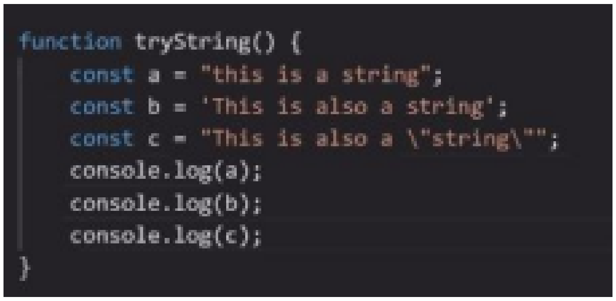
Day 2

# Javascript

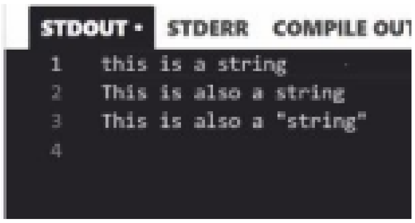
Solving yesterday’s in-class pattern question.

- As console.log(“print something”) will always print in new line therefore you have to create a single line first.
- In javascript you can use double quote or single quote to make a string but remember in java single quotes were used to create a char type value and double quotes were used to create a string value. ● But if you are using double or single quote to make a string how to print single or double quote? this can be done using escape character (\) backslash. In java you were using (\\) double backslash but in javascript you just need to use (\) single backslash.

Example:

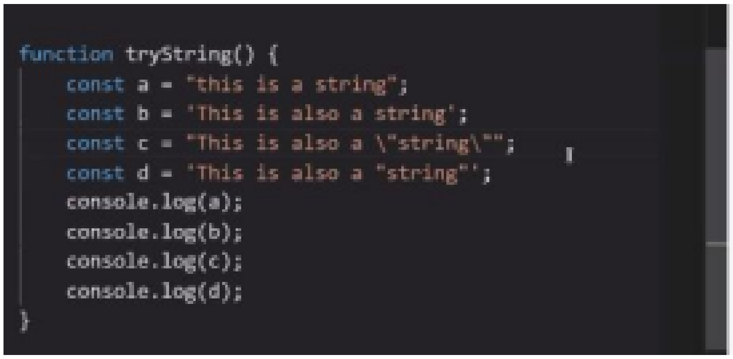


```
function tryString() {
    const a = "this is a string";
    const b = 'This is also a string';
    const c = "This is also a \"string\"";
    console.log(a);
    console.log(b);
    console.log(c);
}
```

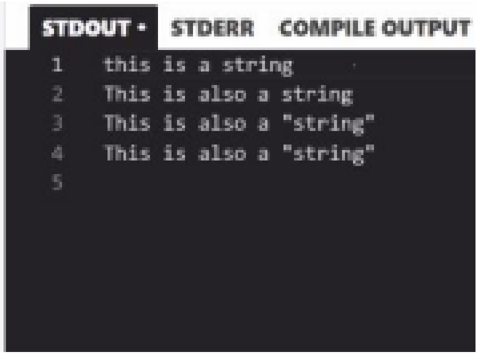


```
this is a string
This is also a string
This is also a "string"
```

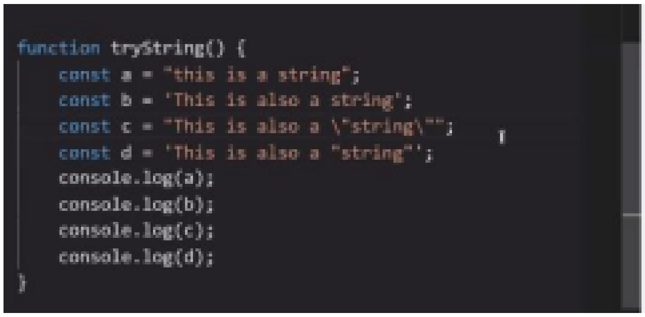
- To print single quote you can also use double quotes and to print double quote you can also use single quotes.



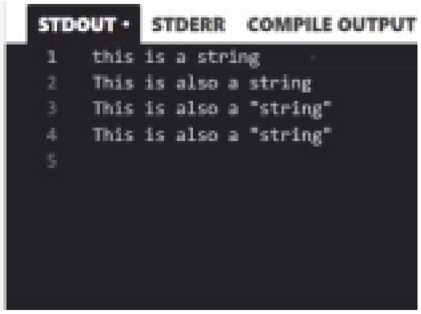
```
function tryString() {
    const a = "this is a string";
    const b = 'This is also a string';
    const c = "This is also a \"string\"";
    const d = 'This is also a "string"';
    console.log(a);
    console.log(b);
    console.log(c);
    console.log(d);
}
```



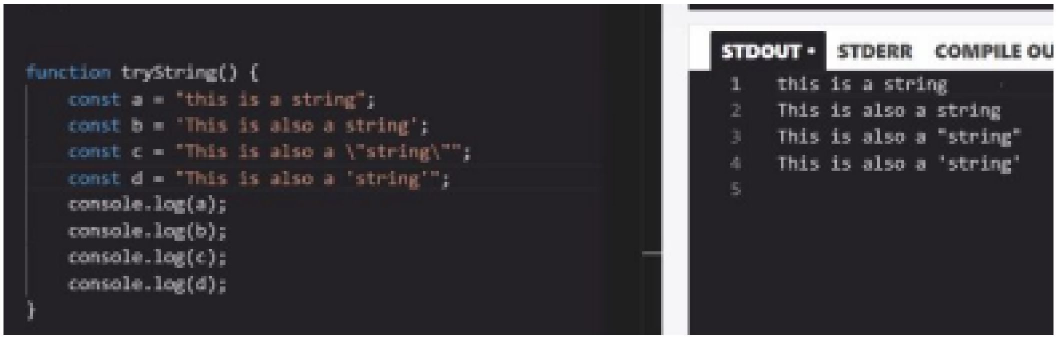
```
this is a string
This is also a string
This is also a "string"
This is also a "string"
```



```
function tryString() {
    const a = "this is a string";
    const b = 'This is also a string';
    const c = "This is also a \"string\"";
    const d = 'This is also a "string"';
    console.log(a);
    console.log(b);
    console.log(c);
    console.log(d);
}
```

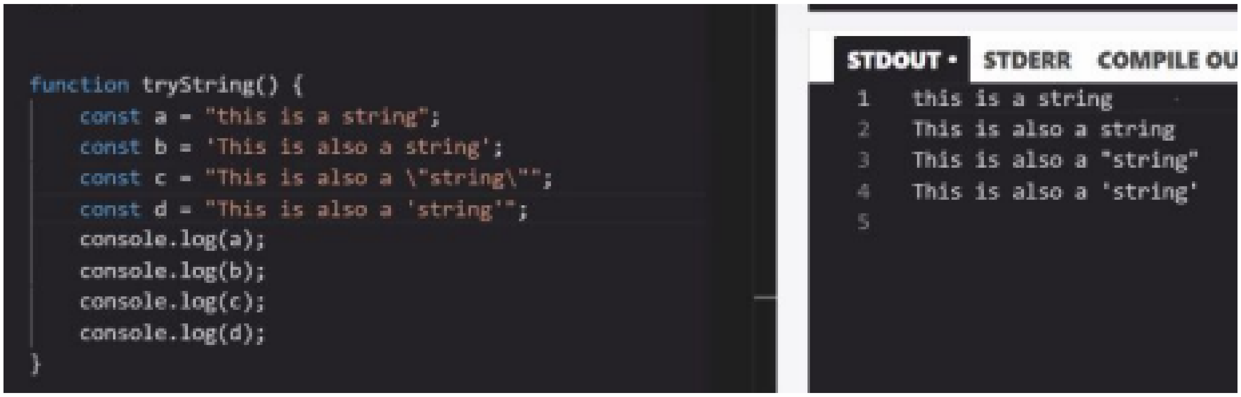


```
this is a string
This is also a string
This is also a "string"
This is also a "string"
```



```
function tryString() {
    const a = "this is a string";
    const b = 'This is also a string';
    const c = "This is also a \"string\"";
    const d = "This is also a 'string'";
    console.log(a);
    console.log(b);
    console.log(c);
    console.log(d);
}
```

```
this is a string
This is also a string
This is also a "string"
This is also a 'string'
```

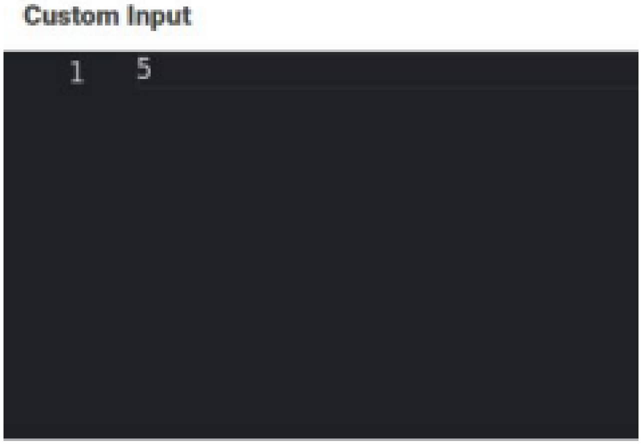


```
function tryString() {
    const a = "this is a string";
    const b = 'This is also a string';
    const c = "This is also a \"string\"";
    const d = "This is also a 'string'";
    console.log(a);
    console.log(b);
    console.log(c);
    console.log(d);
}
```

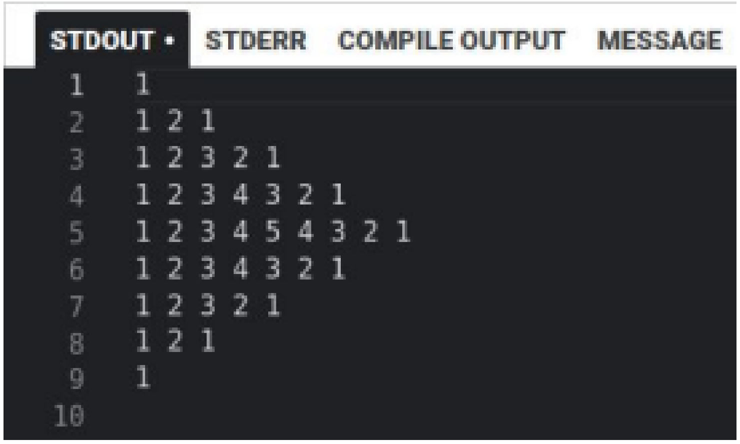
```
this is a string
This is also a string
This is also a "string"
This is also a 'string'
```

Solution to yesterday's in class pattern question

```
// Print the pattern as mentioned in Output Format
// N: contains the input
// you need to print the pattern in the below function
function patternMaking(N)
 {
 // Your code here
 for(let j=1;j<=N;j++){
 let line="";
 for(let i=1;i<=j;i++){
 line=line + i + " ";
 }
 for(let i=j-1;i>=1;i--){
 line=line + i + " ";
 }
 console.log(line);
 }
 for(let j=N-1;j>=1;j--){
 let line ="";
 for(let i=1;i<=j;i++){
 line =line+i+" ";
 }
 for(let i=j-1;i>=1;i--){
 line =line+i+" ";
 }
 console.log(line);
}
}
```

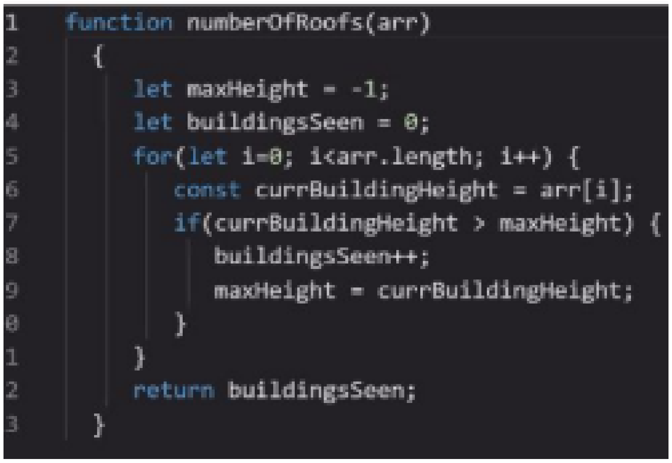


```
5
```



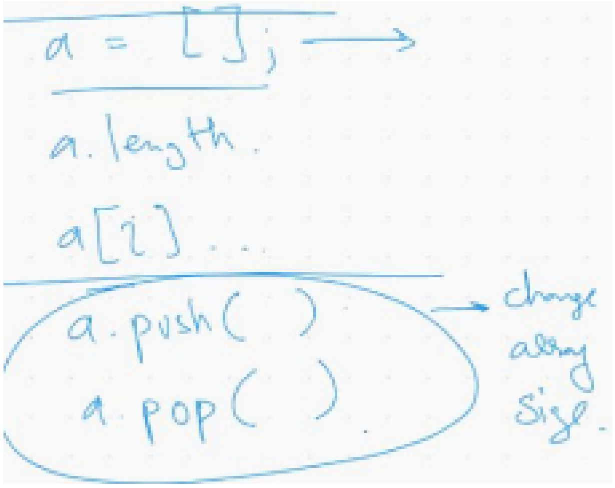
```
1
1 2 1
1 2 3 2 1
1 2 3 4 3 2 1
1 2 3 4 5 4 3 2 1
1 2 3 4 3 2 1
1 2 3 2 1
1 2 1
1
```

Yesterday's in-class Buildings question



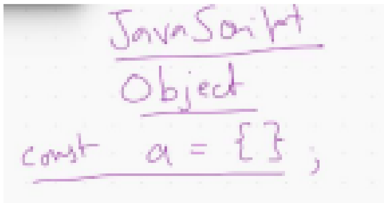
```javascript
function numberOfRoofs(arr)
 {
   let maxHeight = -1;
   let buildingsSeen = 0;
   for(let i=0; i<arr.length; i++) {
     const currBuildingHeight = arr[i];
     if(currBuildingHeight > maxHeight) {
       buildingsSeen++;
       maxHeight = currBuildingHeight;
     }
   }
   return buildingsSeen;
 }
```

Arrays in javascript



In javascript array is like an arraylist in java. First line 'a=[];' is the syntax to declare an array in javascript. 'a.length' will give you length of array. To access an array its the syntax in 3rd line 'a[i];'. 'a.push(x)' adds the element at the end of the array. 'a.pop' will remove the element form the array.

Object in Javascript.



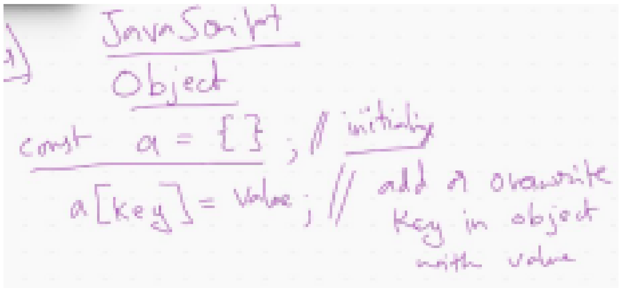
JavaScript
Object

```
const a = { } ;
```

Initialization of the javascript object.

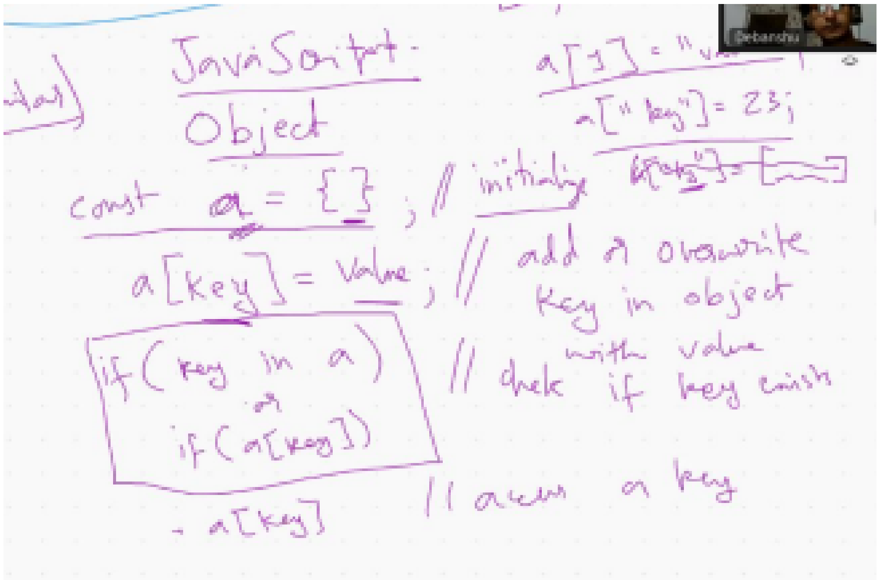
How to access and add values in this object.

a[key]=value;



JavaScript
Object

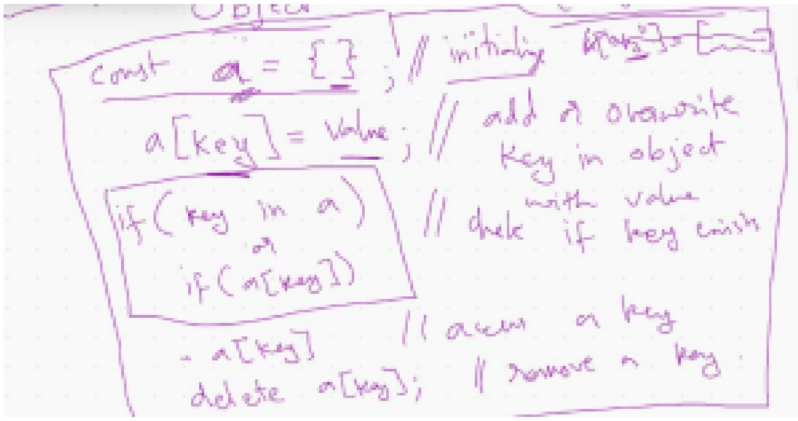
```
const a = { } ; // initialize
a[key] = value ; // add or overwrite
                    key in object
                    with value
```



JavaScript
Object

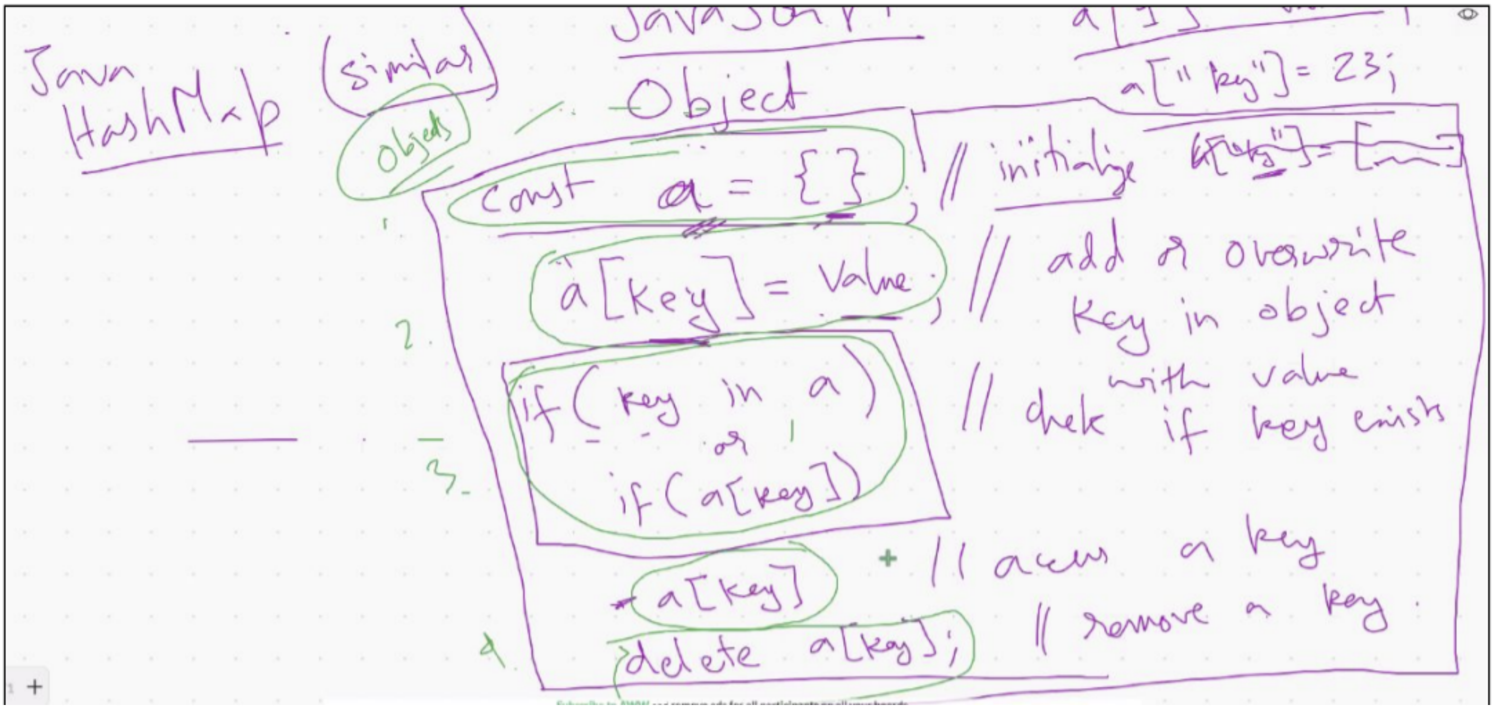
```
a[1] = "va
a["key"] = 23;
a["key"] = [      ]

const a = { } ; // initialize
a[key] = value ; // add or overwrite
                    key in object
                    with value
if ( key in a )   // chek if key exists
   or
if ( a[key] )
- a[key]          // access a key
```

Access a key
‘a[key]’



```
const a = { } ; // initialize     a["key"] = [      ]
a[key] = value ; // add or overwrite
                    key in object
                    with value
if ( key in a )   // chek if key exists
   or
if ( a[key] )
- a[key]          // access a key
delete a[key];    // remove a key
```

This is how you use hash map in javascript



Java HashMap (similar) Objects — Object

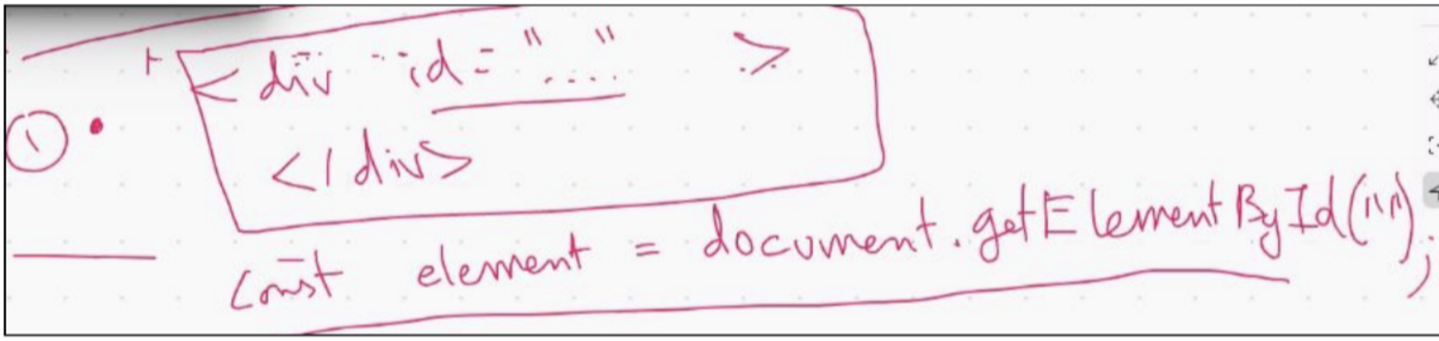
JavaScript

```
a[ ]
a["key"] = 23;

const a = {};          // initialize   a["key"] = [ ]
2. a[key] = value;     // add or overwrite key in object with value
3. if( key in a )      // check if key exists
   or
   if(a[key])
   a[key]              // access a key
4 delete a[key];       // remove a key
```

Topic: html+javascript

1st

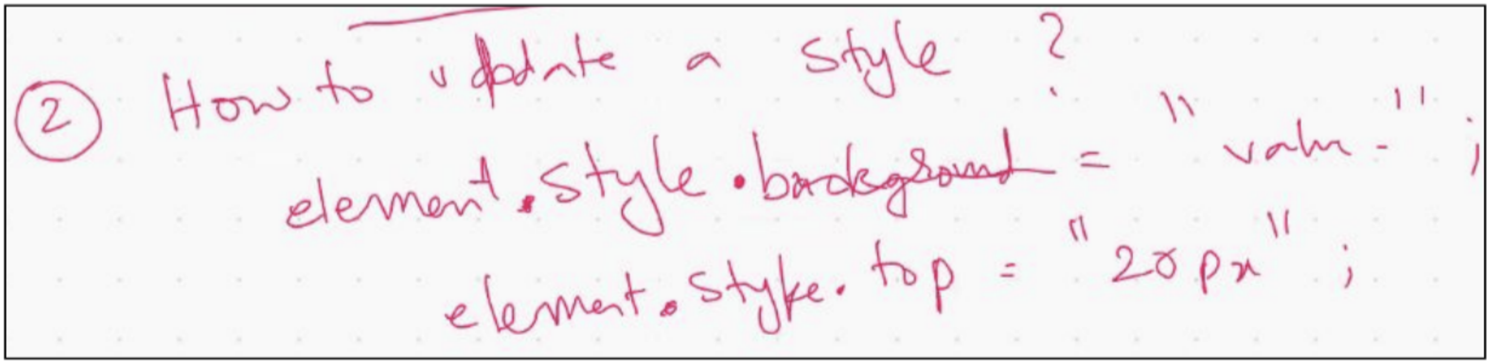
Html div with id and How to get this div in javascript?



```
① <div id="..." >
   </div>
const element = document.getElementById("");
```

2nd

How to update a style using javascript



② How to update a style?

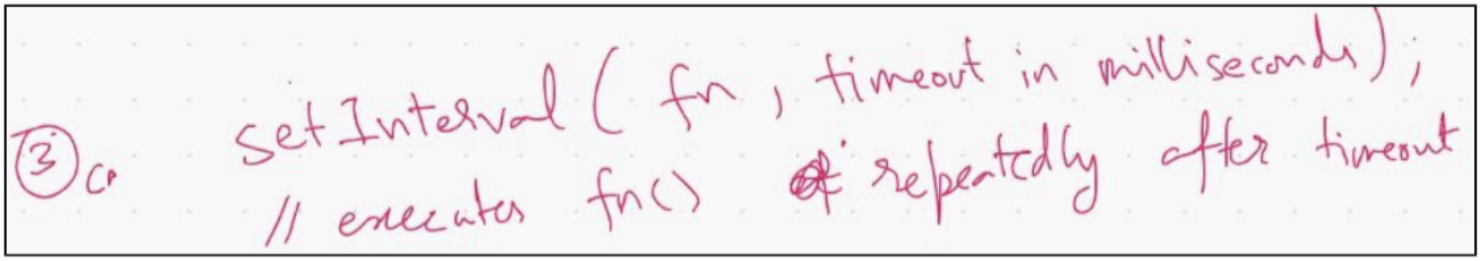
```
element.style.background = "value";
element.style.top = "20px";
```

element.style.background=”value”

3rd

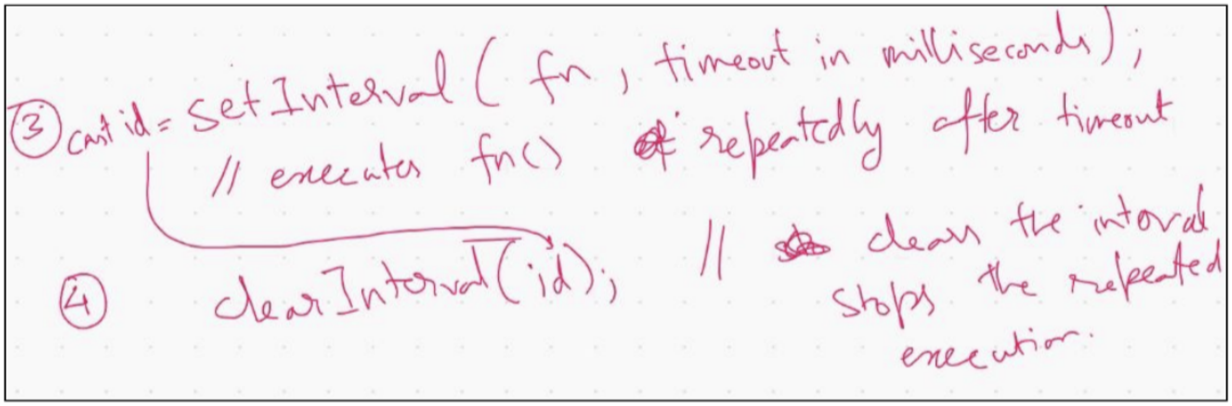
Inbuilt function in javascript

setInterval(fn,timeout in milliseconds); -> it executes fn() repeatedly after timeout



3 Co setInterval ( fn , timeout in milliseconds );
// executes fn() repeatedly after timeout

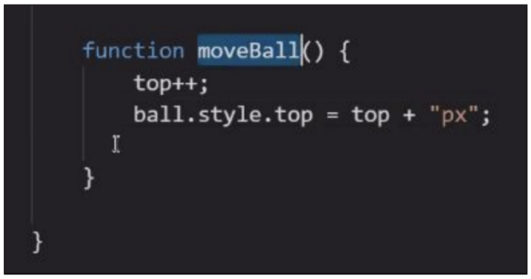
‘clearInterval(id);’ -> stops the repeated execution.



3 cnst id = setInterval ( fn , timeout in milliseconds );
// executes fn() repeatedly after timeout

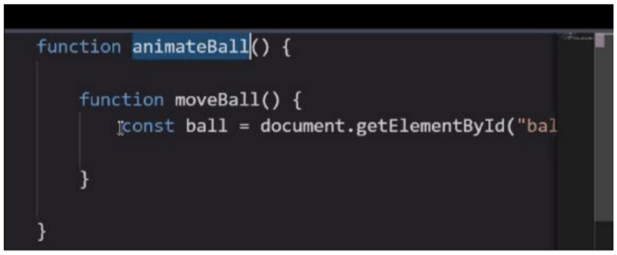
4 clearInterval(id); // clears the interval
stops the repeated
execution.

Defining a function inside nanother function



```
    function moveBall() {
        top++;
        ball.style.top = top + "px";

    }

}
```

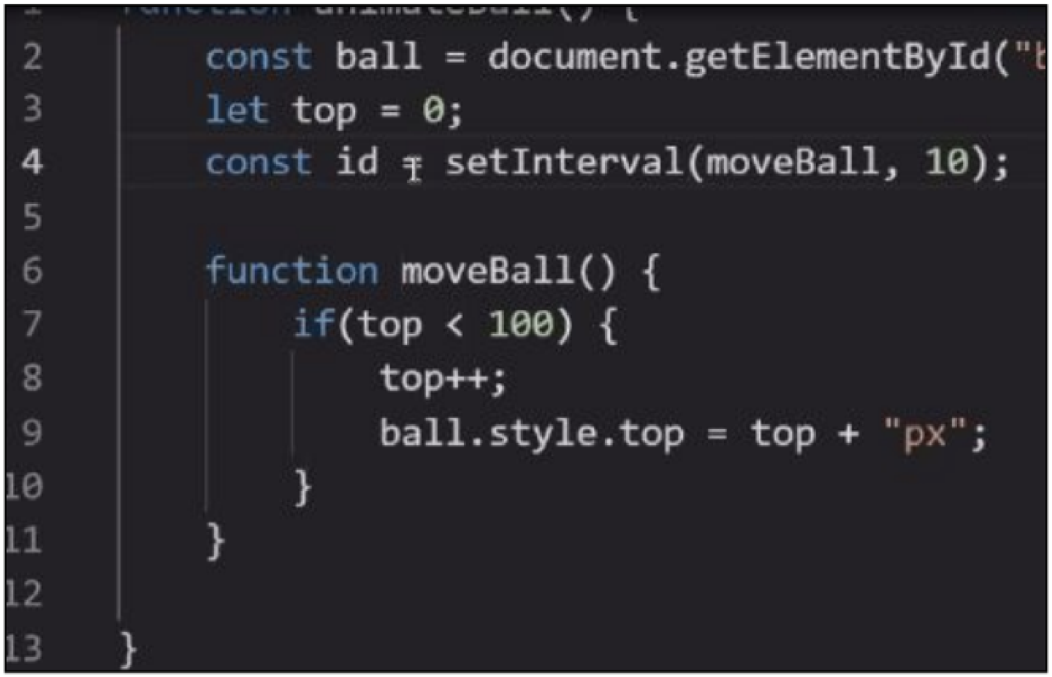
Block scoping



```
function animateBall() {

    function moveBall() {
        const ball = document.getElementById("bal

    }

}
```

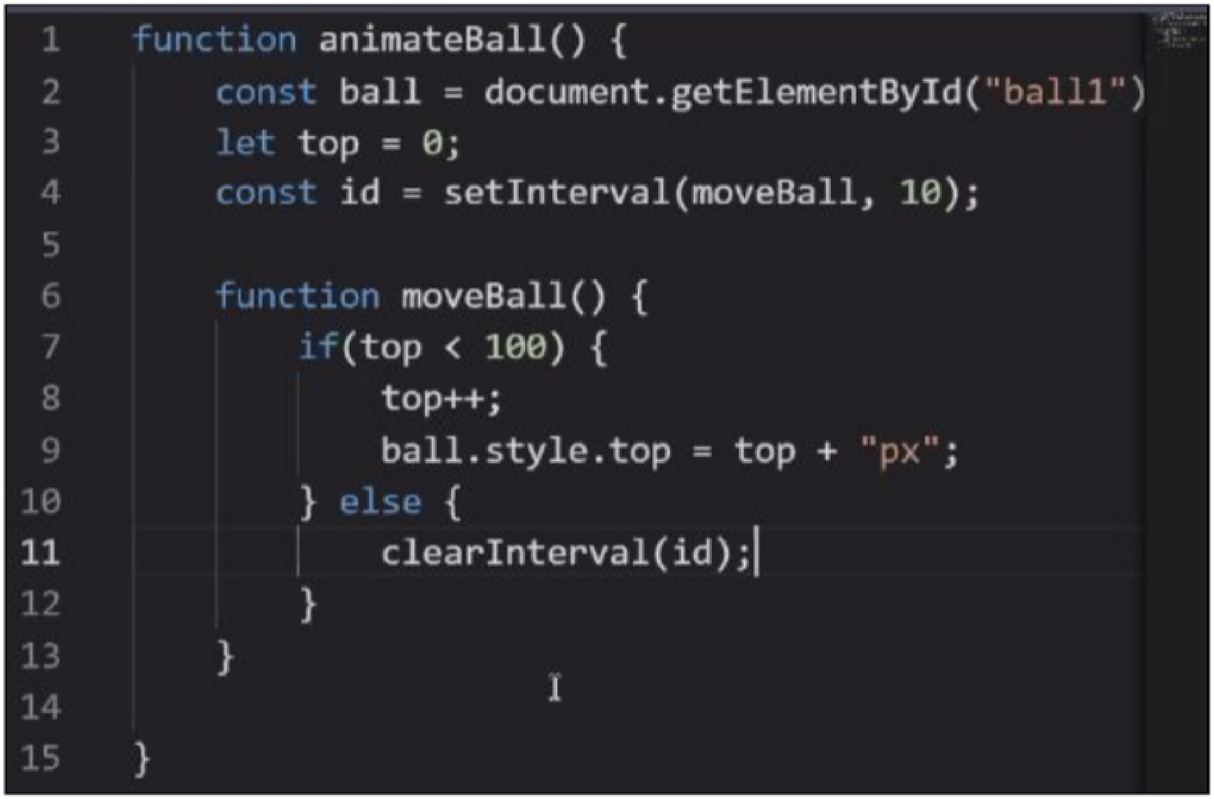
Moving the ball by one pixel in the top direction

Setting interval and calling move ball every 10milisec



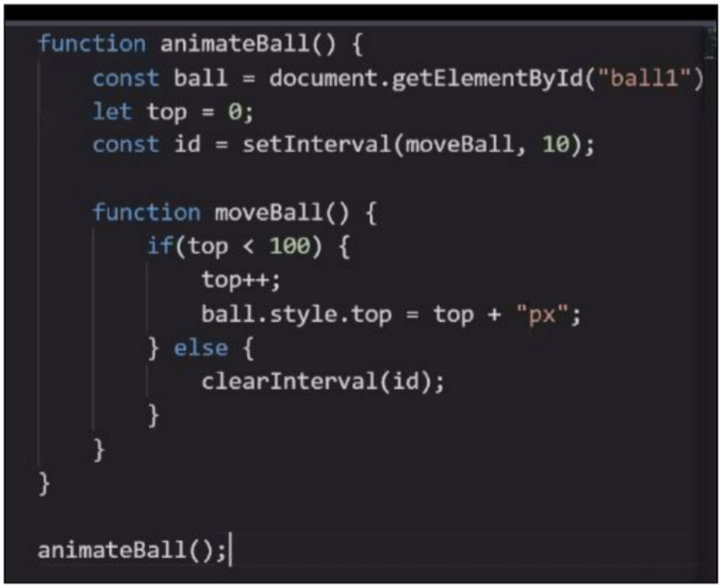
```
function animateBall() {
    const ball = document.getElementById("b
    let top = 0;
    const id = setInterval(moveBall, 10);

    function moveBall() {
        if(top < 100) {
            top++;
            ball.style.top = top + "px";
        }
    }

}
```

Stop when count reach 100

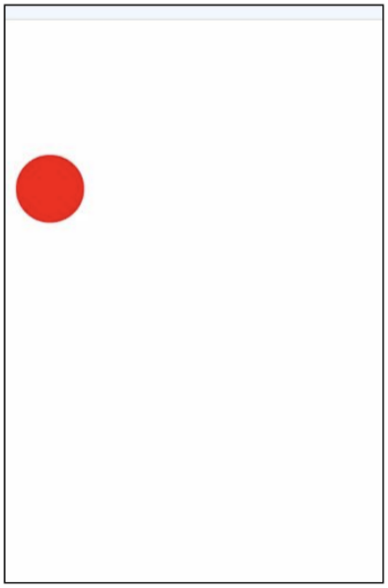


```
function animateBall() {
    const ball = document.getElementById("ball1")
    let top = 0;
    const id = setInterval(moveBall, 10);

    function moveBall() {
        if(top < 100) {
            top++;
            ball.style.top = top + "px";
        } else {
            clearInterval(id);
        }
    }

}
```

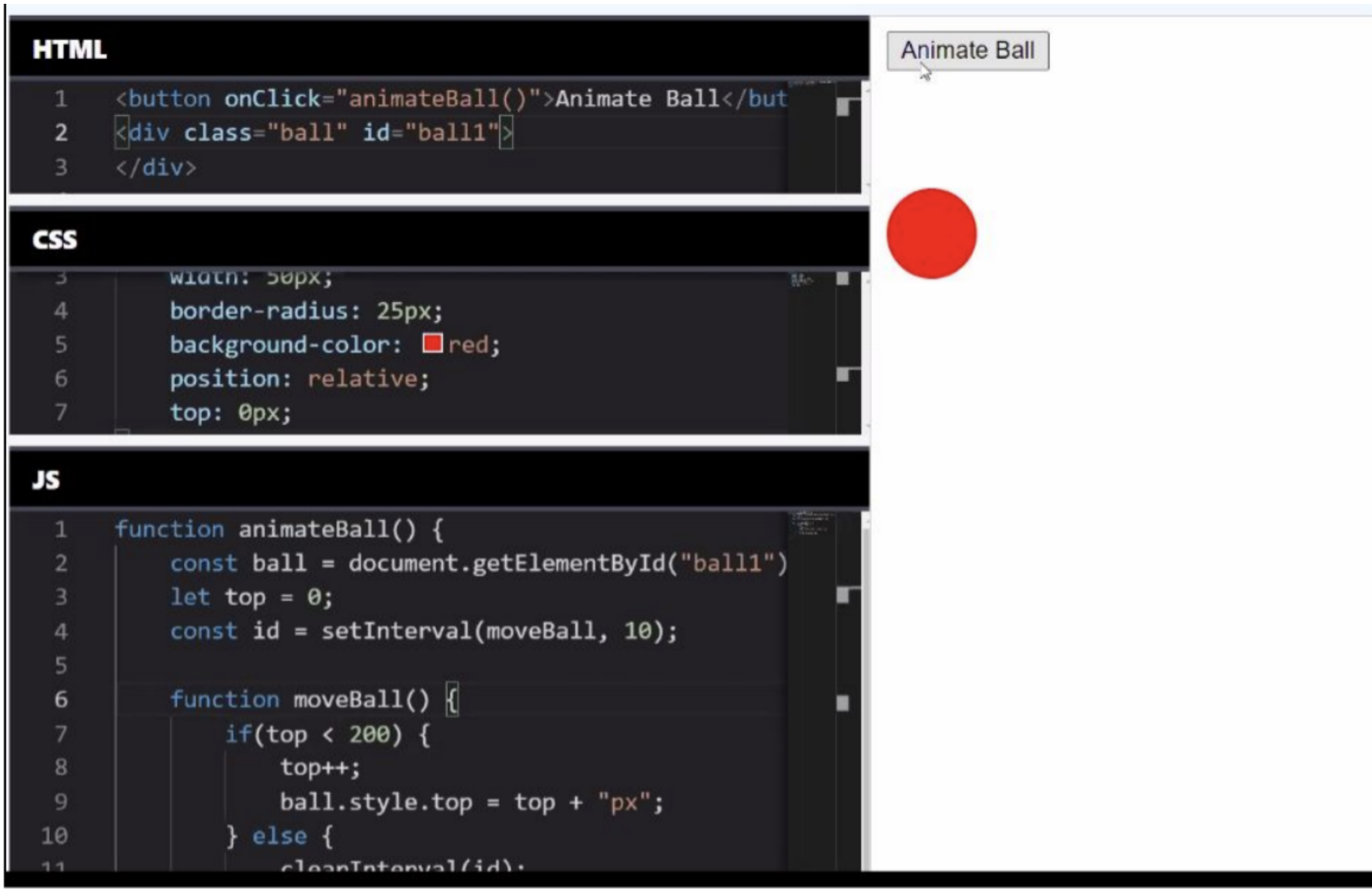
Calling the function



```
function animateBall() {
    const ball = document.getElementById("ball1")
    let top = 0;
    const id = setInterval(moveBall, 10);

    function moveBall() {
        if(top < 100) {
            top++;
            ball.style.top = top + "px";
        } else {
            clearInterval(id);
        }
    }
}

animateBall();
```



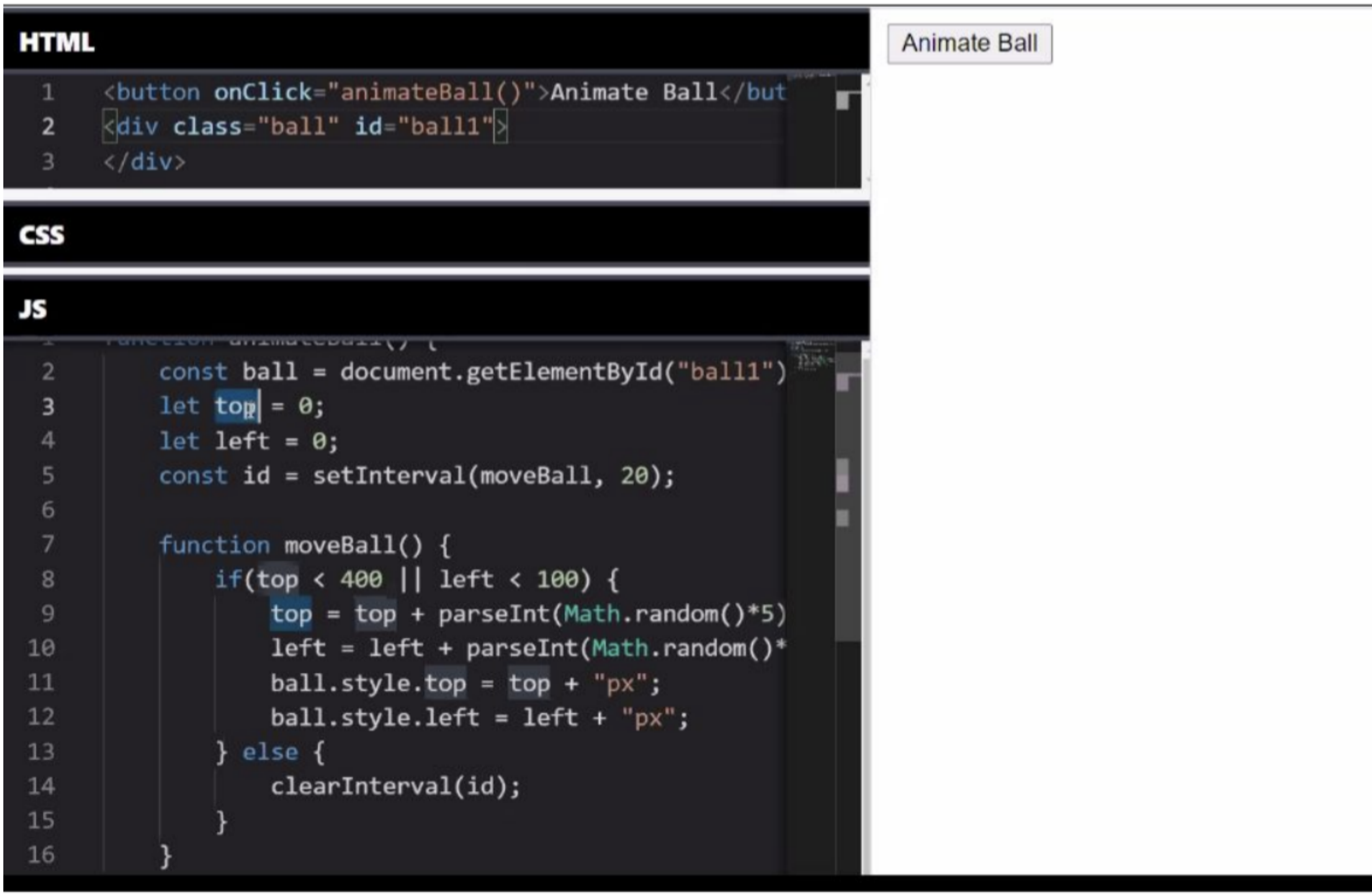
Another way of calling animate function (function called after the click on the button ‘Animate Ball’)



```html
<button onClick="animateBall()">Animate Ball</but
<div class="ball" id="ball1">
</div>
```

```css
    width: 50px;
    border-radius: 25px;
    background-color: red;
    position: relative;
    top: 0px;
```

```javascript
function animateBall() {
    const ball = document.getElementById("ball1")
    let top = 0;
    const id = setInterval(moveBall, 10);

    function moveBall() {
        if(top < 200) {
            top++;
            ball.style.top = top + "px";
        } else {
            clearInterval(id);
```



```html
<button onClick="animateBall()">Animate Ball</but
<div class="ball" id="ball1">
</div>
```

```javascript
function animateBall() {
    const ball = document.getElementById("ball1")
    let top = 0;
    let left = 0;
    const id = setInterval(moveBall, 20);

    function moveBall() {
        if(top < 400 || left < 100) {
            top = top + parseInt(Math.random()*5)
            left = left + parseInt(Math.random()*
            ball.style.top = top + "px";
            ball.style.left = left + "px";
        } else {
            clearInterval(id);
        }
    }
```