```
// var allows to redeclare variables
var x = 10;
console.log(x);
var x = 20;
console.log(x);

// let does not allow to redeclare variables
let a = 10;
console.log(a);
let a = 20;
console.log(a);

// var is function scoped (any function)
// var is scoped to the nearest function block
// A variable declared with var is defined
// throughout the program as compared to let

function something(){
    var a = 20;
    console.log(a);
    if(true){
        var a = 30;
        console.log(a);
    }
    console.log(a);
    var a = 25;
    console.log(a);
}
something();
```

```
// let is block scoped (if, for,..)
// let is scoped to the nearest enclosing block
function somefunc(){
    let age=27;
    console.log(age);
    if(true){
        let age=30;
        console.log(age);
    }
    console.log(age);
    age=29;
    console.log(age);
}
somefunc();

function func(){
    for(var i=0;i<10;i++)
        console.log(i);
    console.log(i);
}
func();

function timer(){
    for(var i=0;i<5;i++){
        setTimeout(function(){
            console.log(i);
        }),1000
    }
}
timer();
```

```
function timer(){
    for(let i=0;i<5;i++){
        setTimeout(function(){
            console.log(i);
        }),1000
    }
}
timer();

// Hoisting occurs in var
x = 10;
console.log(x);
var x;

// Hoisting does not occur in let
age=27;
console.log(age);
let age;

function dosmth(){
    age=27;
}
let age;
dosmth();
console.log(age);

// const variables are immutable Error!
const x;
const x =10;
x = 20;
```

```
const person = {
    name : "xyz",
    age : 20
};
// if we try to update something in the const object by typing
person ={
    name : "abc",
    age : 30
};
// it won't allow us to do that
// However, the properties of a const variable can change.
// That's because the entire object is not immutable.
// It just can't be reassigned entirely
console.log(person.name);
person.age = 30;
console.log(person.age);

const arr = [26,30,27];
console.log(arr);
arr.push(25);
console.log(arr);

// var vs let vs const
// var
function blocky(){
    if(true){
        var something = "something";
        console.log(something);
    }
    console.log(something);
```

```
}
blocky();

// let
function blocky(){
    if(true){
        let something = "something";
        console.log(something);
    }
    console.log(something);
}
blocky();

function blocky(){
    let something = "something";
    console.log(something);
    if(true){
        let something = "something2";
        console.log(something);
    }
    console.log(something);
}
blocky();

// const variables are immutable (you can't change them)
function blocky(){
    const something = "something";
    console.log(something);
    if(true){
        const something = "something2";
```

```
        console.log(something);
    }
    console.log(something);
}
blocky();

// let vs const
function blocky(){
    let something = "something";
    console.log(something);
    if(true){
        let something = "something2";
        console.log(something);
    }
    something = "somethingnew";
    console.log(something);
}
blocky();

function blocky(){
    const something = "something";
    console.log(something);
    if(true){
        const something = "something2";
        console.log(something);
    }
    something = "somethingnew";
    console.log(something);
}
blocky();
```