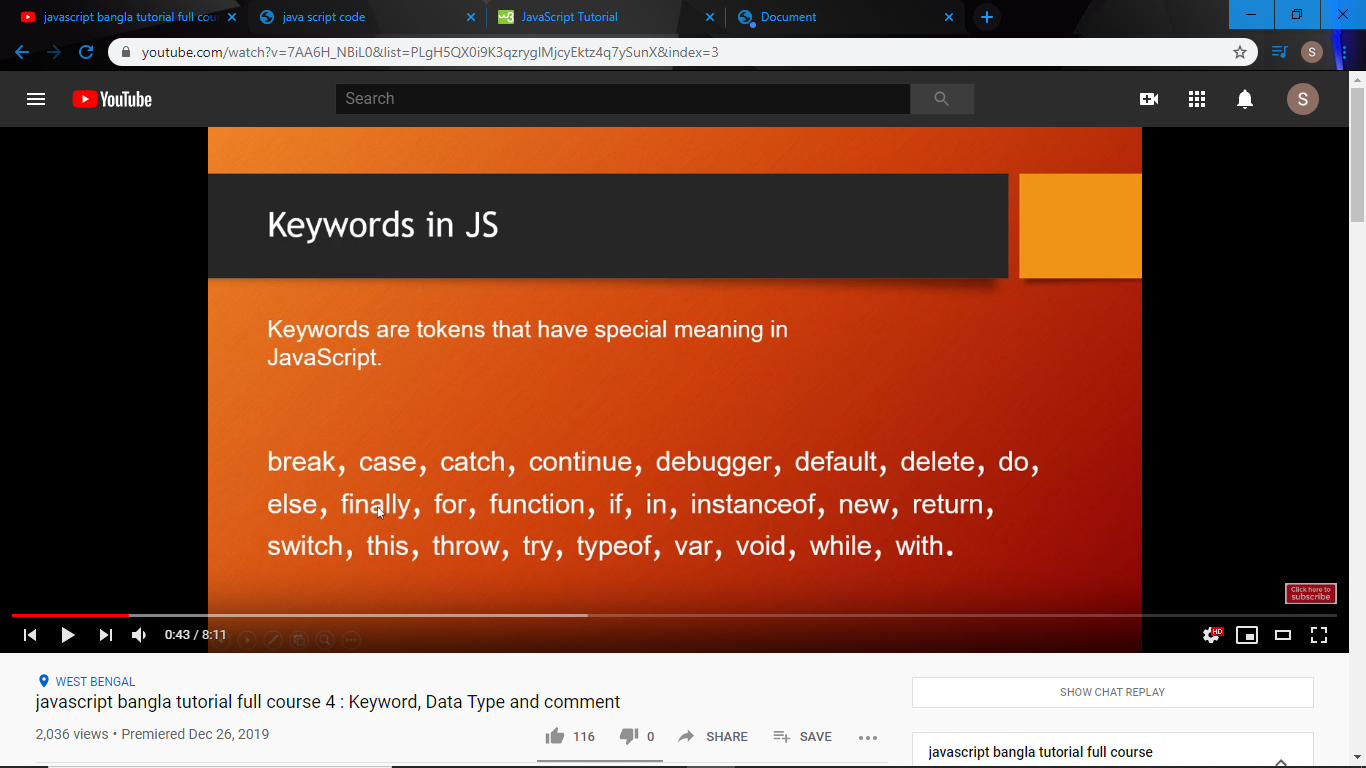
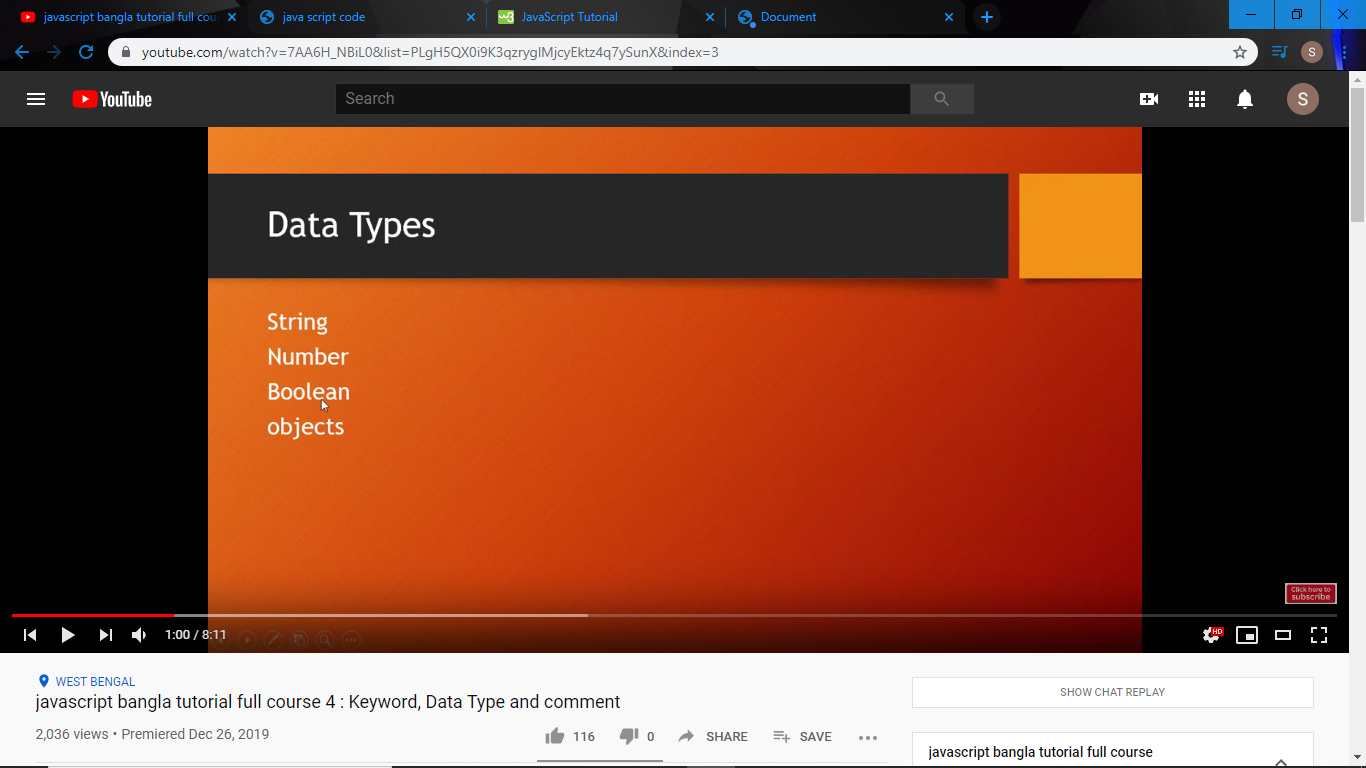
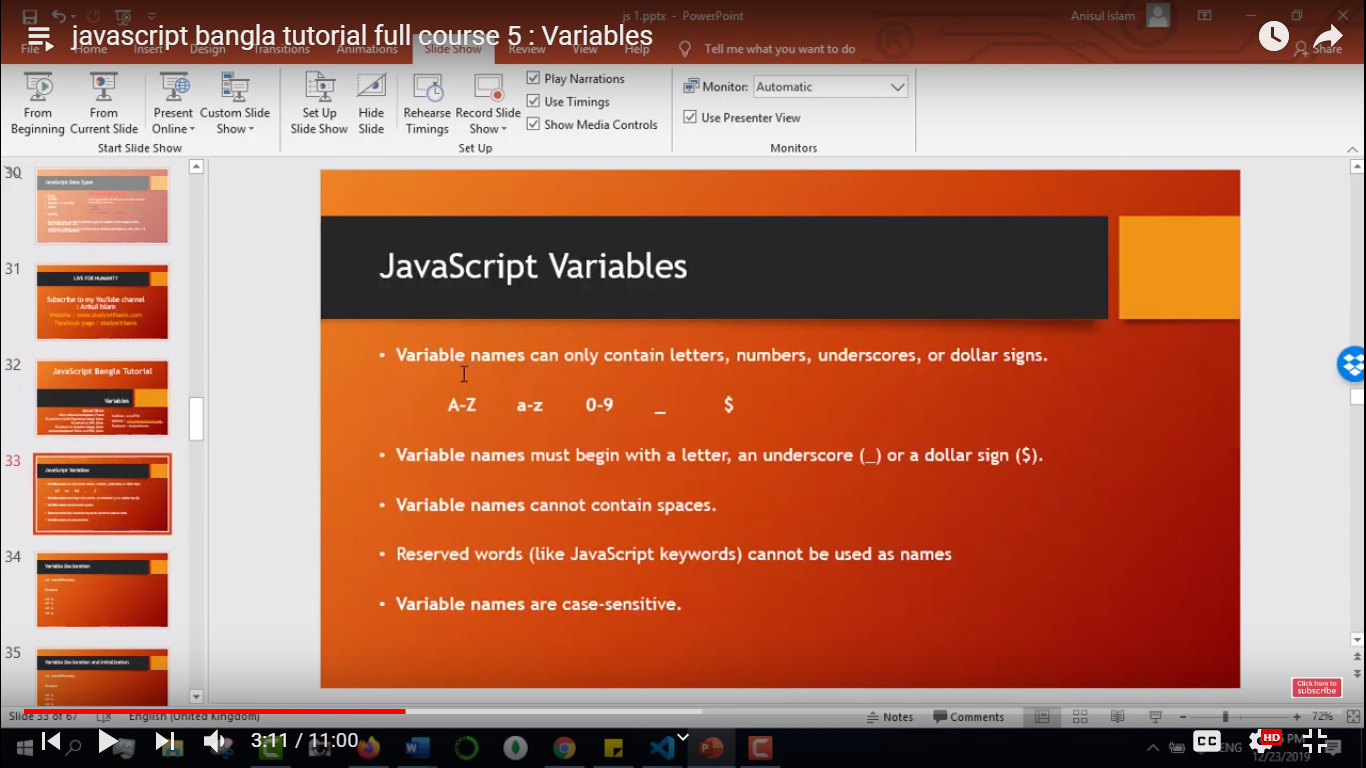
About ES6 Book: <https://exploringjs.com/es6/ch_destructuring.html#sec_overview-destructuring>





Undefine,null is also datatype;



\*\*We can declare variable as a var, let, const\*\*

Variable=We can called variable is a container. It can contain data; We can not use Reserve word as a variable.

Var a=b=c=3; Means a,b and c value is 3;

Var name=”shanto”; console.log(typeof(name));//it return data type// output: String

Var= is a functional scoped.

let= is a Block scoped;

// var is functional scope

var x = 10;

if(true) {

    var x = 20;

    console.log(x);

}

console.log(x); // x er man aser kotha silo 10 . ans hobe 20. x 20 dara replace hoise.this is the problem

// let is blocked scope

let y = 20;

if(true){

    let y = 10;

    console.log(y);// blocked scope // output: 10

}

console.log(y); // output: 20

let firstName=”shanto”; //Lower camel case… console.log(“My name is shanto”, firstName);

let Firstname; //Upper camel case…

let first\_name; //Underscore..

let FirstName; //pascal case…

const name=”shanto”; // after, you can’t change variable value;

name=”islam”; // compiler must be show error.

var x;

var y=4;

x=y++; // here at first x=y ; then y=y+1;

console.log(x,y); output : x=4,y=5;

x=++y; // here at first y=y+1; then x=y;

console.log(x,y); output : x=5,y=5;

var i=0;

while(i++<5){

    document.write("<br/>",i,"<br/>");//12345

}

i=0;

while(++i<5){

    document.write(i,"<br/>");//1234

}

8==n && n==8 are same;

Var name = true; if(name){} this is enough.

var ternary;

var num=3

ternary=(3==num) ? true : false;// ? means if, : means else.? is ternary operator

document.write(ternary);

Array:

Push(); added value in the array as a last index.

Unshift(); added value in the array as a first index.

Pop(); remove value in the array as a last index.

Shift(); remove value in the array as a first index.

Delete(); example: var week=["Sat","Sun","Mun","Twe","Wed","Thu","Fri"]; delete(week(2)); // delete mun from week array

Week.slice(2,4); // output will be: [Mun Twe Wed] // var let=week.slice(). Array copy hobe week to let a

Week.slice(-3); // output will be: ["Wed","Thu","Fri"]

Var s=”hi I am shanto islam”;

s.split(“ “);// it splited by the “ “;

s.split(““);// it splited by the ““;

var len=s.split(“a”); console.log(len.length);// it count total number of a.

Function:

Function shanto(name = “default”, age = “unknown”){

}

Shanto(“hi”,23);

Shanto();// here condiser default and unknown.

function reverseString(name){

    var result=name.split('').reverse().join('');

    return result;

}

var str="I love bangladesh";

console.log(reverseString(str));

Here we find text from button….

for(var i=0;i<3;i++){

    document.querySelectorAll(".Button")[i].addEventListener("click",function(){

    var text=this.innerHTML;

    // console.log(text);

    // document.write(text);

    });

}

KeyPress Listener:

// document.addEventListener("keypress",function(event){

//     var get=event.key;

//     document.querySelector("b").innerHTML="You press "+get;// For b tag

// });

var count=0;

document.querySelector("textarea").addEventListener("keypress",function(event){

    count++;

    var get=event.key;

    document.querySelector("b").innerHTML="You press "+count;// For textarea tag

});

From You don’t know js book:

the log( b ) part is referred to as a function call

the console. part is an object reference where the log(..)  
function is located.

you declare a variable (container)  
to hold a specific type of value,

Functions are often used for code that you plan to call multiple  
times,

The following built-in types are available:  
• string  
• number  
• boolean  
• null and undefined  
• object  
• symbol (new to ES6);

typeof operator;

typeof null is an interesting case because it errantly returns  
"object" when you’d expect it to return "null"

**Objects**The object type refers to a compound value where you can set  
properties (named locations) that each hold their own values of any  
type.

Properties can either be accessed with *dot notation* (i.e., obj.a) or  
*bracket notation* (i.e., obj["a"])

**Arrays**An array is an object that holds values (of any type)

functions are a subtype of objects;

=== is often called “strict equality” and == loose equality

The difference between == and === is usually characterized that ==  
checks for value equality and === checks for both value and type  
equality.

**var** a = "42";  
**var** b = 42;  
a == b; *// true*a === b; *// false*

“conditional opera‐tor,” often called the “ternary operator.”

**var** b = (a > 41) ? "hello" : "world";

**var** foo = **function**() {  
*// ..*};  
The first function expression assigned to the foo variable is called

*anonymous* because it has no name.

If a function has a this reference inside it, that this reference usually points to an object. It’s important to realize that this *does not* refer to the function itself,

as is the most common misconception.

var foo = {

a: 42

};

*// create `bar` object and link it to `foo`*

var bar = Object.create( foo );

bar.b = "hello world";

console.log(bar);

**Types & Grammar**