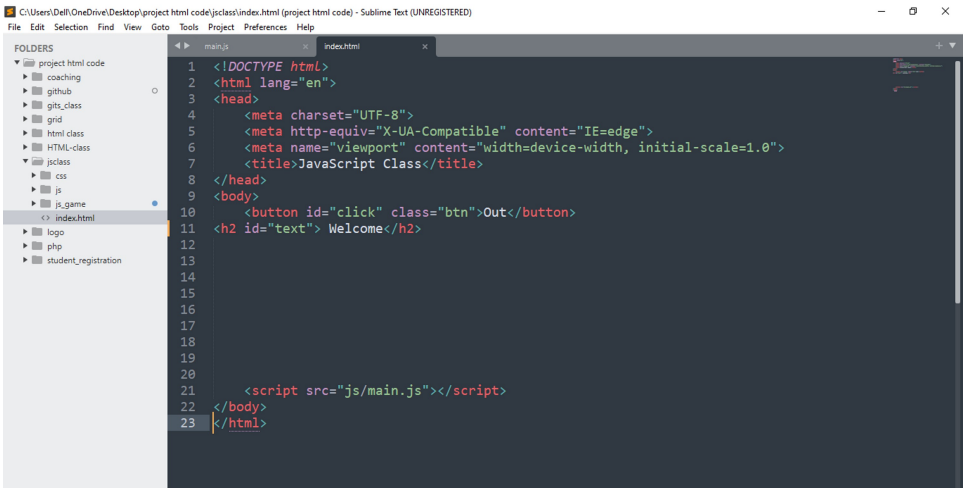


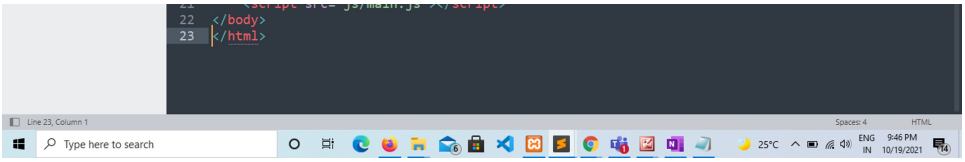
2) **Document.write('hello')**:- Document.write shows on simple browser

example:- document.write('hello');



1) **document.getElementById('IdName')**:- **document.getElementById() has to select the element by Id.**
Example:-document.getElementById('text');





Note:- **To use this function, first we give the id of the script we want to sleep in the html body, this id can be anything.**

How To Use Javascript In HTML

Javascript is used in 2 types of HTML

1 external
2 internal

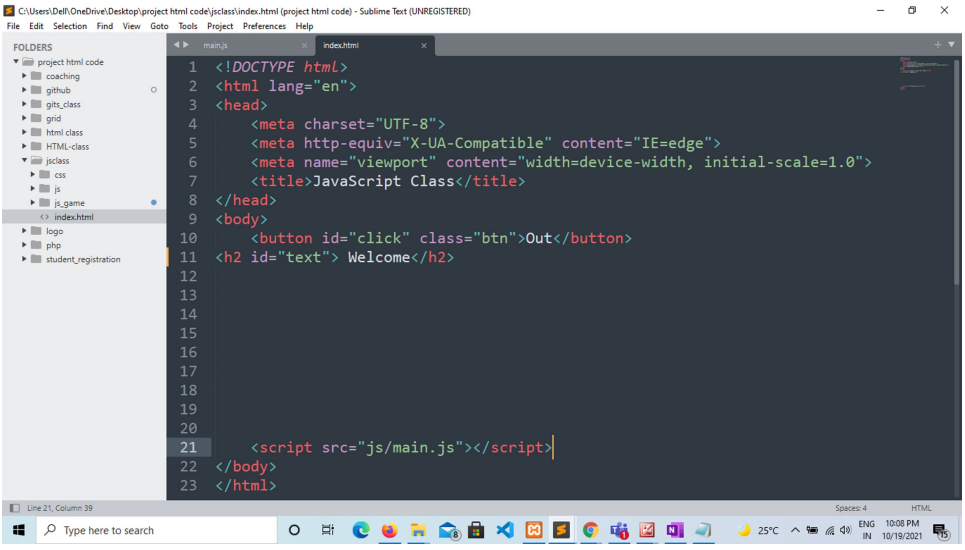
External: To use external JavaScript, we have to create a separate folder and create a file inside it, but the extension of the file (.js) will remain. That is, .js should be in the last of the file name.

Example:- main.js

Note: -

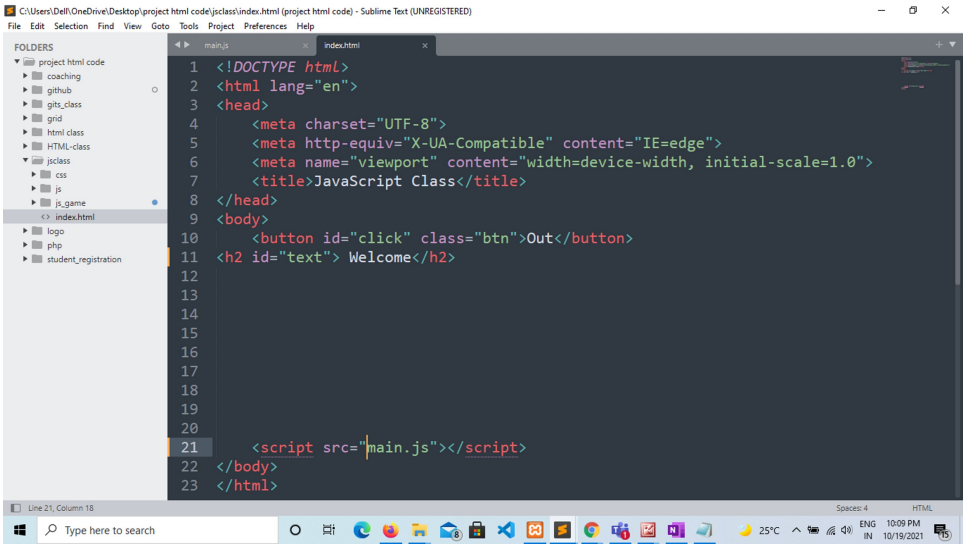
For this in html, we have to add it in the body tag, to add it,

<script src="js/main.js"></script> Function to add this folder together

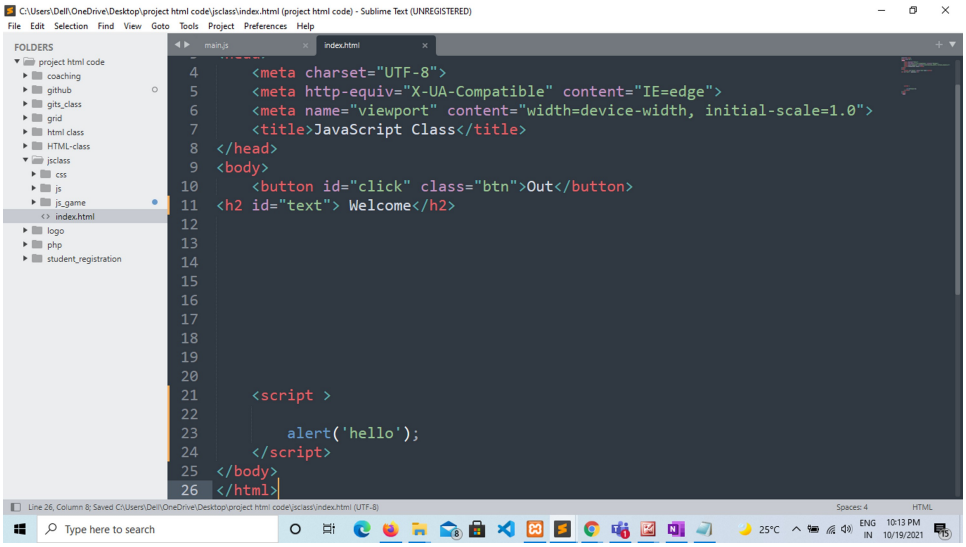


And

<script src="main.js"></script> And this is the script to add files without folders

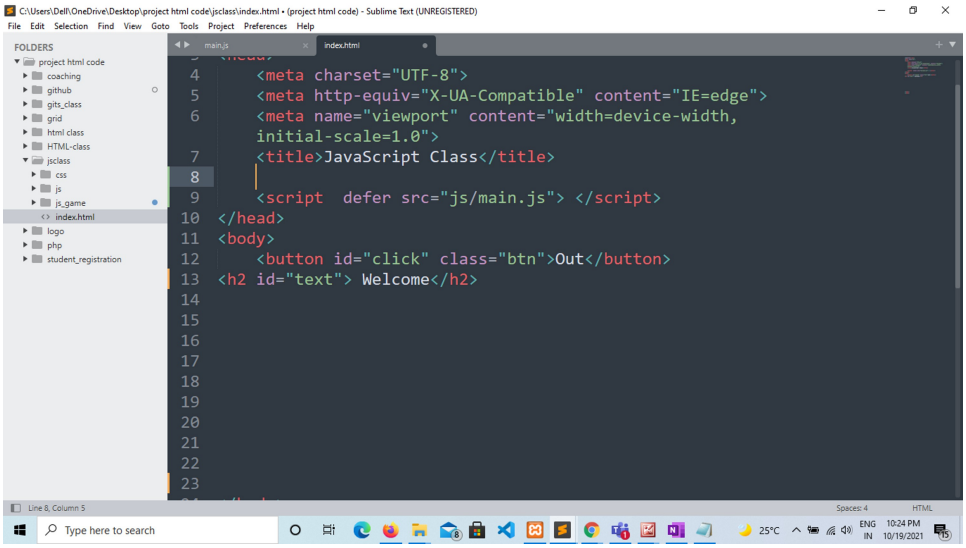


2)internal:-To do it in internal JavaScript, you can simply write your code in the script inside the body tag in HTML.



Defer Function In Javascript

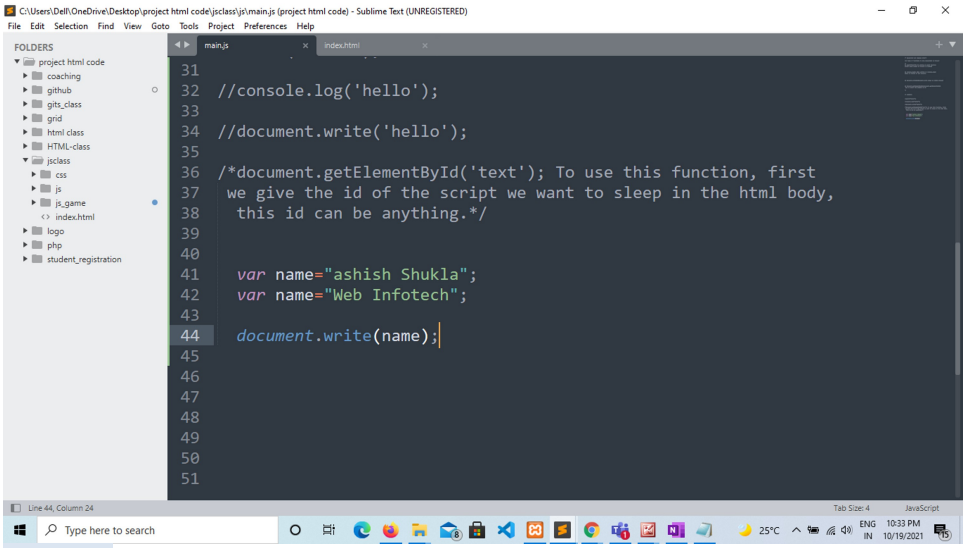
When we write or add JavaScript in the head tag of the defer function, then we use the defer function so that the script written in the bottom of the body tag can run smoothly.



Variable In Javascript

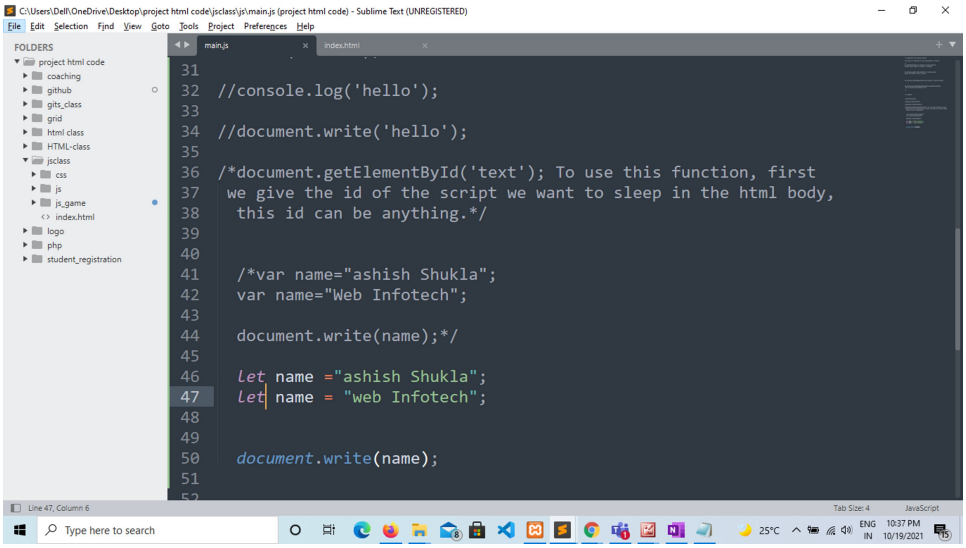
Variable types of functions in JavaScript are created by doing
1) var
2) let
3) const

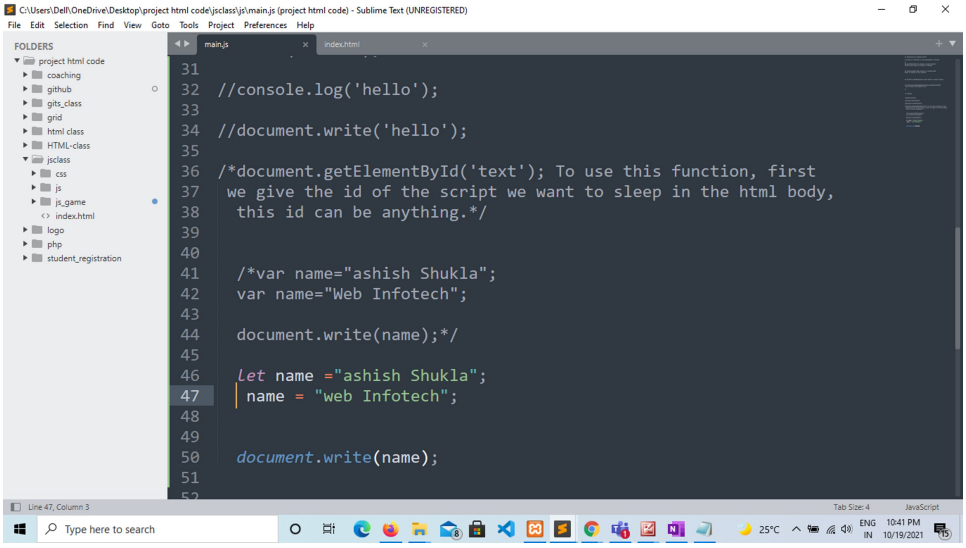
Var:- If you create a variable using the var function, then you can create another variable with that name.



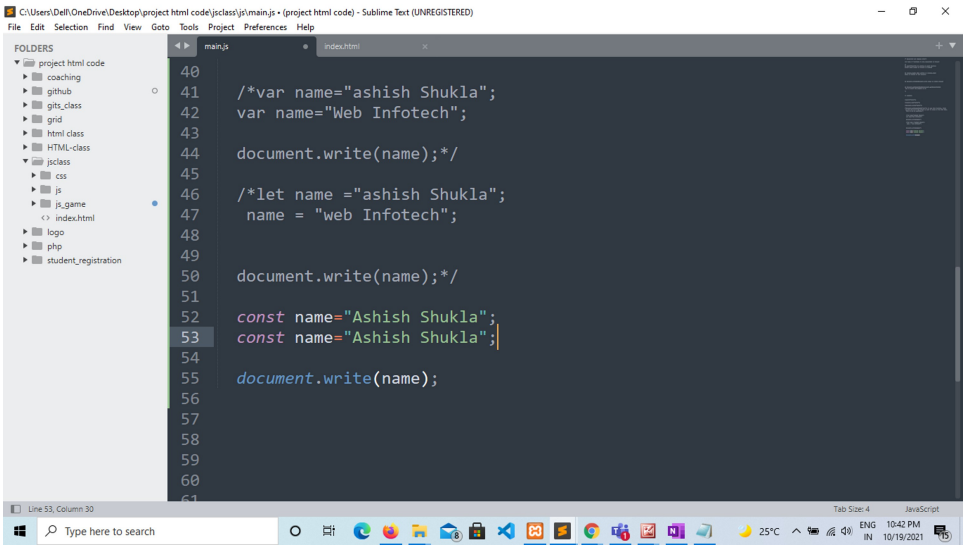
Note: In this, whenever you create another variable with the same name, the second show will be done.

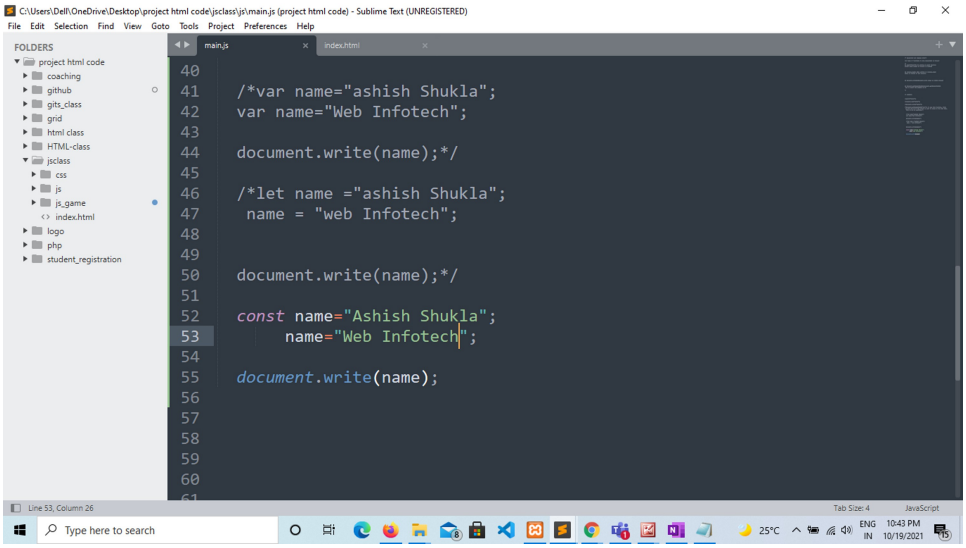
Let:- In this, creating a variable of that name once, cannot create a second variable of that name, but can be created without using let.





Const:-In this, creating a variable of that name once, cannot create a second variable of that name.





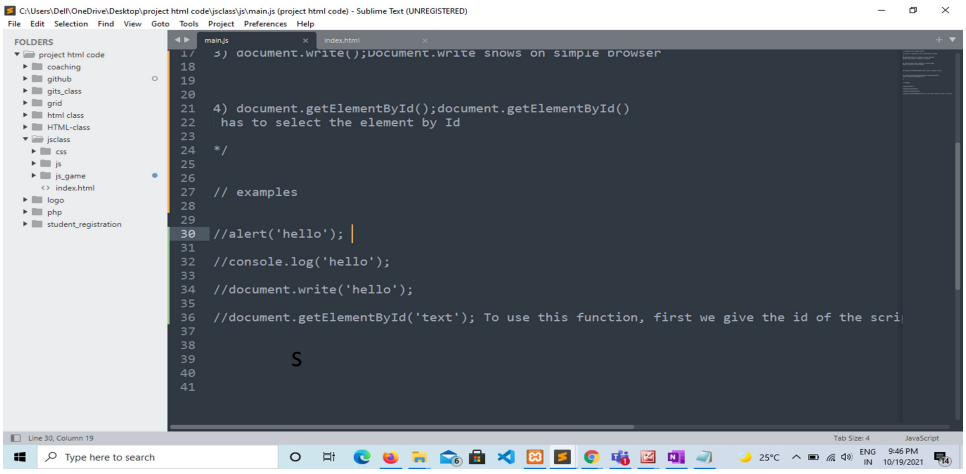
Variable Rule

1) **_ ,$,a-z , A-Z**
   Example: $name, _name, Firstname , firstName.

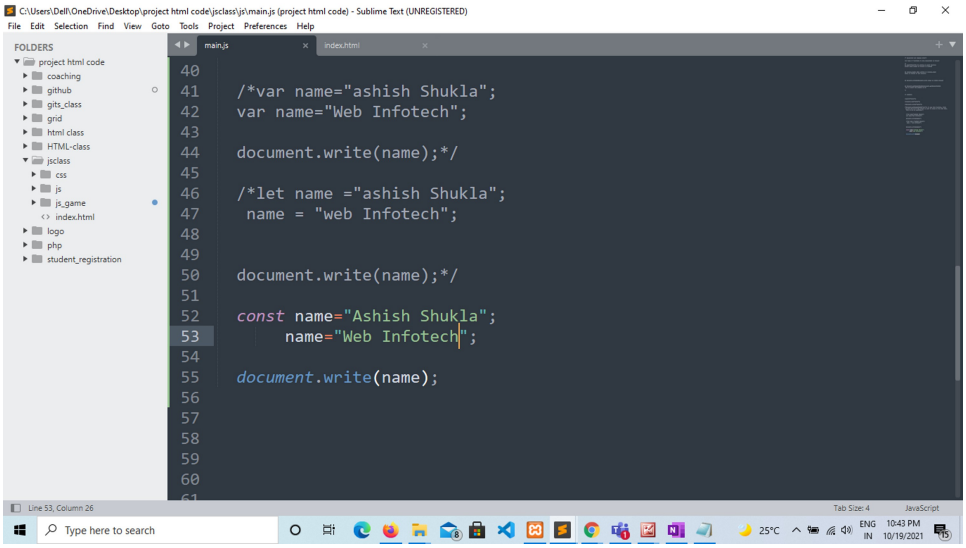
2) **End $, a-z, A-Z, 0-9**
   example: name$,name , Name firstName1.

Comment In javascript

There are two types of comments
1 single line comment //Hello Welcome



2 multiline comments



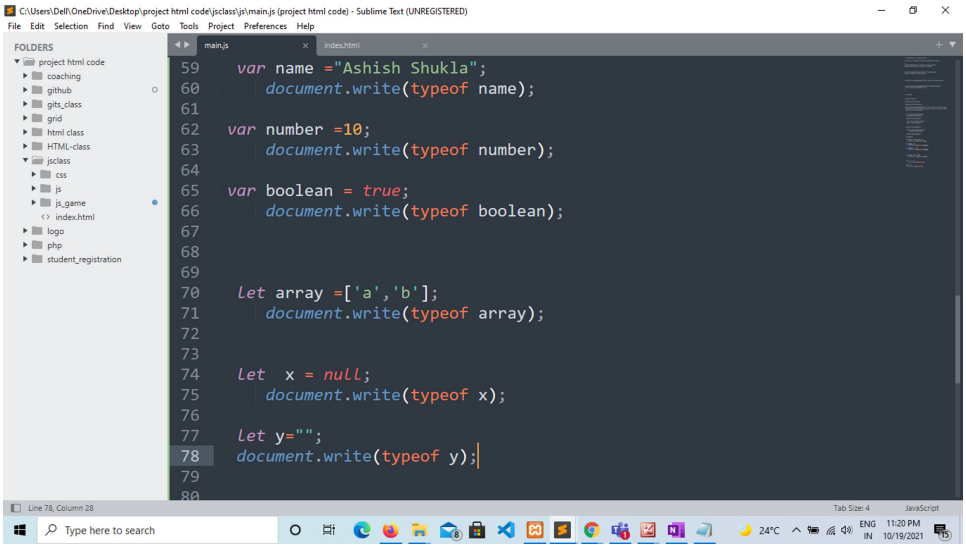

```
/* Hello
 welcome
*/
```

DATATYPE IN JAVASCRIPT

There are 7 types of datatypes in JavaScript

1) **String:- If any b value is in "double " inverted comma then that string datatype is example "name" "10";**
2) **Number:- Number can be any number in datatype 10, 10.10;**
3) **Boolean :- Will show true or false in boolean true/false**
4) **Array:- Array datatype shows object on printing ['a', 'b'];**
5) **Object:-The object will show the object in the datatype ['a':10, y:'20'];**
6) **Null :-NULL Means x = NULL will show its datatype object**
7) **undefine:-Undefined will sleep when the variable is created and when the variable is entered.**

Note: Array, object and null datatype show object datatype in all three



Event In Js

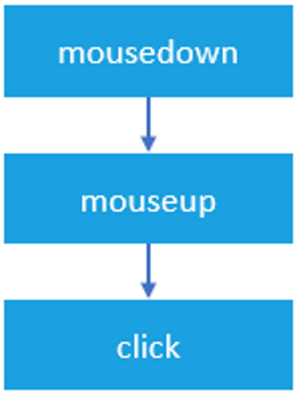
Click -> single click , db click;

Museover
Museown
mouseUp
Mousedown
Mouse wheel
Mouse move

mousedown, mouseup, and click:-

When you click an element, there are no less than three mouse events fire in the following sequence:

1. The **mousedown** fires when you depress the mouse button on the element.
2. The **mouseup** fires when you release the mouse button on the element.
3. The click fires when one **mousedown** and one **mouseup** detected on the element.



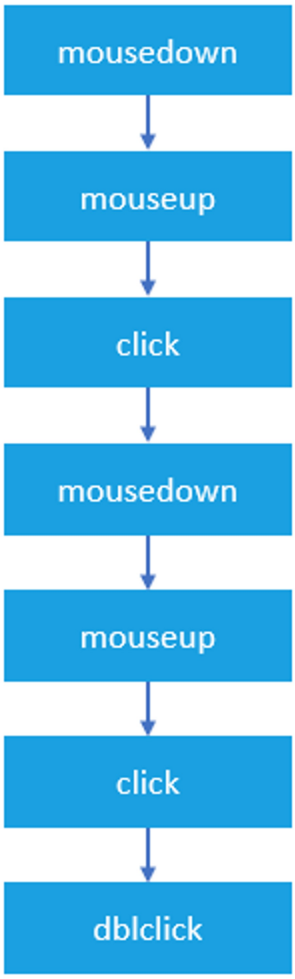
If you depress the mouse button on an element and move your mouse off the element, and then release the mouse button. The only **mousedown** event fires on the element.

Likewise, if you depress the mouse button, and move the mouse over the element, and release the mouse button, the only **mouseup** event fires on the element.
In both cases, the click event never fires.

dblclick

In practice, you rarely use the **dblclick** event. The **dblclick** event fires when you double click over an element.
It takes two click events to cause a **dblclick** event to fire. The **dblclick** event has four events fired in the following order:

1. **mousedown**
2. **mouseup**
3. **click**
4. **mousedown**
5. **mouseup**
6. **click**
7. **dblclick**



As you can see, the click events always take place before the **dblclick** event. If you register both click and dblclick event handlers on the same element, you will not know exactly what user actually has clicked or double-clicked the element.

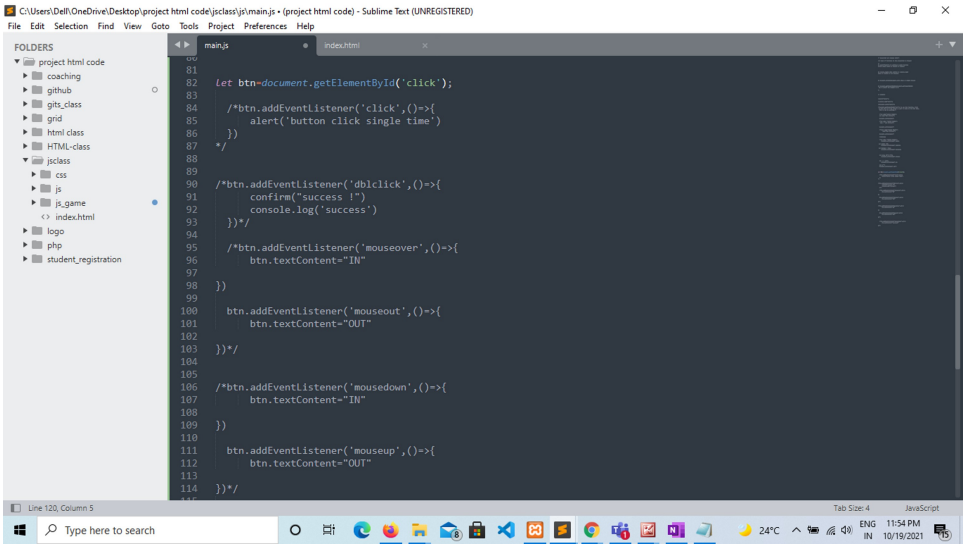
mousemove

The **mousemove** event fires repeatedly when you move the mouse cursor around an

element. Even when you move the mouse one pixel, the **mousemove** event still fires. It will cause the page slow, therefore, you only register **mousemove** event handler only when you need it and immediately remove the event handler as soon as it is no longer used,

mouseover / mouseout

The **mouseover** fires when the mouse cursor is outside of the element and then move to inside the boundaries of the element.
The **mouseout** fires when the mouse cursor is over an element and then move another element.

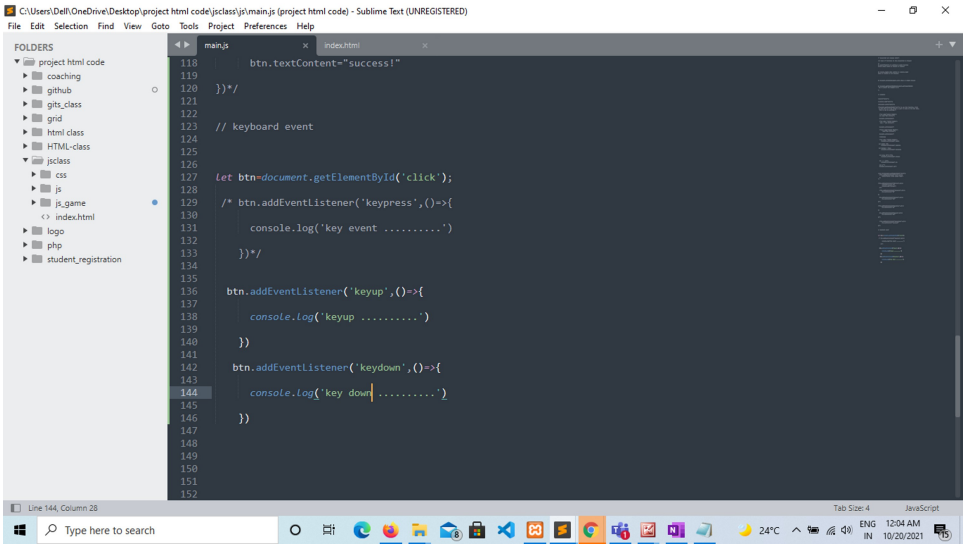


Keyboard Event in Js

1) **Keypress:- This event is fired when an alphabetic, numeric, or punctutaion key is pressed down.**

2) **Keydown:-Keydown happens when key is pressed down,and auto repeats if the key is pressed down for long.**

3) **Keyup:- keyup happens when the key is released.**



Example:-

```
document.getElementById('click');

 document.addEventListener('keypress',()=>{
        console.log('key event .....')
        })

 document.addEventListener('keyup',()=>{
        console.log('keyup .....')
        })

 document.addEventListener('keydown',()=>{
        console.log('key down .....')
        })
```

GET ELEMENT & QUERY SELECTOR

Let btn1= document.getElementById('id'); document.getElementById('id') This selects the id of the tag
Let btn2= document.getElementsByClassName('class');This tag selects the class name
Let btn2=document.getElementsByTagName('button');This selects the tag
Let btn4 =document.querySelector('button);
Let btn5 =document.querySelectorAll('button);This selects the all tag

Operators In javascript

- **Arithmetic Operators**
- **Comparison Operators**
- **Logical Operators**
- **Assignment Operators**
- **Ternary Operators**

Arithmetic Operators:

There are various Arithmetic Operators –

- **+ (Addition):**
  '+' operator performs addition on two operands.
  Example:
  `Y = 5 + 5 gives Y = 10`

  – (Subtraction) :
  '-' operator performs subtraction on two operands.
  Example :
  `Y = 5 - 3 gives Y = 2`

- *** (Multiplication) :**
  '*' operator performs multiplication on two operands.
  Example :
  `Y = 5 * 5 gives Y = 25`

- **/ (Division) :**
  '/' operator performs division on two operands (divide the numerator by the denominator).
  Example :
  `Y = 5 / 5 gives Y = 1`

- **% (Modulus) :**

'%' operator gives remainder of an **integer** division.

- `A % B means remainder (A/B)`
  `Y = 5 % 4 gives Y = 1`
- **+ + (Increment) :**
  '+ +' operator increases an integer value by one.
  Example :
  `let` **`A = 10`** `and` **`Y = A + +`** `then A = 11, Y=10`
  `if` **`A = 10`** `and` **`Y = + + A`** `then A = 11, Y=11`
- **– – (Decrement) :**
  '- -' operator decreases an integer value by one.
  Example :
  `let` **`A = 10`** `and` **`Y = A - -`** `then A = 9, Y=10`
  `if` **`A = 10`** `and` **`Y = - - A`** `then A = 9, Y=9`
  Assignment Operators:

  There are various Assignment Operators in JavaScript –

- **= (Assignment Operator) :**
  Assigns right operand value to left operand.
  Example :
  `If A = 10 and Y = A then Y = 10`
- **+= (Add and Assignment Operator) :**
  Sums up left and right operand values and then assign the result to the left operand.
  Example :
  `Y += 1 gives Y = Y + 1`
- **– = (Subtract and Assignment Operator) :**
  It subtract right side value from left side value and then assign the result to the left operand.
  Example :
  `Y -= 1 gives Y = Y - 1`
- similarly there are ***= (Multiply and Assignment)**, **/= (Divide and Assignment)**, **%= (Modules and Assignment)**
  Example:
  `Y *= A is equivalent to Y = Y * A`
  `Y /= A is equivalent to Y = Y / A`
  `Y %= A is equivalent to Y = Y % A`
  Comparison Operators:

  There are various Comparison Operators in JavaScript –

- **= = :**
  Compares the equality of two operands. If equal then the condition is true otherwise false.
  Example :
  `Y = 5 and X = 6`
  `Y = = X is false.`

Note : Type not considered in '= =' operator.

- **= = = :**
  this operator compares equality of two operands with type.If equal(type and value both) then condition is true otherwise false.
  Example :

  ```
  X = 10 then X = = = "10" is false.
  X = = = 10 is true.
  ```

- **!= (Not Equal):**
  Compares inequality of two operands. True if operands are not equal.
  Example :
  `given X = 10 then X ! = 11 is true.`
- **> (Greater than):**
  this operator checks whether the left side value is greater than the right side value. If yes then it returns true otherwise it returns false.
  Example :
  `given X = 10 then X > 11 is false.`
- **< (Less than):**
  this operator checks whether the left side value is less than right side value. If yes then it returns true otherwise it returns false.
  Example :
  `given X = 10 then X < 11 is true.`
- **> = (Greater than or Equal to):**
  this operator checks whether the left side operand is greater than or equal to the right side operand. If yes then it returns true otherwise it returns false.
  Example :
  `given X = 10 then X > = 11 is false.`
- **<= (Less than or Equal to):**
  this operator checks whether the left side operand value is less than or equal to the right side operand value. If yes then it returns true otherwise it returns false.
  Example :
  `given X = 10 then X < = 10 is true.`
  Logical Operators:

  There are various Logical Operators in JavaScript –

- **&& (Logical AND):**
  It checks whether two operands are non-zero (0, false, undefined, null or "" are considered as zero), if yes then return 1 otherwise 0.
  Example :
  ```
  Y = 5 and X = 6
  Y && X is true.
  ```
- **|| (Logical OR) :**
  It checks whether any one of the two operands is non-zero (0, false, undefined, null, or "" is considered as zero). Thus || returns true if either operand is true and if both are false it returns false.
  Example :

```
Y = 5 and X = 0
Y || X is true.
```

- **! (Logical NOT):**
  It reverses the boolean result of the operand (or condition).
  Example :
  ```
  Y = 5 and X = 0
  !(Y || X) is false.
  ```
  Ternary Operator:

- **: ? Operator :**
  It is like the short form of the if-else condition.
  Syntax:
  ```
  Y =  ? A : B
  ```
  where A and B are values and if condition is true thhen Y = A otherwise Y = B.
  Example:
  ```
  Y = (6>5) ? 6 : 5
  therefore Y = 6
  ```
  typeof Operator: It returns the type of a variable.

  Syntax:

  ```
  typeof variable;
  ```
  Example: