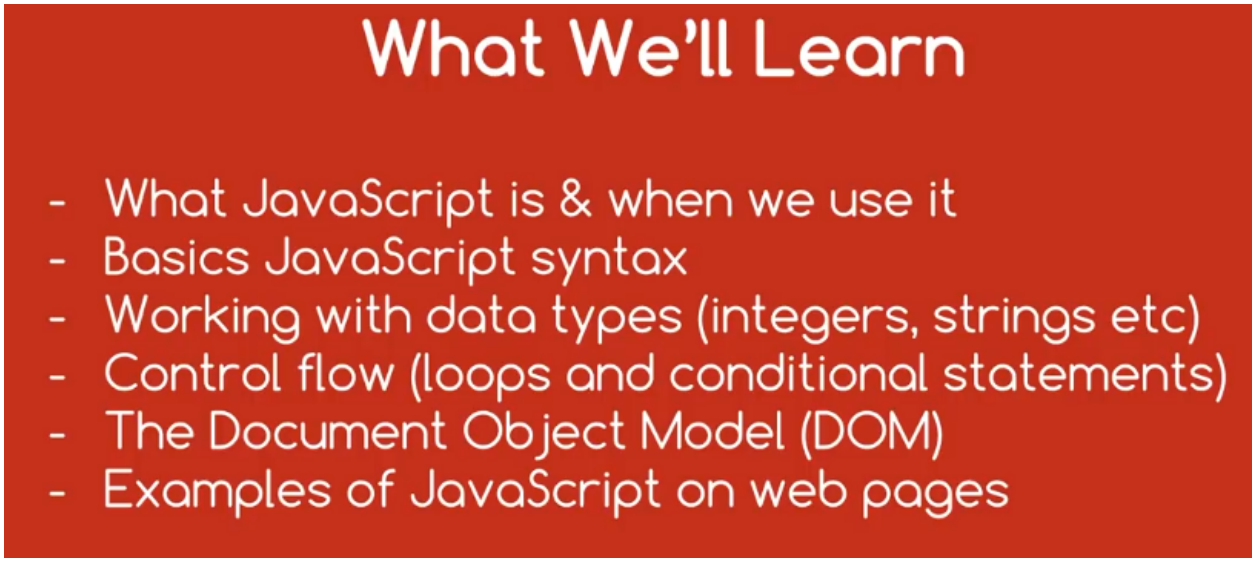
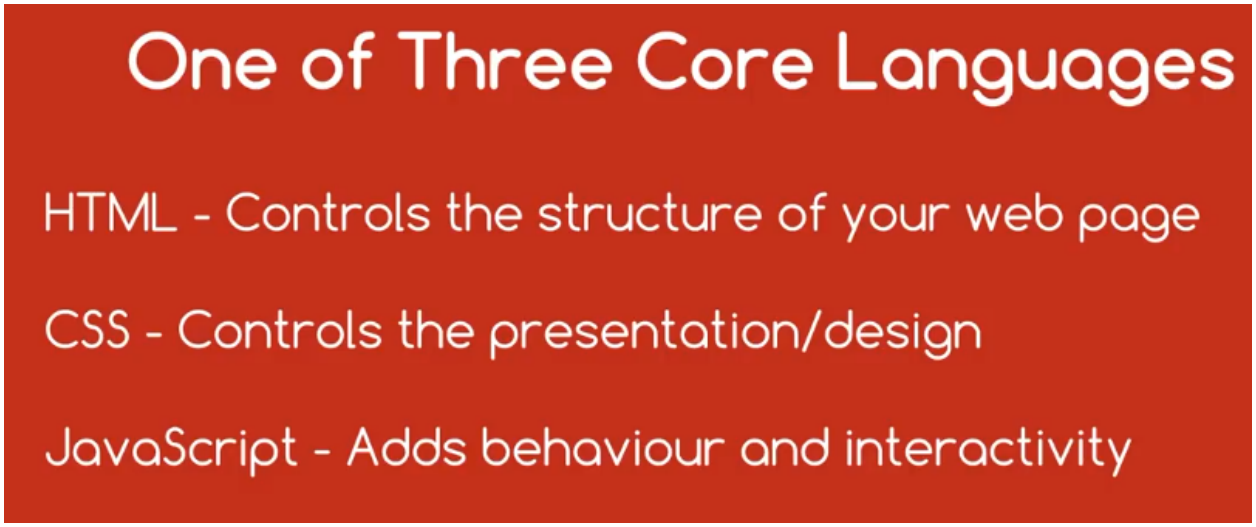


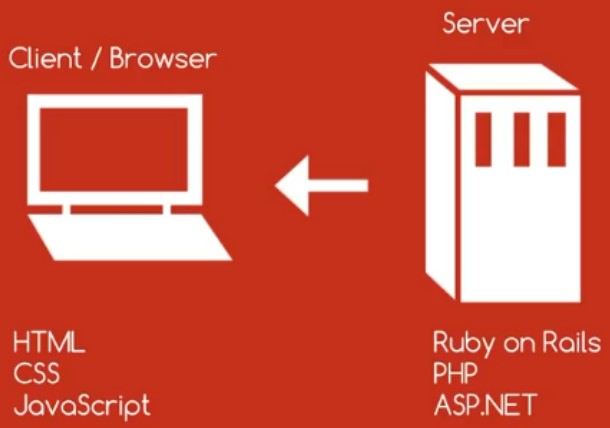
2. What is JavaScript?

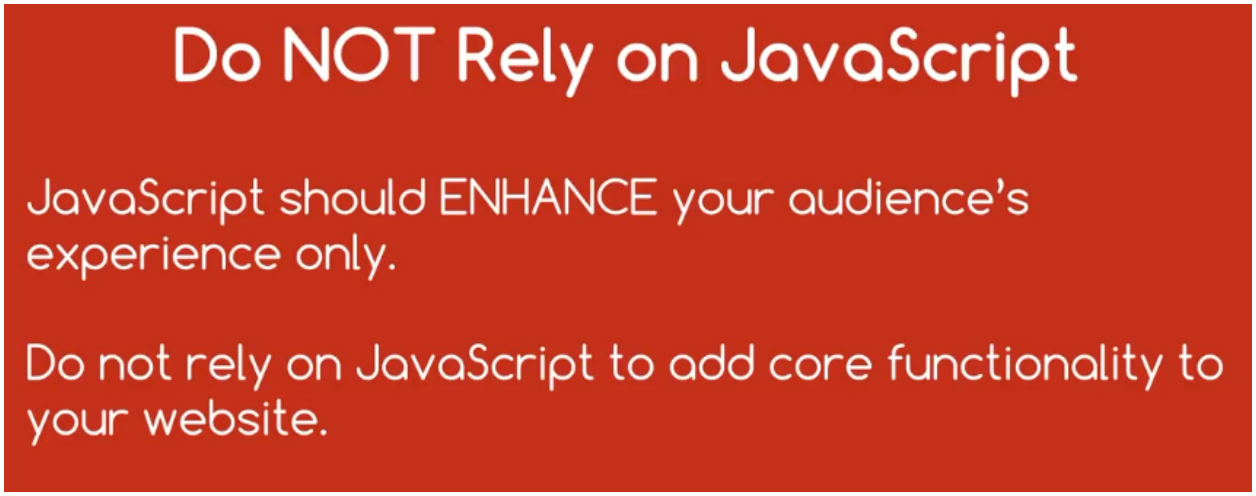


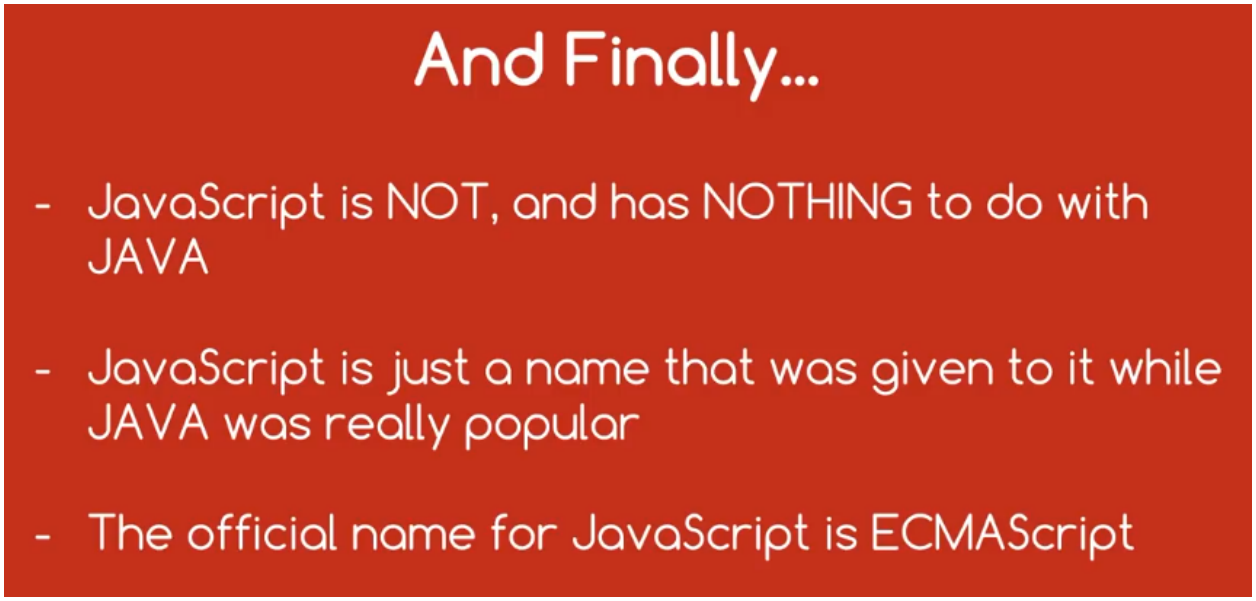
JavaScript is a Scripting Language

- Intentionally limited
- Doesn't have the same features as other programming languages (C++, Java)
- Cannot communicate directly with a database, or file system on a computer
- However it is great at manipulating web pages

JavaScript is a Client Side Language







3. Hello World! In JavaScript

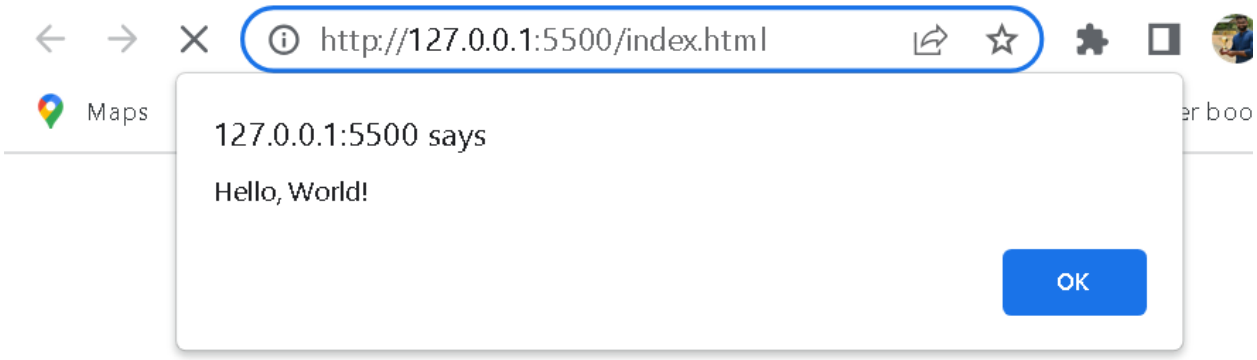
In JavaScript, the alert() function is used to display a dialog box with a message to the user. The message is typically a string or variable that you want to present as a notification or prompt.

```
C:\Users\gudiw>cd documents

C:\Users\gudiw\Documents>cd tuts

C:\Users\gudiw\Documents\tuts>mkdir JavaScript

C:\Users\gudiw\Documents\tuts>cd JavaScript

C:\Users\gudiw\Documents\tuts\JavaScript>echo. > index.html

C:\Users\gudiw\Documents\tuts\JavaScript>brackets index.html

C:\Users\gudiw\Documents\tuts\JavaScript>
```

index.html

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8" />
    <title>Hello, World</title>
</head>
<body>
    <script>
        alert("Hello, World!");
    </script>
</body>
</html>
```

O/P:

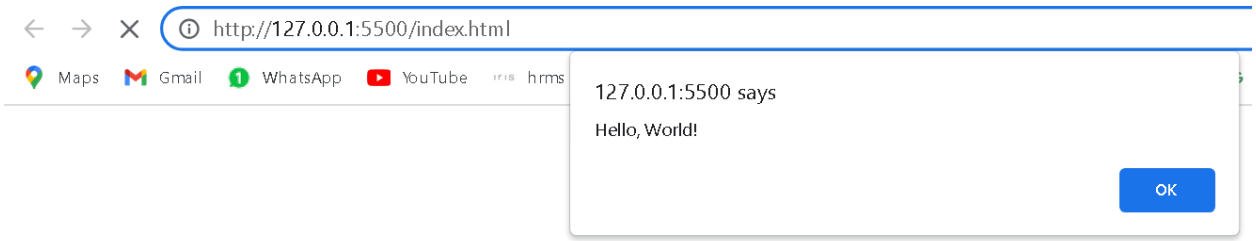


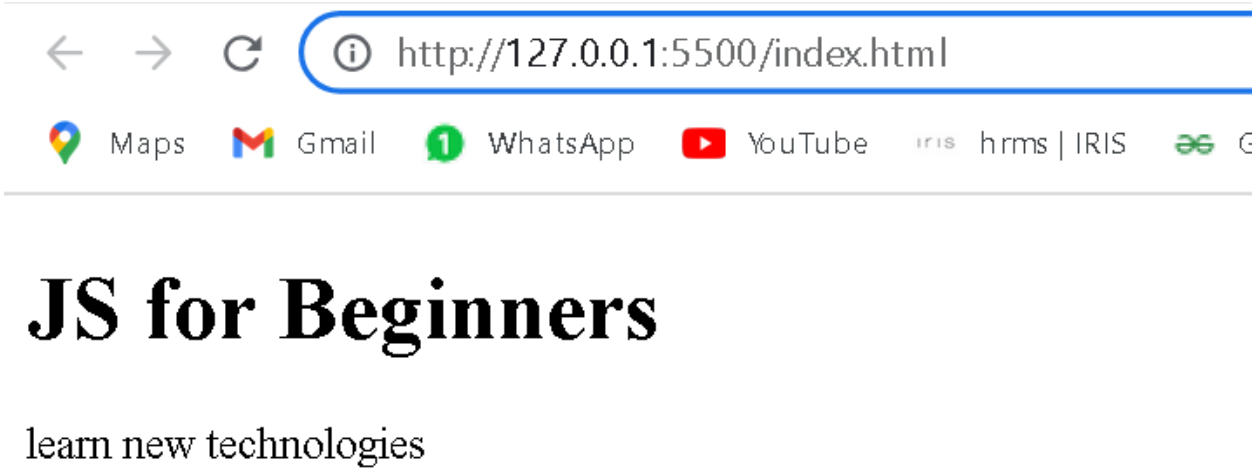
4. Where to put your JS

index.html

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <title>Hello, World</title>
</head>
<body>
    <h1>JS for Beginners</h1>

    <div id="content">
        <p>learn new technologies</p>
    </div>

    <script>
        alert("Hello, World!");
    </script>

</body>
</html>
```

O/P:





index.html

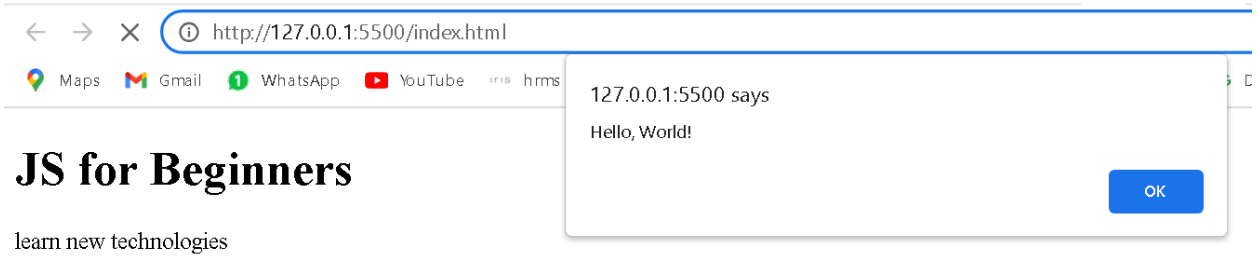
```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <title>Hello, World</title>
</head>
<body>
    <h1>JS for Beginners</h1>

    <div id="content">
```

```
        <p>learn new technologies</p>
    </div>

    <script src="test.js"></script>

</body>
</html>
```

test.js

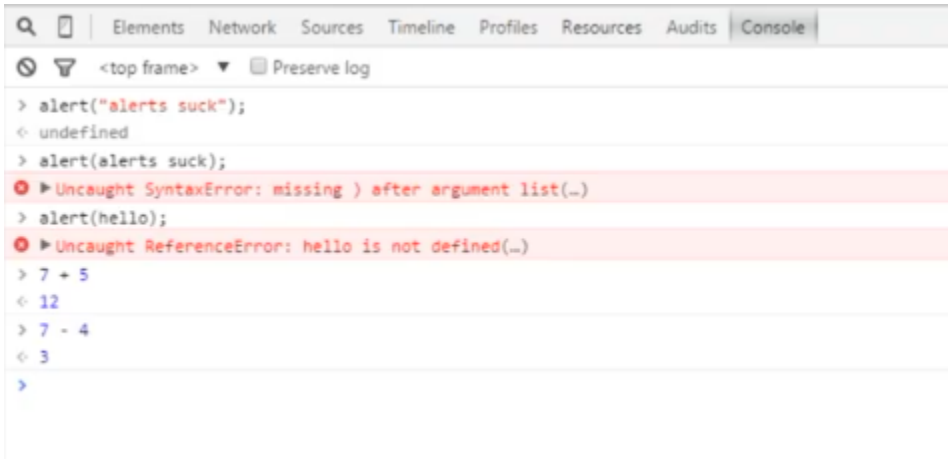
```
alert("Hello, World!");
```

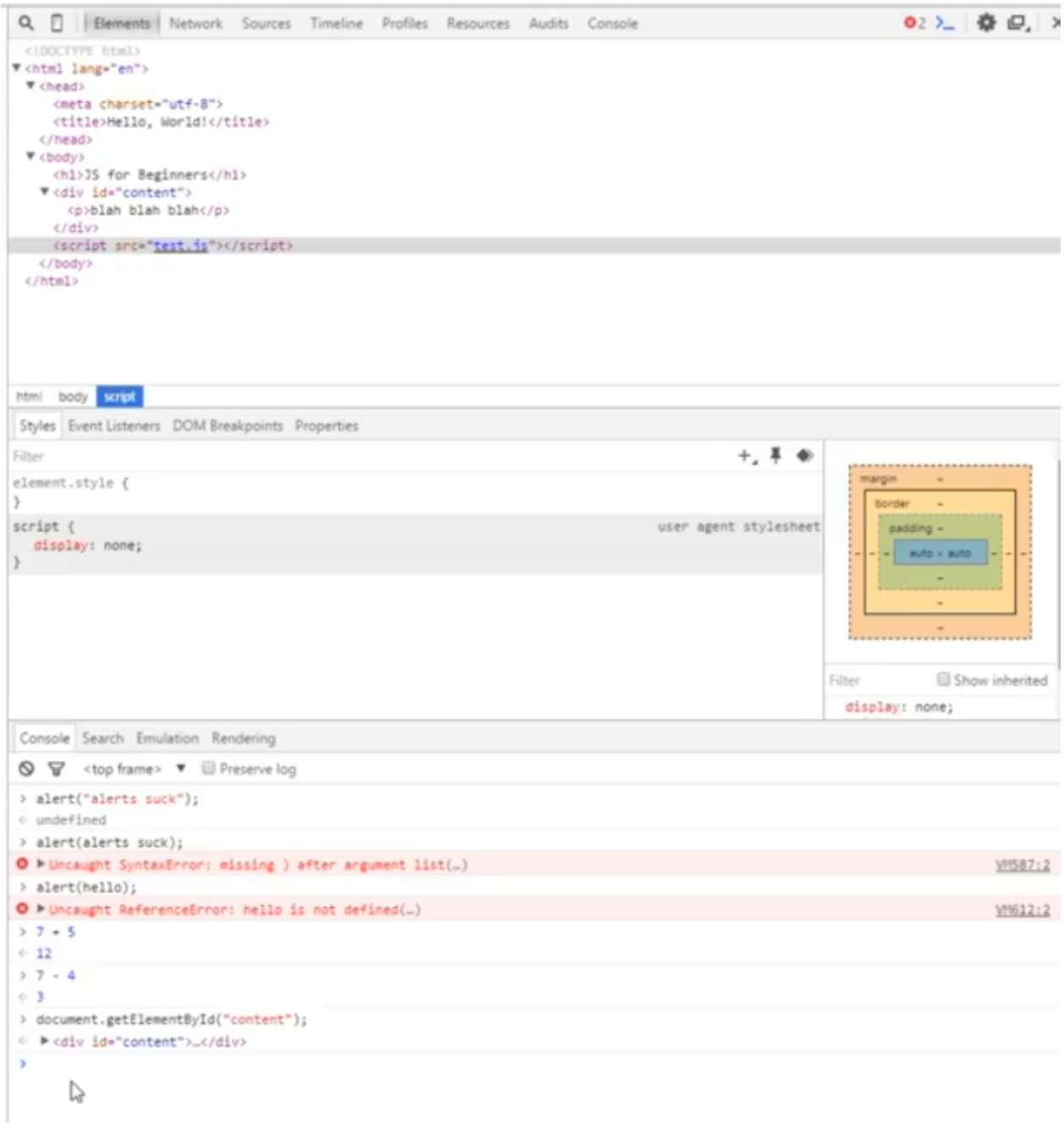
O/P:



5. Google Chrome Developer Tool

→Inspect





6. Basic JavaScript Syntax & Rules

JavaScript is a case-sensitive programming language, which means that it distinguishes between uppercase and lowercase letters. This applies to variable names, function names, object properties, and any other identifiers used in JavaScript code.

```
let message = "Hello, World!";
let Message = "Hi, there!";

console.log(message);  // Output: Hello, World!
console.log(Message);  // Output: Hi, there!
```

O/P:

```
PS C:\Users\gudiw\Documents\tuts\JavaScript> node test.js
Hello, World!
Hi, there!
```

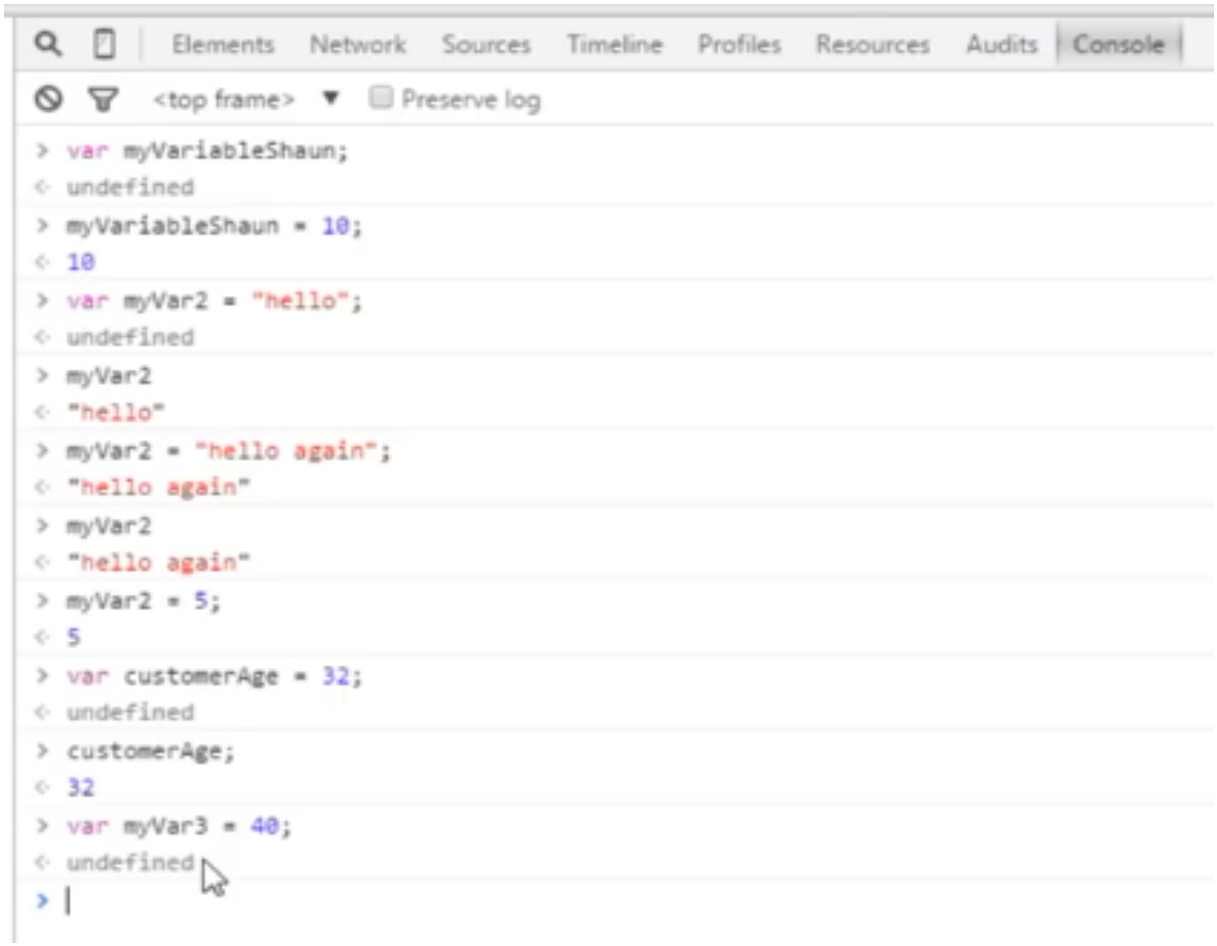
In this example, message and Message are two different variables. JavaScript treats them as separate entities because of the difference in casing.

```
//If two alerts then two times press ok in browser
alert("Hello, World!");

alert("Hello, World!");
```

Summary

- JavaScript is case sensitive
- Contains many statements, all ending with a semicolon ;
- Not sensitive to whitespace or line breaks
- Write one-line comments using //
- Write multi line comments using /* */
- JavaScripts runs from top to bottom

7. JavaScript variables

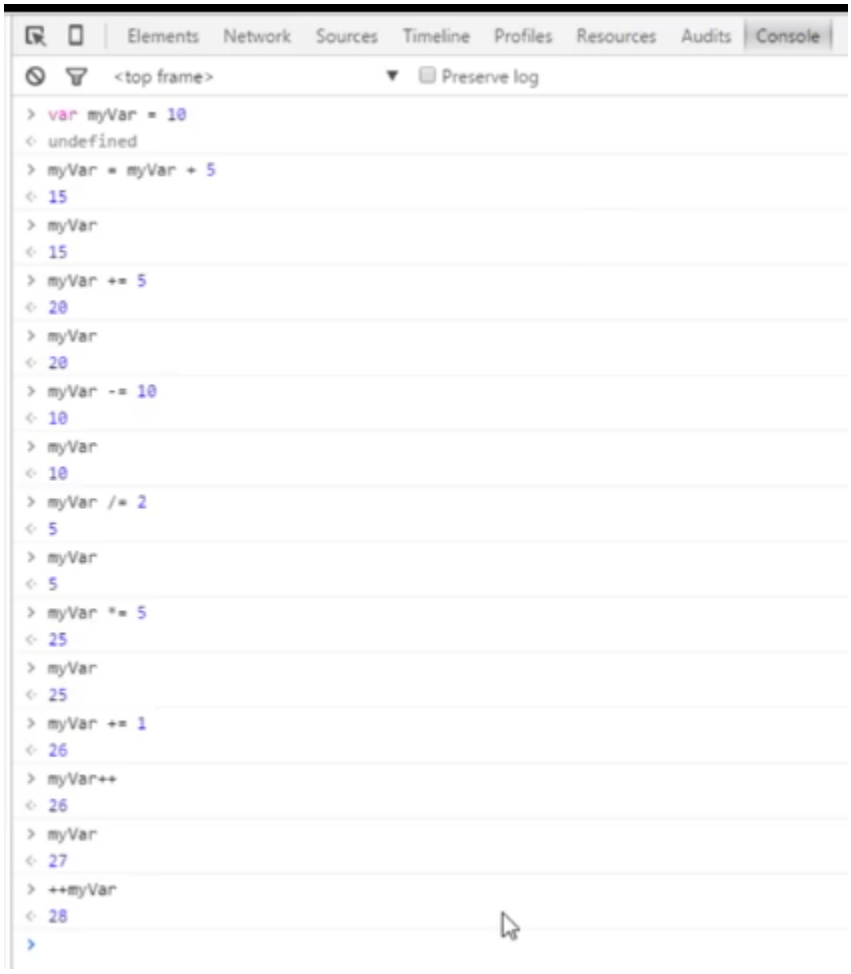


8. Basic Mathematical Operators

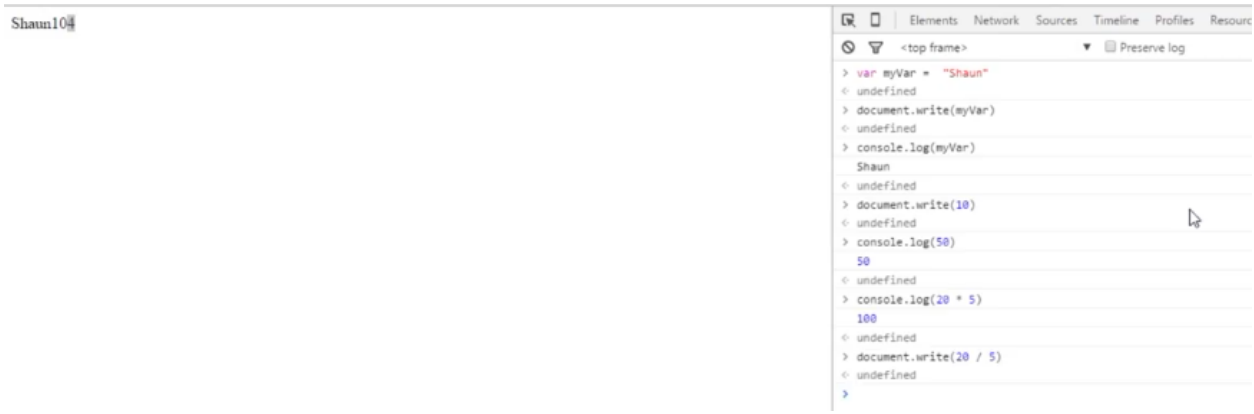

```
> var myVar = 5
< undefined
> myVar
< 5
> 5 + 5
< 10
> myVar + 10
< 15
> myVar
< 5
> myVar = myVar + 10
< 15
> myVar
< 15
> 10 - 5
< 5
> myVar - 3
< 12
> myVar = myVar - 5
< 10
> myVar
< 10
> 10 * 10
< 100
> 20 / 10
< 2
> myVar = myVar / 2
< 5
> myVar
< 5
>
```

```
> myVar
< 5
> 5 + 5
< 10
> 5 + "hello"
< "5hello"
> 100 + "hello"
< "100hello"
> "hello" + "david"
< "hellodavid"
> "hello" + " david"
< "hello david"
> 5 * "hello"
< NaN
>
```

9. Math Operator Short-hand



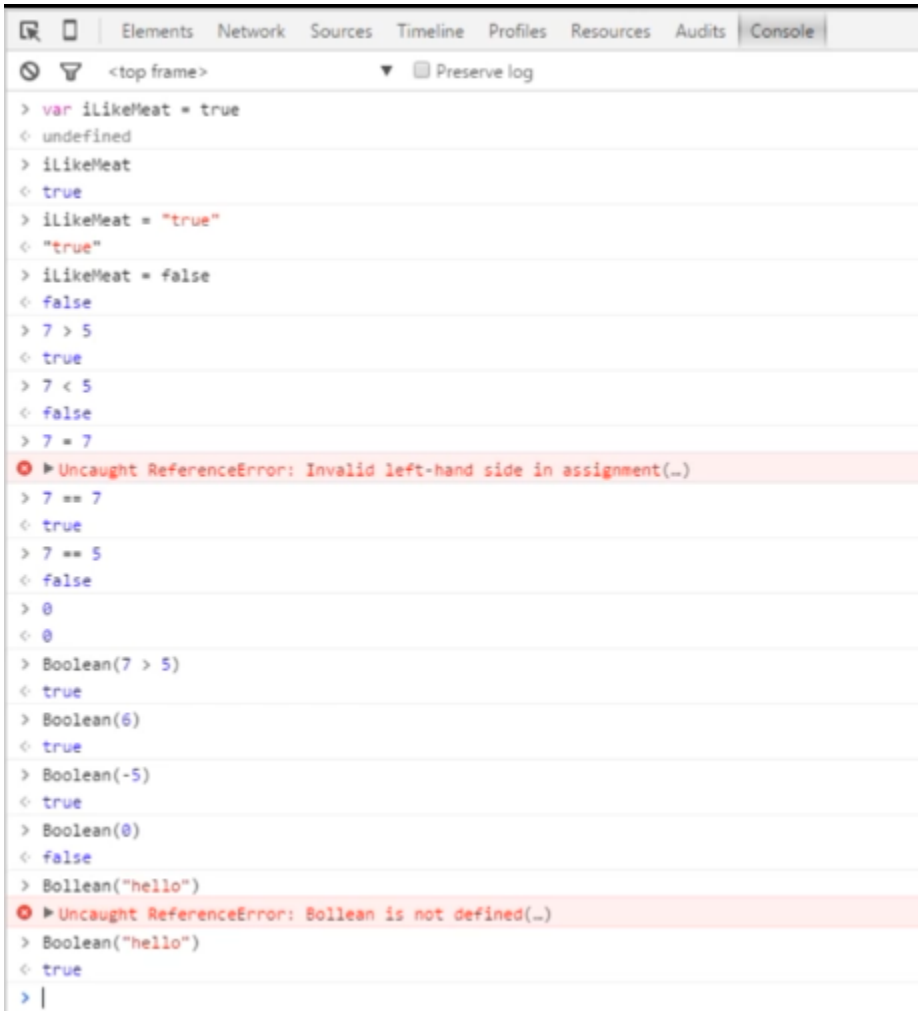
10. Logging to the Console

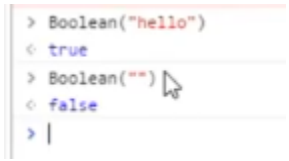


Built in object → console, document

Method →log, write

11. Booleans in JavaScript





12. If Statements

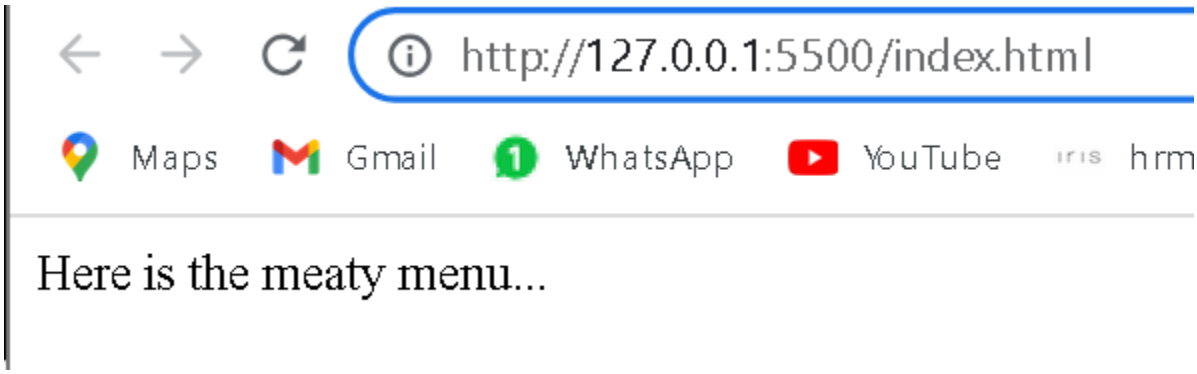
index.html

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <title>Hello, World</title>
</head>
<body>

    <script src="test.js"></script>

</body>
</html>
```

test.js

```
var youLikeMeat = true;

if (youLikeMeat){
    document.write("Here is the meaty menu...");
}
```

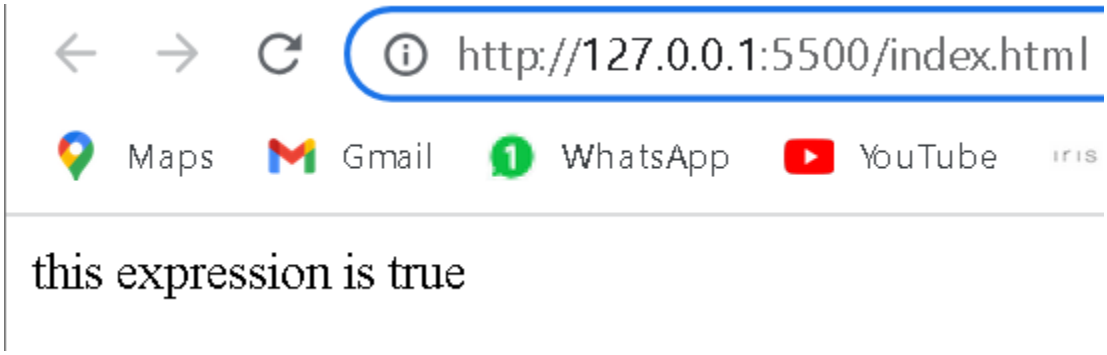
O/P:



test.js

```
var youLikeMeat = true;

if (7 > 5){
    document.write("this expression is true");
}
```

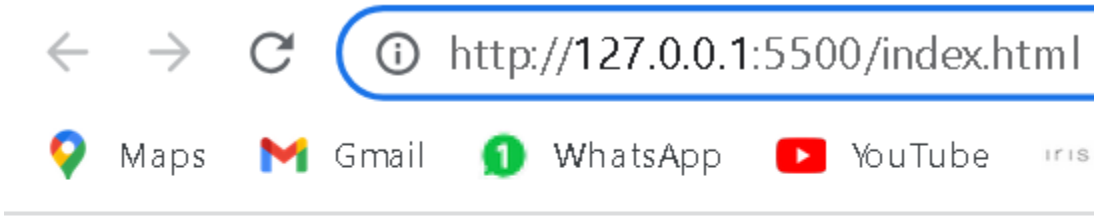
O/P:



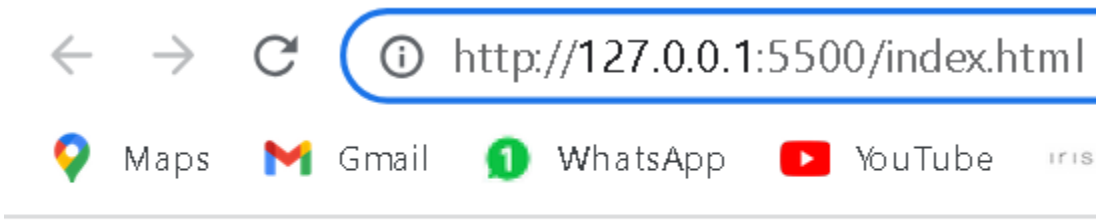
test.js

```
var youLikeMeat = true;
var myNum = 10;

if (myNum > 10){
    document.write("myNum is greater than 10");
}
```

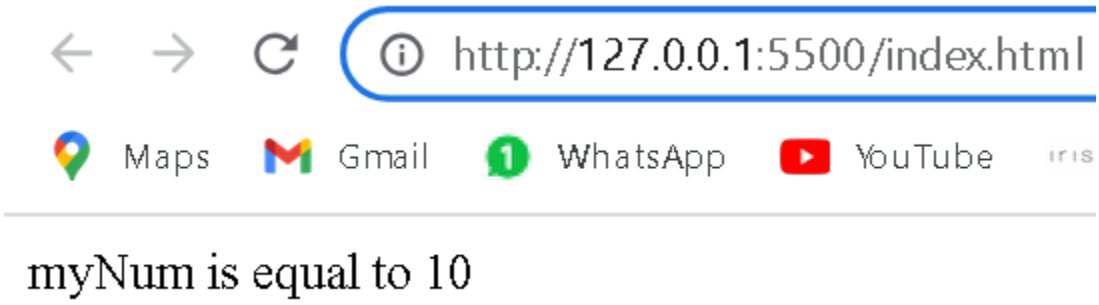
O/P:



test.js



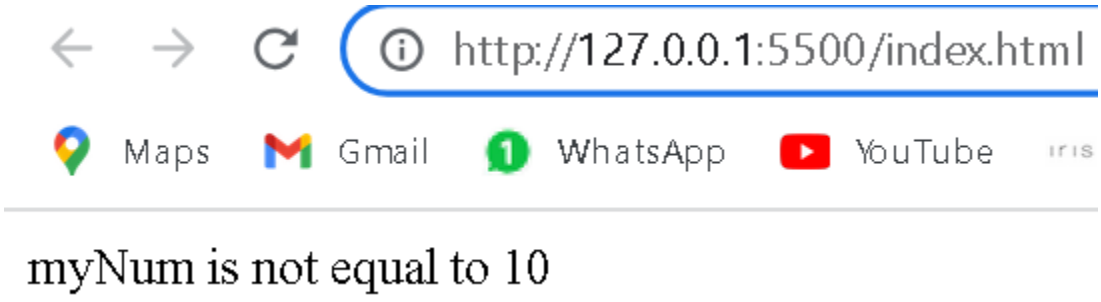
O/P:



test.js

```
var youLikeMeat = true;
var myNum = 8;

if (myNum == 10){
    document.write("myNum is equal to 10");
}else{
    document.write("myNum is not equal to 10");
}
```

O/P:

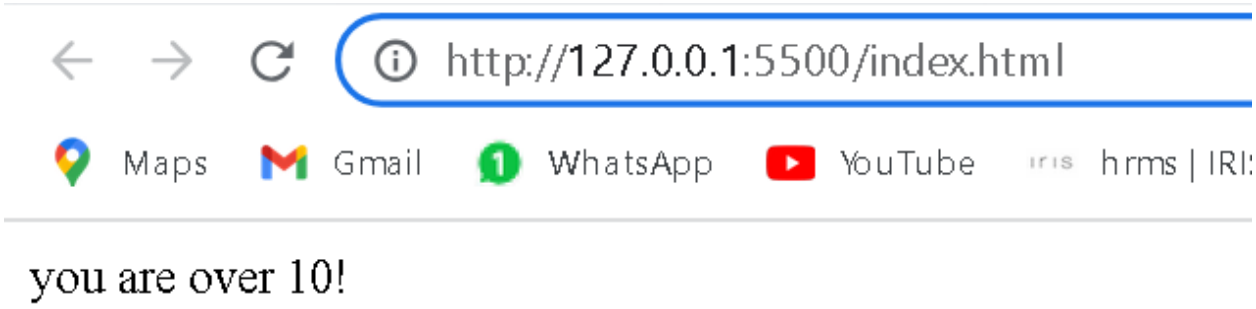


13. Else If Statements

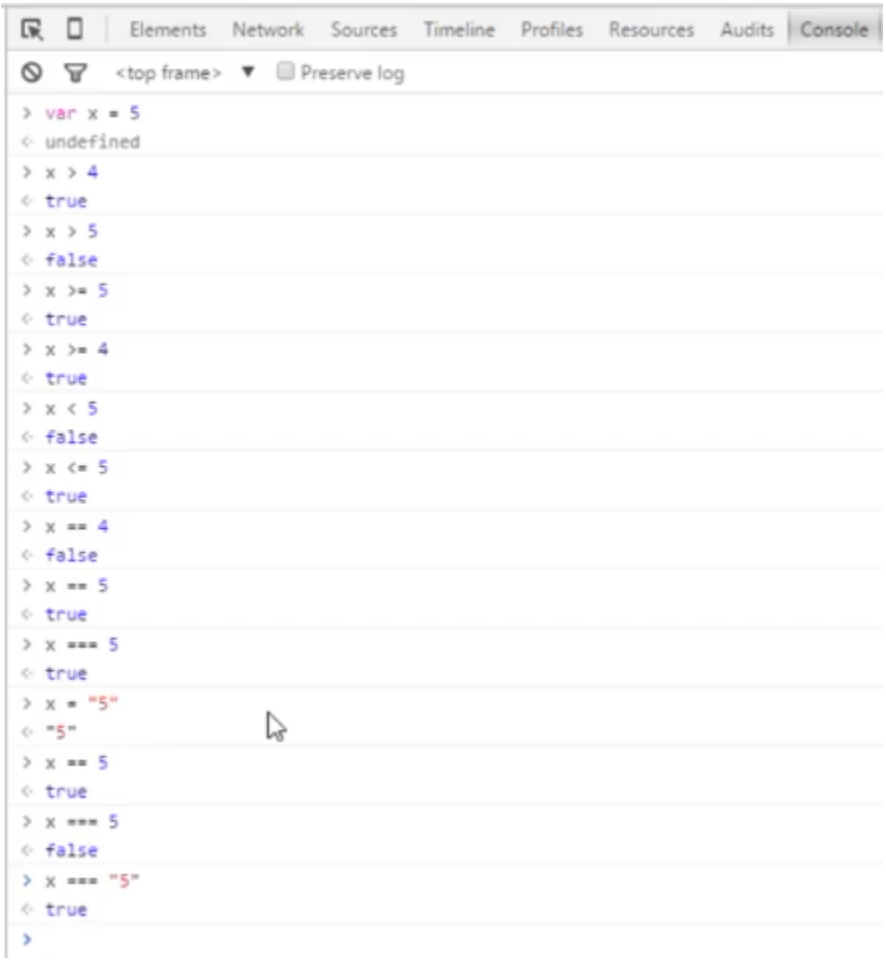
test.js

```
var myAge = 19;

if (myAge > 30){
    document.write("you are over 30!");
} else if(myAge > 20){
    document.write("you are over 20!");
} else if(myAge > 10){
    document.write("you are over 10!");
} else{
    document.write("you are not over 10!");
}
```

O/P:



14. Comparison Operators



In JavaScript, the == operator is used for equality comparison between two values. It compares the values on both sides and performs type coercion if necessary before making the comparison.

1. **Type Coercion: When comparing values using ==, JavaScript will attempt to convert the values to a common type before making the comparison. This can lead to unexpected results in certain cases.**

   console.log(5 == "5"); // Output: true

   In this case, the string "5" is converted to a number before the comparison, resulting in true.

2. **Loose Equality: The == operator performs loose equality comparison. It allows for type conversion and considers values equal if they are deemed equivalent after type coercion.**

   console.log(false == 0); // Output: true

   In this case, the boolean value false is converted to the number 0 before the comparison, resulting in true.

3. **Abstract Equality Comparison Algorithm: The == operator follows the Abstract Equality Comparison Algorithm defined in the JavaScript specification. It considers certain rules for comparing different types of values and determining their equality. It's important to be aware of these rules and understand the potential pitfalls of using ==**

In JavaScript, the === operator is the strict equality operator. It is used to compare two values for both equality of value and equality of type.

1. **Value and Type Comparison: The === operator compares the values of the operands while also checking their types. It returns true if the values and types are identical, and false otherwise.**

```
console.log(5 === 5);       // Output: true
console.log(5 === "5");     // Output: false
console.log(true === 1);    // Output: false
```

In the first comparison, both operands are numbers with the same value, so the result is true. In the second comparison, the operands have different types (number and string), so the result is false. In the third comparison, the operands have different types (boolean and number), so the result is false.

2. **Strict Comparison: The === operator does not perform type coercion. It requires the values being compared to have the same type to be considered equal. This can help prevent unexpected behavior that may occur due to type coercion with the loose equality operator ==.**
3. **Recommended Usage: It's generally recommended to use the === operator for equality comparisons in JavaScript, as it provides more precise control over the comparison and avoids unexpected type coercion. By using strict equality, you can ensure that the values being compared are both equal in value and type.**

In JavaScript, the != operator is the inequality operator. It is used to check if two values are not equal to each other, regardless of their types.

1. **Value Comparison: The != operator compares the values of the operands and returns true if the values are not equal, and false if the values are equal. Type coercion is performed if necessary before the comparison.**

```
console.log(5 != 5);       // Output: false
console.log(5 != "5");     // Output: false
console.log(true != 1);    // Output: false
console.log(true != false);  // Output: true
```

In the first three comparisons, the operands have different types, but their values are considered equal when using the != operator. Only in the last comparison, where true and false are compared, the result is true because they are not equal.

2. **Type Coercion: Similar to the loose equality operator ==, the != operator performs type coercion if necessary before the comparison. This can lead to unexpected results, especially when comparing values of different types.**
3. **Inverse of Equality: The != operator is the inverse of the equality operator ==. It provides the opposite result of the equality comparison. For example, x != y is equivalent to !(x == y).**

In JavaScript, the !== operator is the strict inequality operator. It is used to compare two values for both inequality of value and inequality of type.

1. **Value and Type Comparison: The !== operator compares the values of the operands while also checking their types. It returns**

true if the values or types are different, and **false** if the values and types are identical.

```
console.log(5 !== 5);       // Output: false
console.log(5 !== "5");     // Output: true
console.log(true !== 1);    // Output: true
console.log(true !== false);  // Output: true
```

In the first comparison, both operands are numbers with the same value, so the result is **false**. In the second comparison, the operands have different types (number and string), so the result is **true**. In the third comparison, the operands have different types (boolean and number), so the result is **true**. In the last comparison, both operands have the same type (boolean), but their values are different, so the result is **true**.

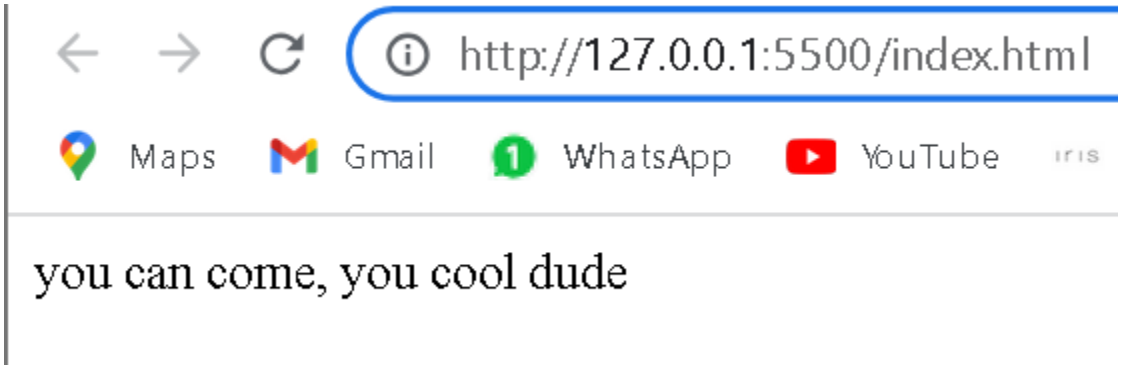
2. Strict Comparison: The **!==** operator does not perform type coercion. It requires the values being compared to have both different values and different types to be considered unequal.
3. Recommended Usage: It's generally recommended to use the !== operator for inequality comparisons in JavaScript, as it provides more precise control over the comparison and avoids unexpected type coercion. By using strict inequality, you can ensure that the values being compared are both unequal in value and type.

15. Logical Operators

test.js

```
var myAge = 25;

if (myAge >= 18 && myAge <= 30){
    document.write("you can come, you cool dude");
} else{
    document.write("you aint coming");
}
```

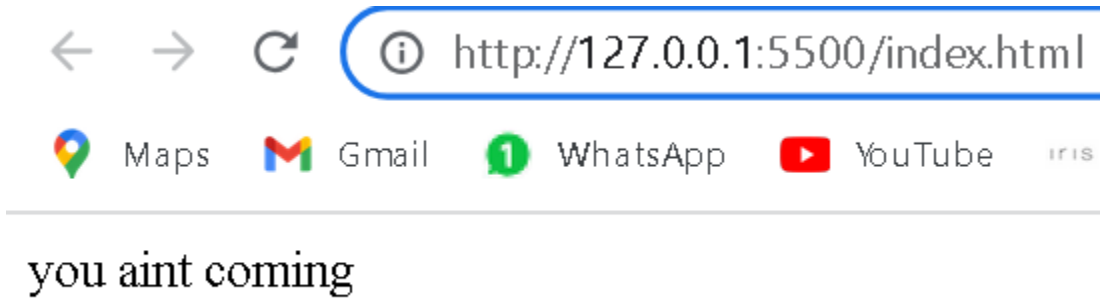
O/P:



test.js

```
var myAge = 35;

if (myAge < 18 || myAge > 30){
    document.write("you aint coming");
} else{
    document.write("you can come, you cool dude");
}
```

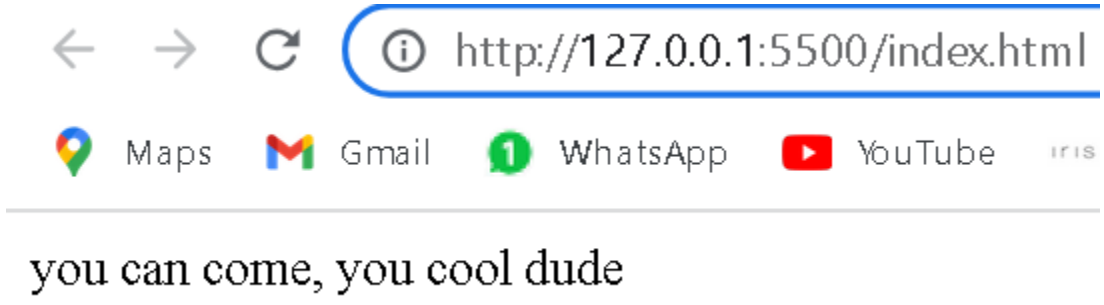
O/P:



test.js

```
var myAge = 26;

if (myAge < 18 || myAge > 30 || myAge === 25){
    document.write("you aint coming");
} else{
    document.write("you can come, you cool dude");
}
```

O/P:

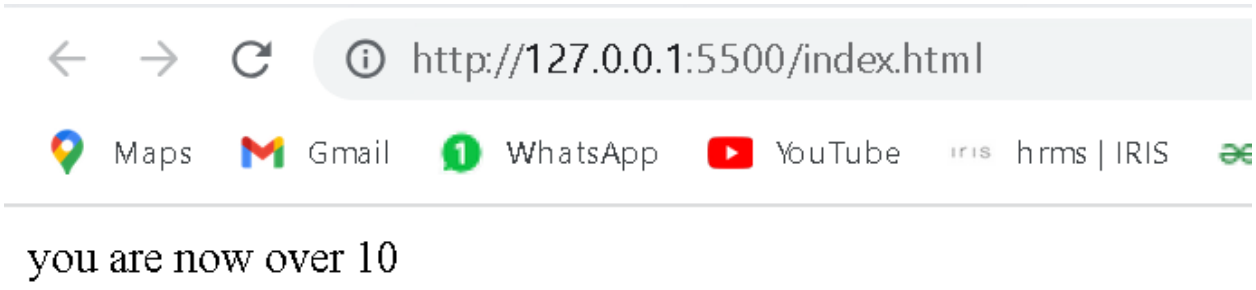


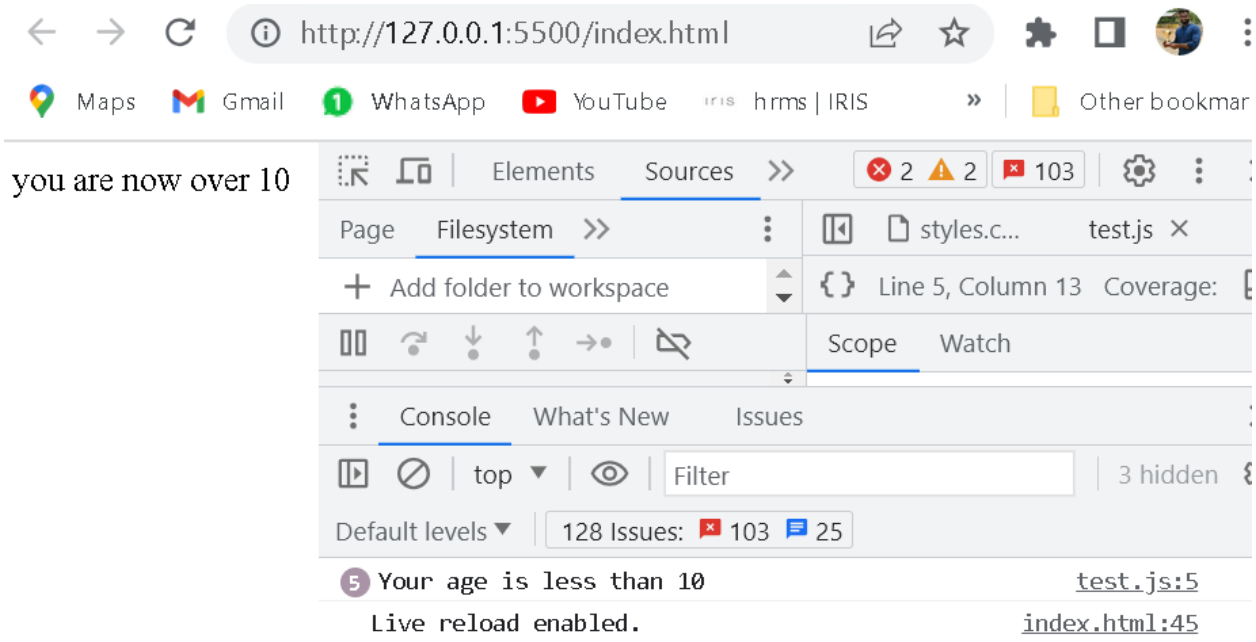
16. While Loops

test.js

```
var age = 5;

while (age < 10) {

    console.log("Your age is less than 10");
    age++;

}

document.write("you are now over 10");
```

O/P:





17. For Loops

index.html

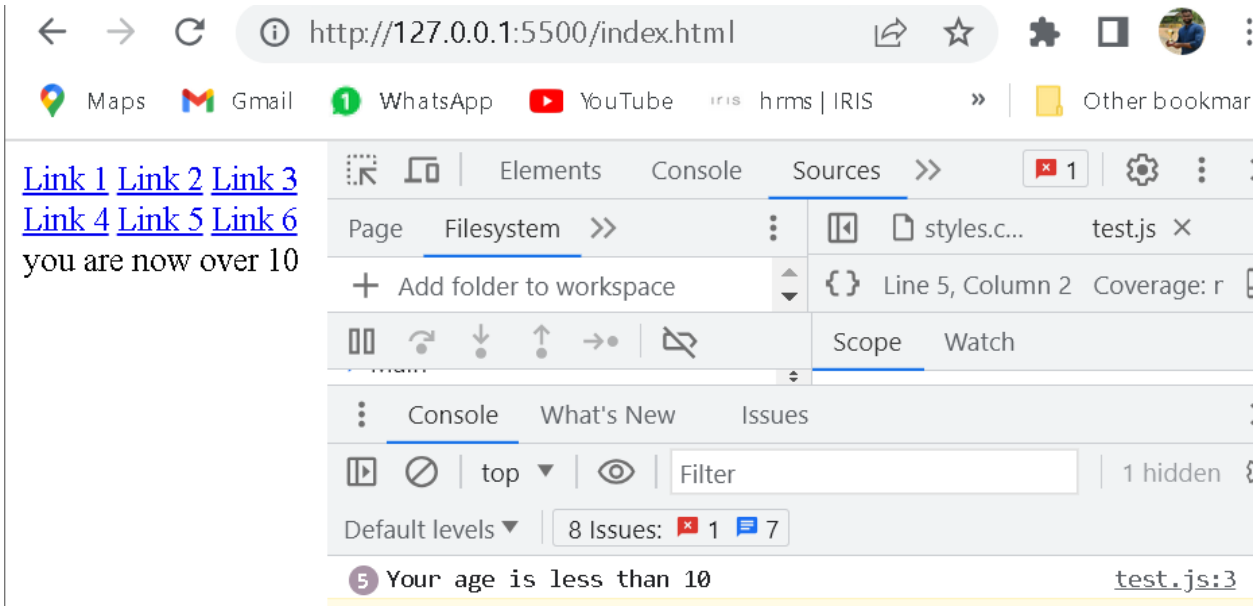
```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <title>JavaScript for Beginners</title>
</head>
<body>
   <a href="#">Link 1</a>
   <a href="#">Link 2</a>
   <a href="#">Link 3</a>
   <a href="#">Link 4</a>
   <a href="#">Link 5</a>
   <a href="#">Link 6</a>

    <script src="test.js"></script>
```

```
</body>
</html>
```

test.js

```
for (age = 5; age < 10; age++) {

    console.log("Your age is less than 10");

}

document.write("you are now over 10");
```

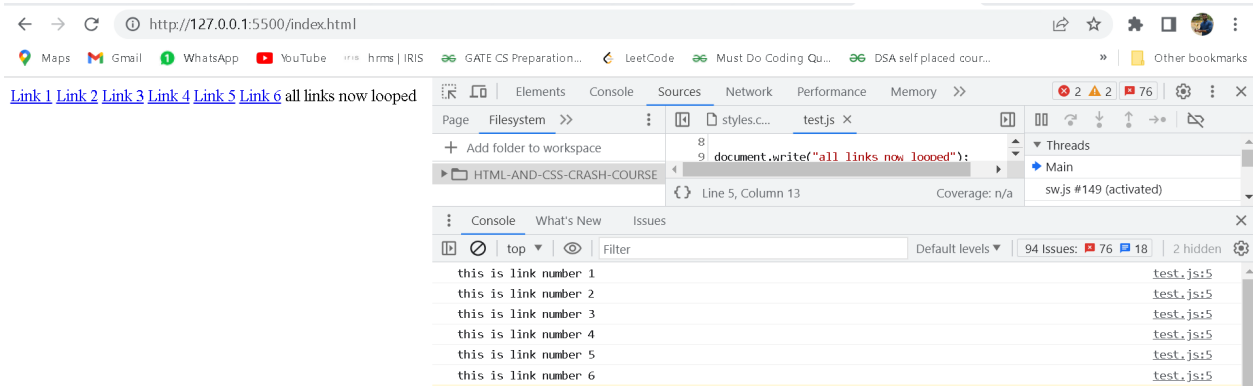
O/P:



test.js

```
var links = document.getElementsByTagName("a");

for (i = 1; i <= links.length; i++) {

    console.log("this is link number " + i);

}

document.write("all links now looped");
```

O/P:



18. Break & Continue

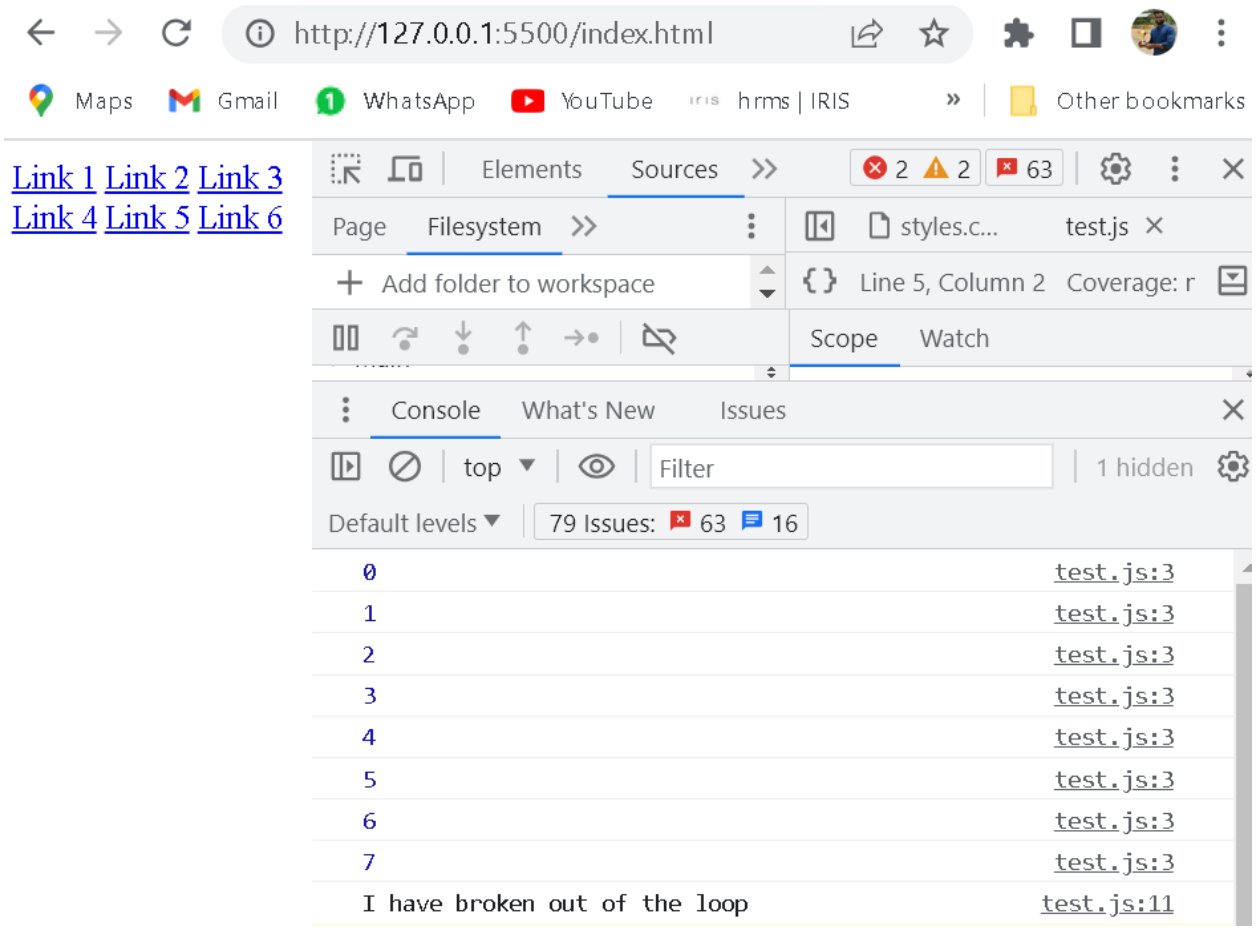
Break

test.js

```
for (i = 0; i < 10; i++) {

    console.log(i);

    if(i === 7){
        break;
```

```
    }

}

console.log("I have broken out of the loop");
```

O/P:

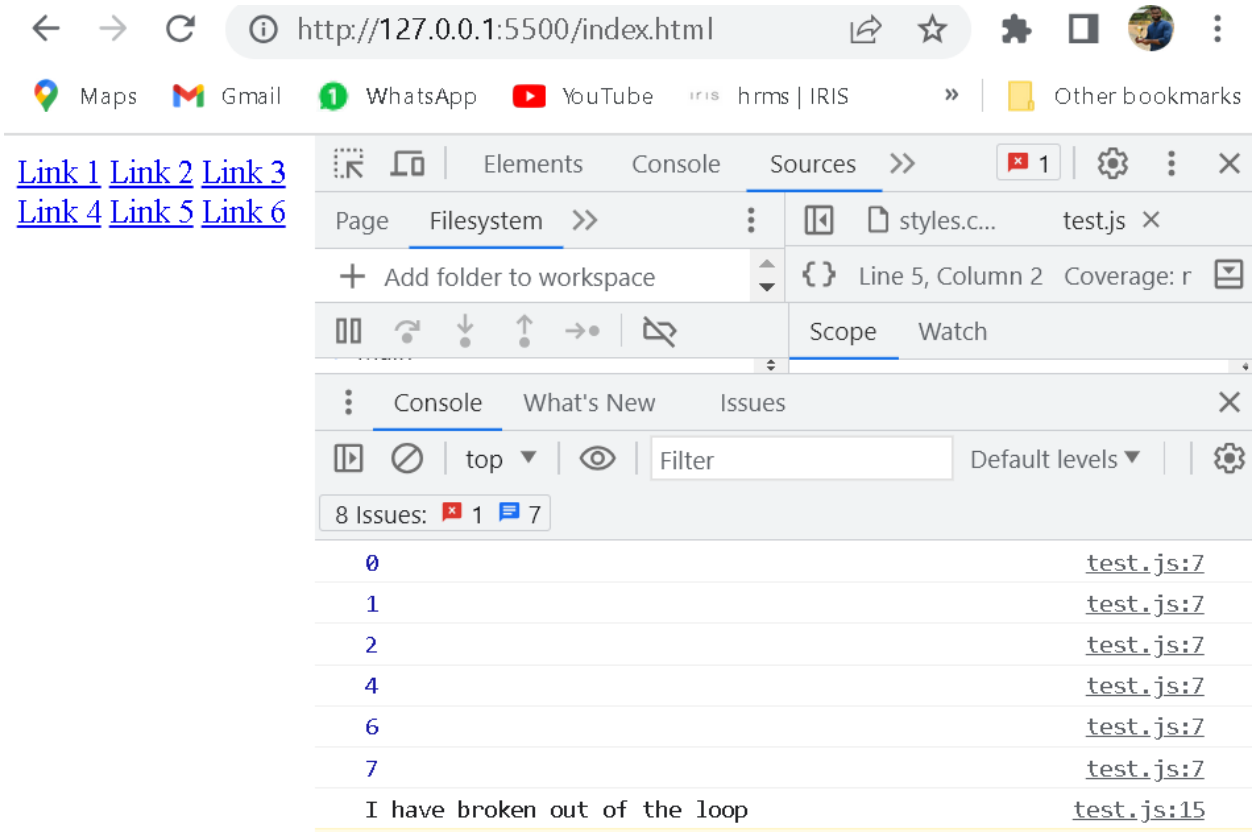


Continue

test.js

```
for (i = 0; i < 10; i++) {

    if(i === 5 || i == 3){
        continue;
    }

    console.log(i);

    if(i === 7){
        break;
    }

}

console.log("I have broken out of the loop");
```

O/P:



19. Practical Example Using Loops

index.html

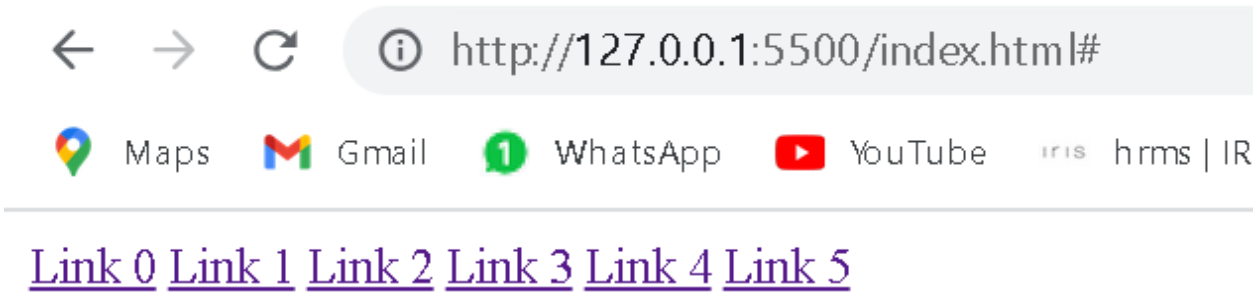
```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <title>JavaScript for Beginners</title>
</head>
<body>

   <a href="#">Link 0</a>
   <a href="#">Link 1</a>
   <a href="#">Link 2</a>
```

```
    <a href="#">Link 3</a>
    <a href="#">Link 4</a>
    <a href="#">Link 5</a>

     <script src="test.js"></script>

</body>
</html>
```

test.js

```
var links = document.getElementsByTagName("a");

for (i = 0; i < links.length; i++) {

    links[i].className = "link-" + i;

}
```

O/P:



20. Functions

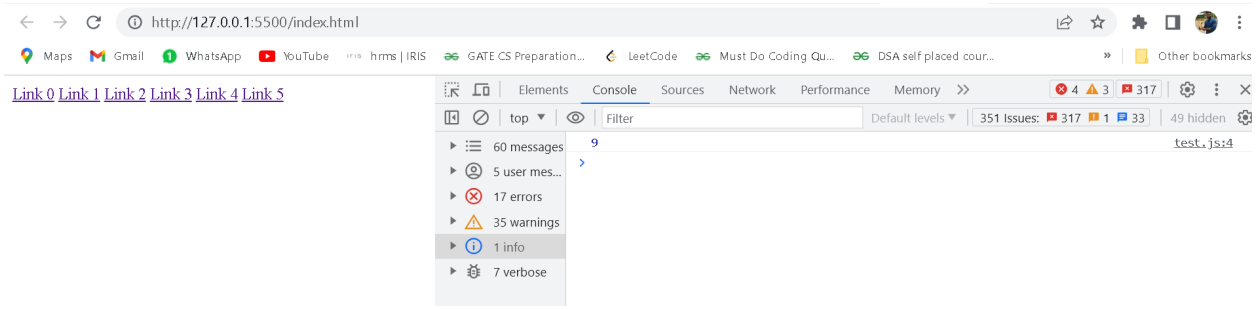
index.html

```
<!DOCTYPE html>
<html>

<head>
    <meta charset="utf-8">
    <title>JavaScript for Beginners</title>
</head>
<body>

   <a href="#">Link 0</a>
   <a href="#">Link 1</a>
   <a href="#">Link 2</a>
   <a href="#">Link 3</a>
   <a href="#">Link 4</a>
   <a href="#">Link 5</a>

    <script src="test.js"></script>

</body>
</html>
```

test.js

```
function getAverage(a, b) {

    var average = (a + b) / 2;
    console.log(average);

}

getAverage(7, 11);
```

O/P:

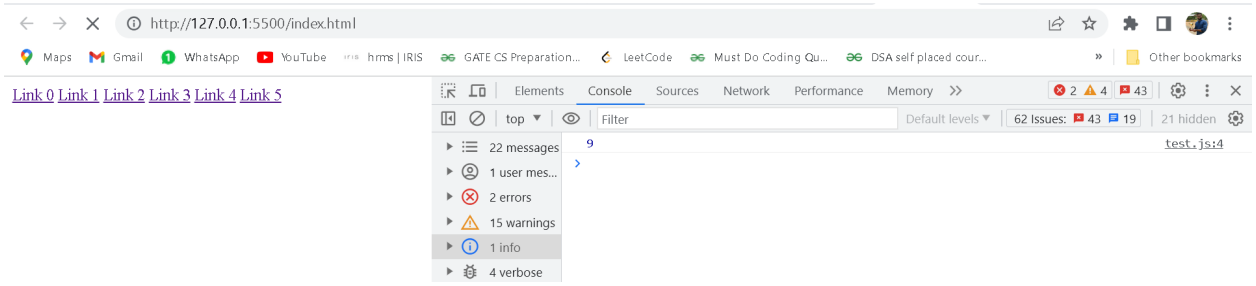


text.js

```
function getAverage(a, b) {

    var average = (a + b) / 2;
    console.log(average);

}

getAverage(7, 11, 13);
```

O/P:

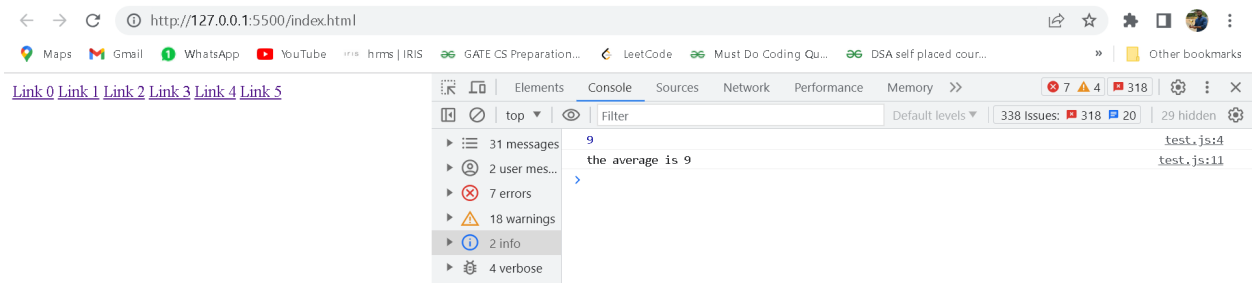
→Here the output of the first two numbers & last one is ignored.



test.js

```
function getAverage(a, b) {

    var average = (a + b) / 2;
    console.log(average);
    return average;

}


var myResult = getAverage(7, 11);
console.log("the average is " + myResult);
```

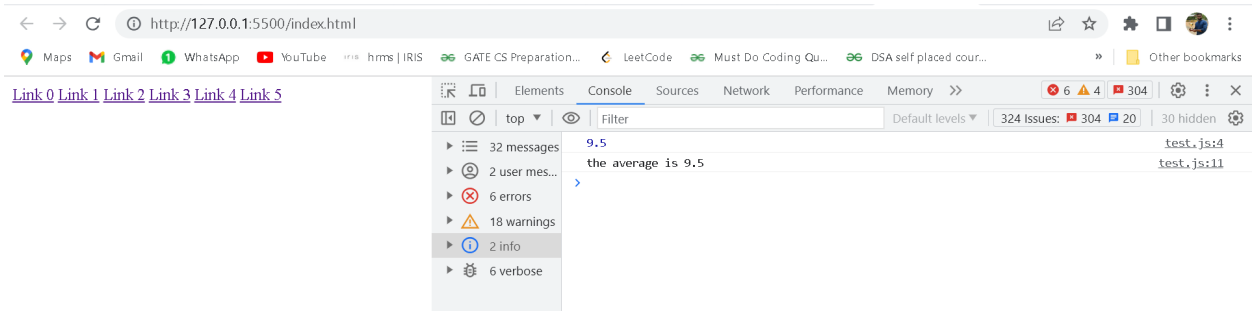
O/P:



test.js

```
function getAverage(a, b, c, d, e, f) {

    var average = (a + b + c + d + e + f) / 6;
    console.log(average);
    return average;

}


var myResult = getAverage(7, 8, 9, 10, 11, 12);
console.log("the average is " + myResult);
```

O/P:

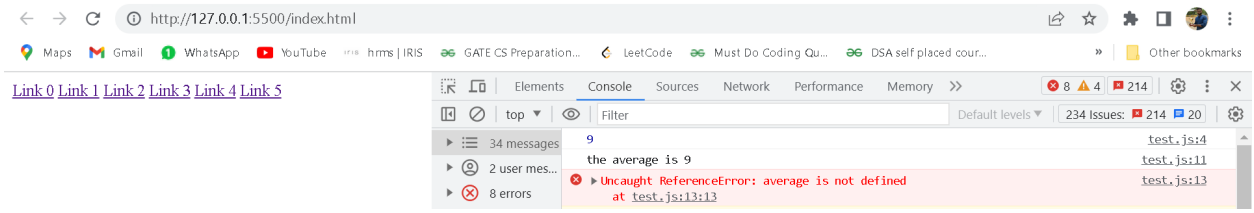

http://127.0.0.1:5500/index.html
Maps
Gmail
WhatsApp
YouTube
hrms | IRIS
GATE CS Preparation...
LeetCode
Must Do Coding Qu...
DSA self placed cour...
Other bookmarks
Link 0 Link 1 Link 2 Link 3 Link 4 Link 5
Elements
Console
Sources
Network
Performance
Memory
6 4 304
top
Filter
Default levels
324 Issues: 304 20
30 hidden
32 messages
2 user mes...
6 errors
18 warnings
2 info
6 verbose
9.5
test.js:4
the average is 9.5
test.js:11

21. Variable Scope

test.js

```
function getAverage(a, b) {

    var average = (a + b) / 2;        //local variable
    console.log(average);
    return average;

}


var myResult = getAverage(7, 11);        //global variable
console.log("the average is " + myResult);

console.log(average);
```

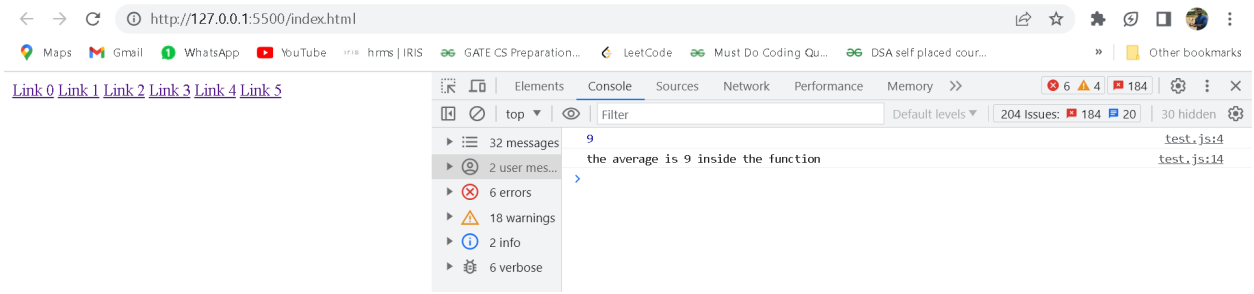
O/P:



test.js

```
function getAverage(a, b) {

    var average = (a + b) / 2;        //local variable
    console.log(average);
    return average;

}


var myResult = getAverage(7, 11);        //global variable

function logResult(){

    console.log("the average is " + myResult + " inside the function");

}

logResult();
```

O/P:



→Local variable is converted into global variable

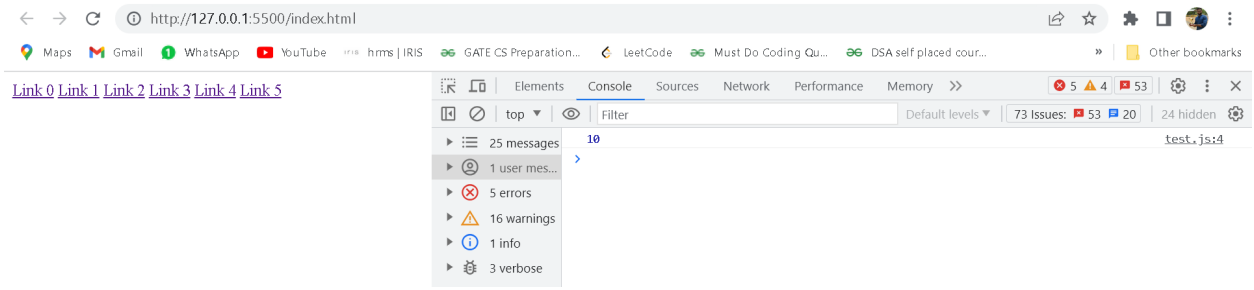
```
var average = 0;

function getAverage(a, b) {

    average = (a + b) / 2;        //global variable
    console.log(average);
    return average;

}


var myResult = getAverage(7, 11);        //global variable

function logResult(){

    console.log("the average is " + myResult + " inside the function");

}

logResult();
```

22. Numbers

test.js

```
var a = 5;
var b = 5;

console.log(a + b);
```

O/P:

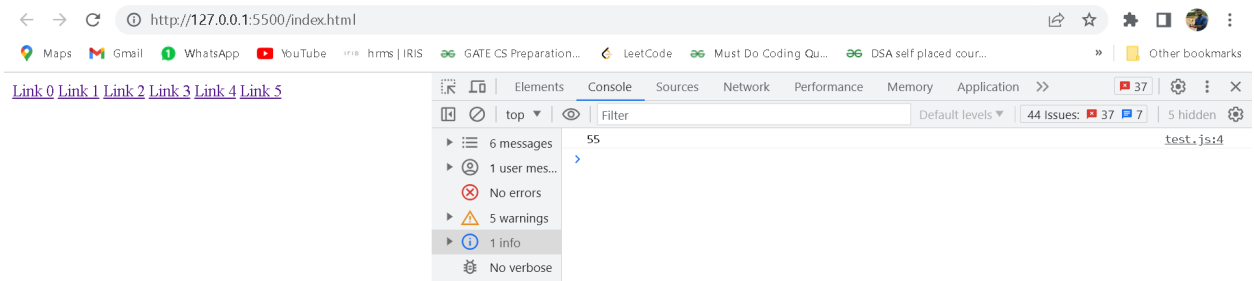


test.js

→Concatenation

```
var a = "5";
var b = 5;

console.log(a + b);
```

O/P:

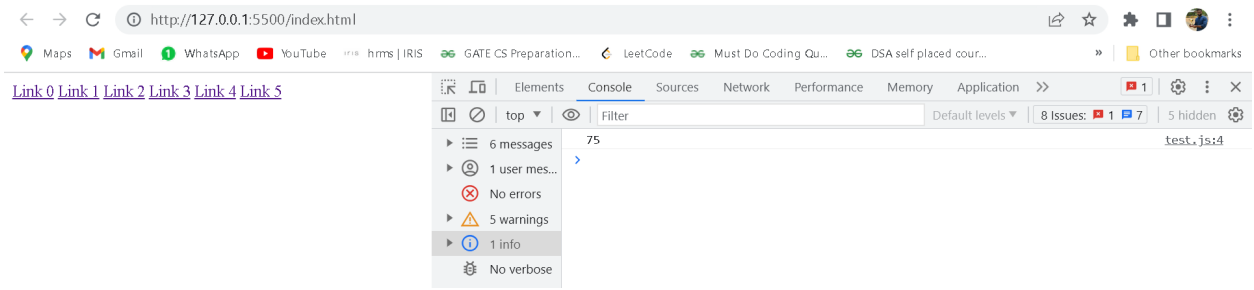


test.js

→Concatenation

```
var a = "7";
var b = 5;

console.log(a + b);
```

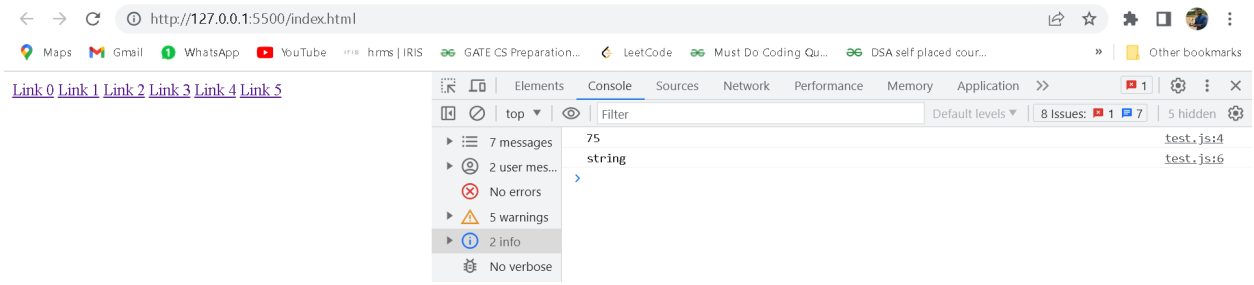
O/P:



test.js

```
var a = "7";
var b = 5;

console.log(a + b);

console.log(typeof (a + b));
```

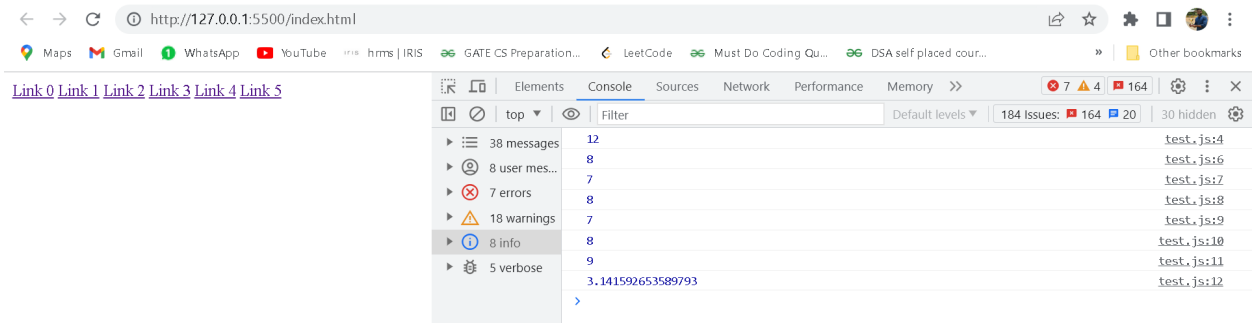
O/P:



test.js

```
var a = 7;
var b = 5;

console.log(a + b);

console.log(Math.round(7.5));
console.log(Math.round(7.3));
console.log(Math.round(7.8));
console.log(Math.floor(7.9));
console.log(Math.ceil(7.2));
console.log(Math.max(7, 4, 9, 8));
console.log(Math.PI)
```

O/P:

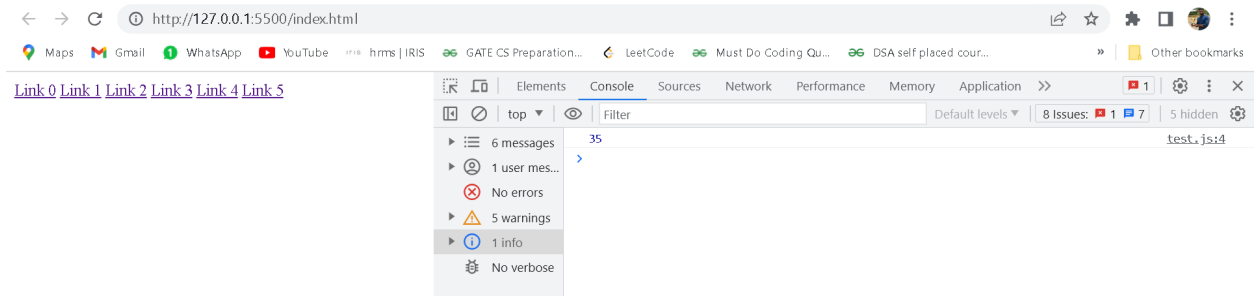


23. NaN(Not a Number)

test.js

```
var a = "7";
var b = 5;

console.log(a * b);
```

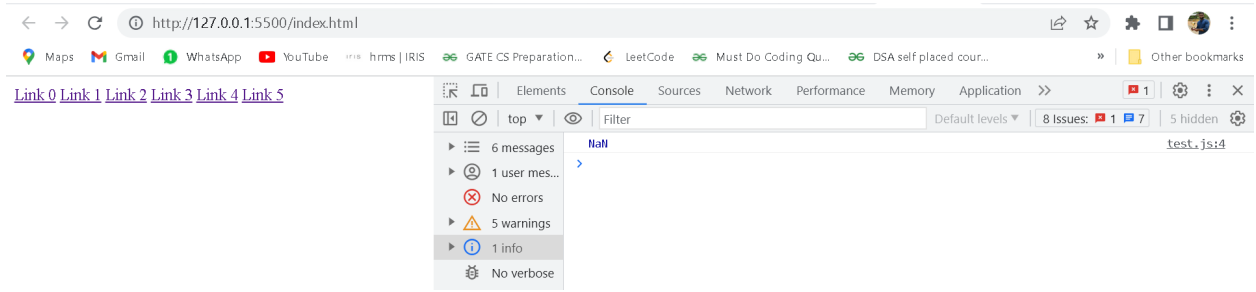
O/P:



test.js

```
var a = "apple";
var b = 5;

console.log(a * b);
```

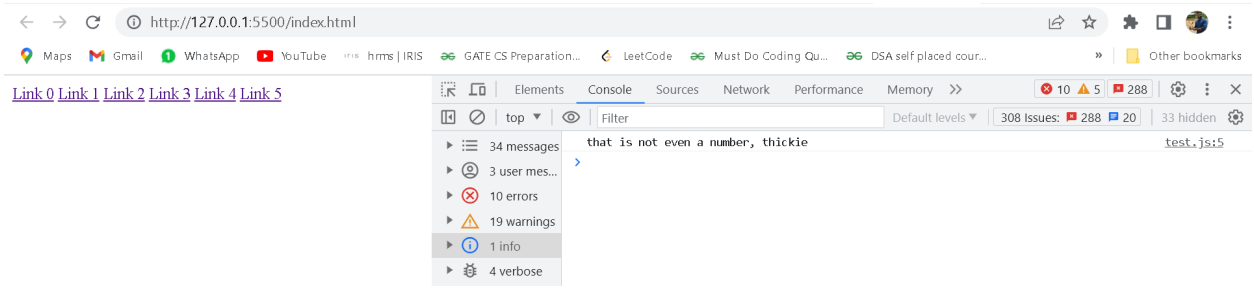
O/P:



test.js

```
var a = "apple";
var b = 5;

if (isNaN(a)){
    console.log("that is not even a number, thickie");
}
else{
    console.log("meaning of life is " + (a * b));
}
```

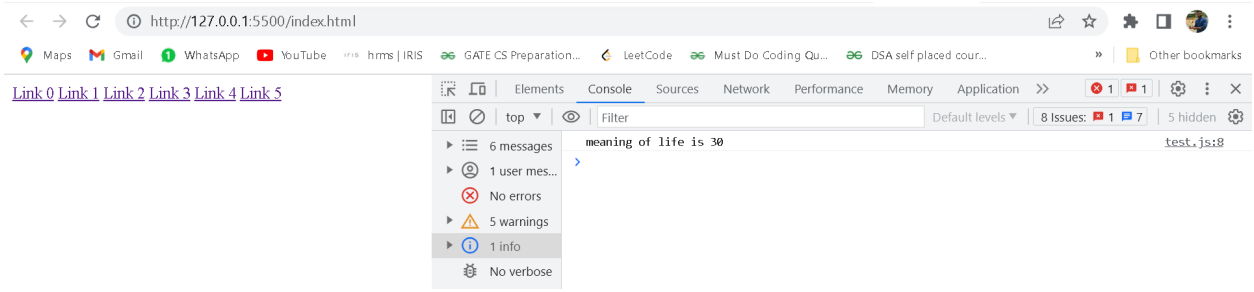
O/P:



test.js

```
var a = 6;
var b = 5;

if (isNaN(a)){
    console.log("that is not even a number, thickie");
}
else{
    console.log("meaning of life is " + (a * b));
}
```

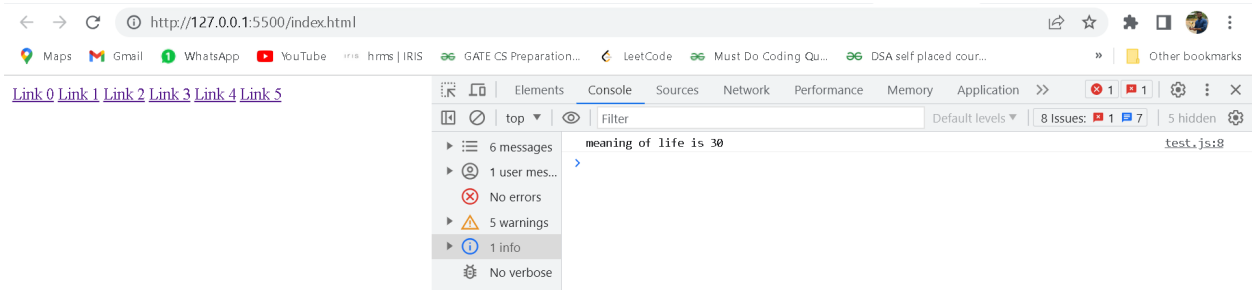
O/P:



test.js

```
var a = 6;
var b = 5;

//double negative
if (!isNaN(a)){
    console.log("meaning of life is " + (a * b));
}
```

O/P:

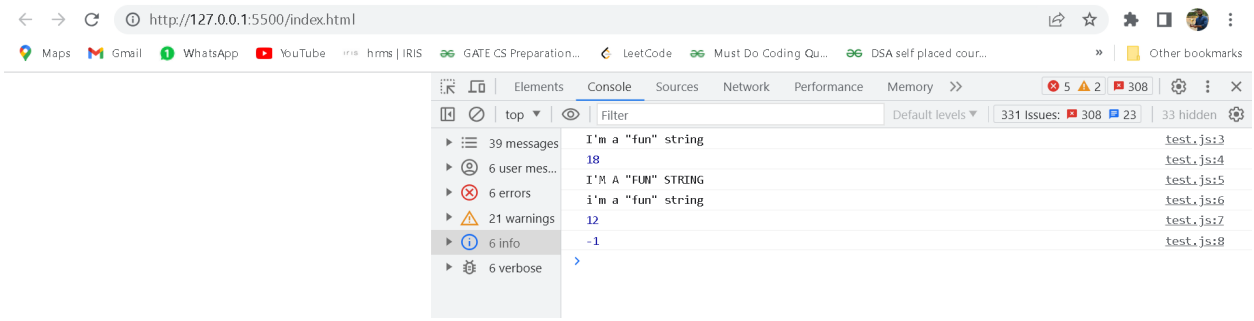


24. Strings

test.js

```
var myString = 'I\'m a "fun" string';

console.log(myString);
console.log(myString.length);
console.log(myString.toUpperCase());
console.log(myString.toLowerCase());
console.log(myString.indexOf("string"));
console.log(myString.indexOf("ninja"));    //not found
```

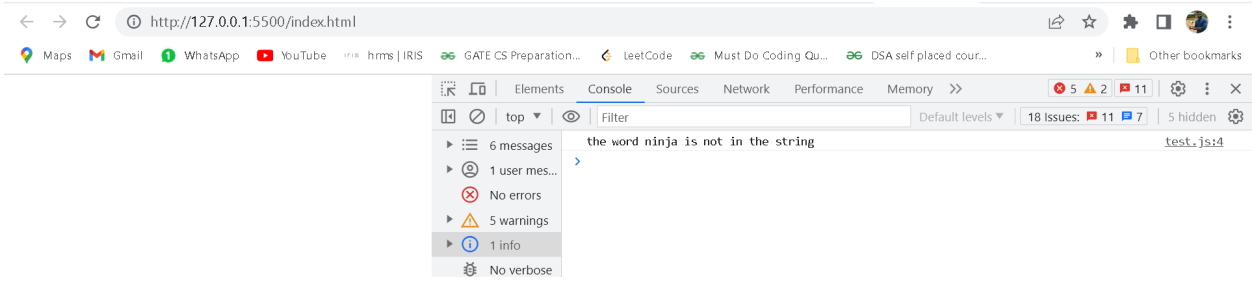
O/P:



test.js

```
var myString = 'I\'m a "fun" string';

if (myString.indexOf("ninja") === -1){
    console.log("the word ninja is not in the string");
} else {
    console.log("the word ninja starts at position " +
myString.indexOf("ninja"));
}
```

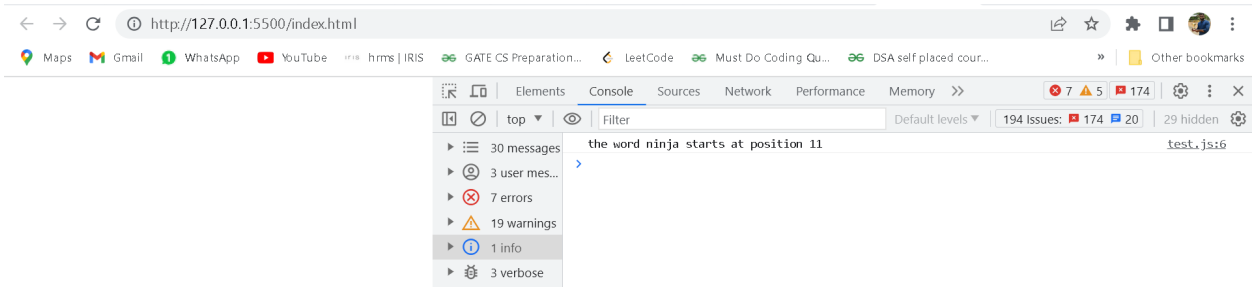
O/P:



test.js

```
var myString = 'I\'m a "fun ninja" string';

if (myString.indexOf("ninja") === -1){
    console.log("the word ninja is not in the string");
} else {
    console.log("the word ninja starts at position " +
myString.indexOf("ninja"));
}
```

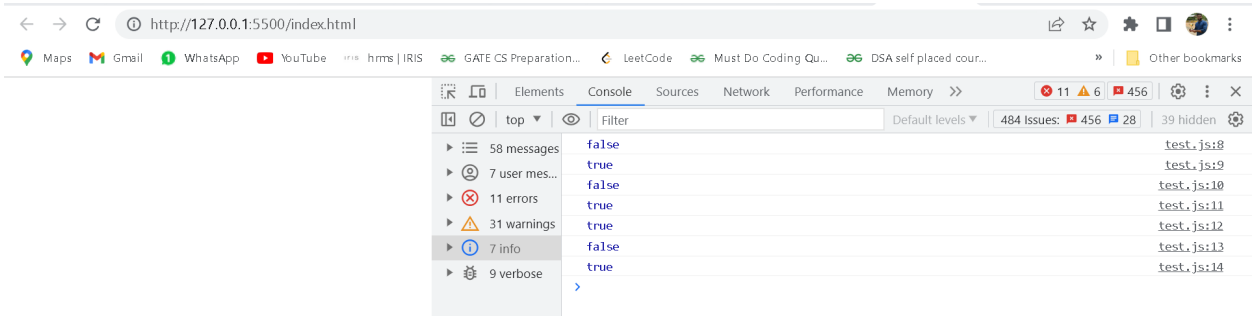
O/P:



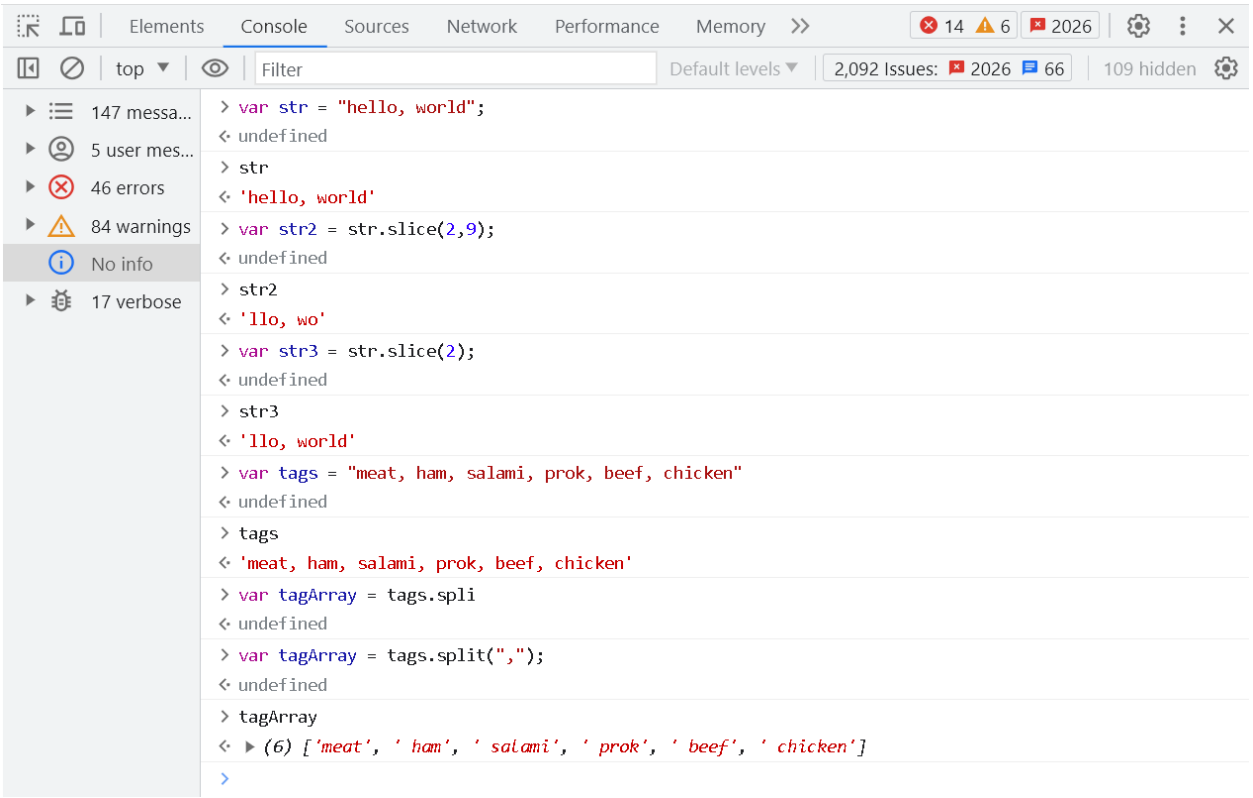
test.js

```
var string1 = "abc";
var string2 = "bcd";
var string3 = "abc";
var string4 = "ABC";
var string5 = "Bcd";
var string6 = "Abc";

console.log(string1 == string2);
console.log(string1 == string3);
console.log(string1 == string4);
console.log(string1.toLowerCase() == string4.toLowerCase());
console.log(string1 < string2);
console.log(string1 < string5);
console.log(string6 < string5);
```

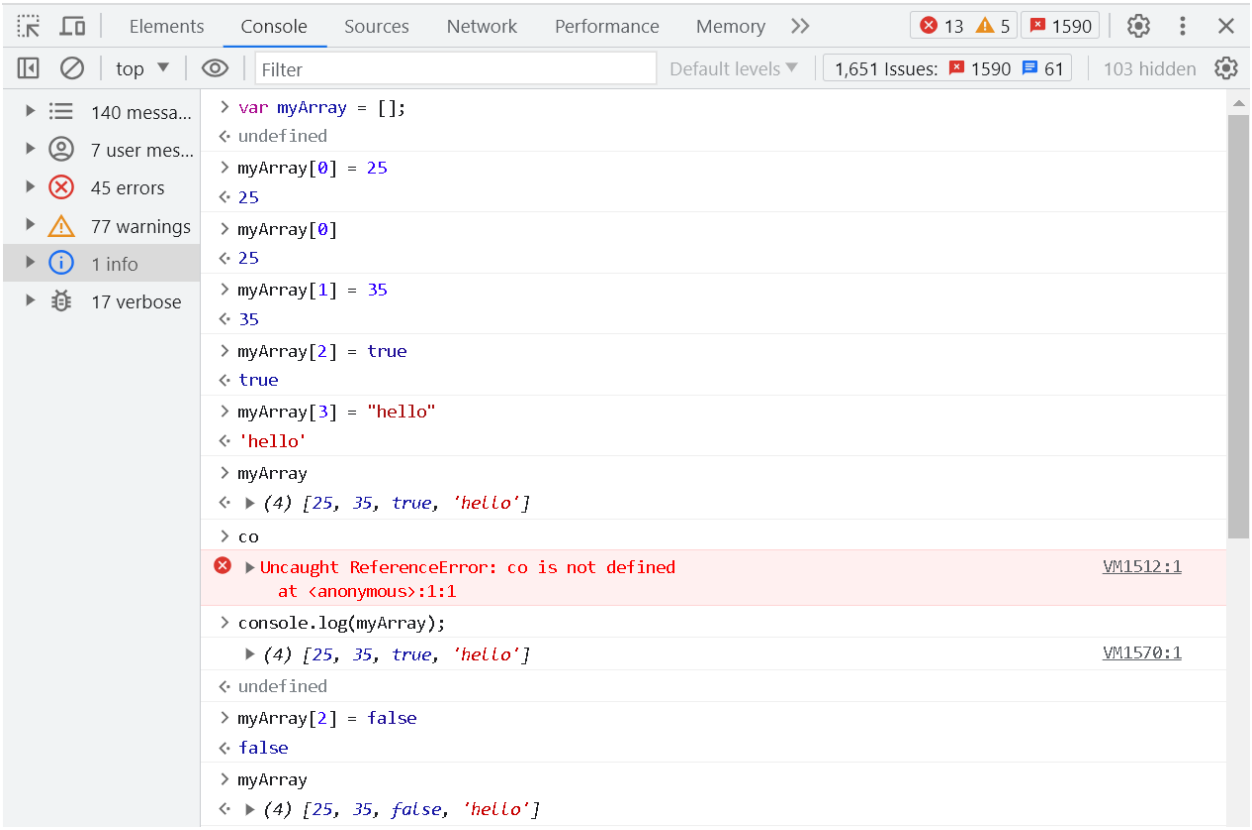
O/P:



25. Slice & Split Strings

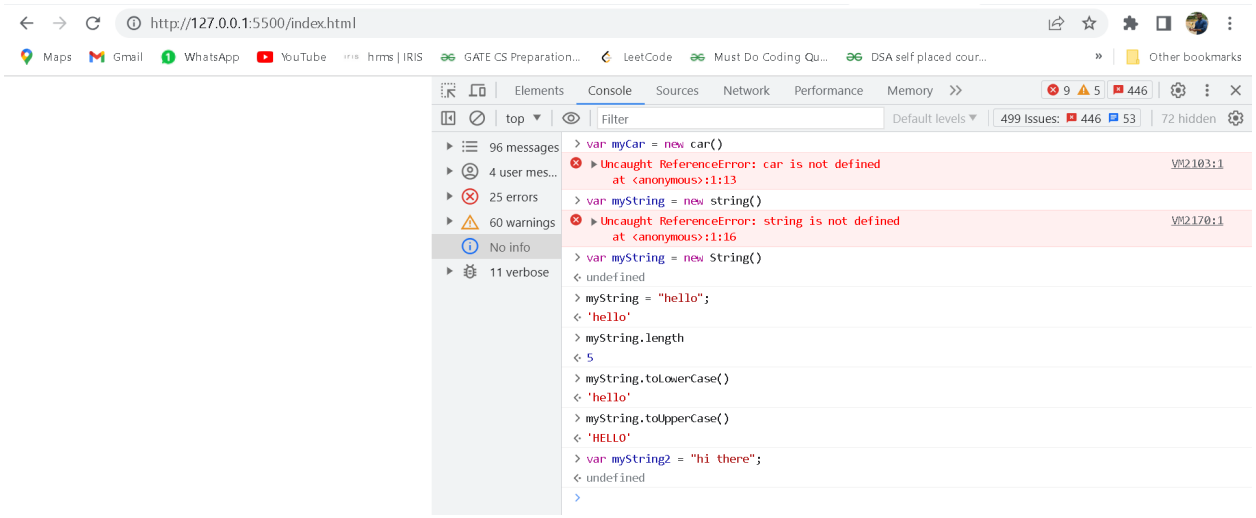


26. Arrays


Elements
Console
Sources
Network
Performance
Memory
13 5 1590
top
Filter
Default levels
1,651 Issues: 1590 61
103 hidden
140 messa...
7 user mes...
45 errors
77 warnings
1 info
17 verbose
var myArray = [];
undefined
myArray[0] = 25
25
myArray[0]
25
myArray[1] = 35
35
myArray[2] = true
true
myArray[3] = "hello"
'hello'
myArray
(4) [25, 35, true, 'hello']
co
Uncaught ReferenceError: co is not defined
at <anonymous>:1:1
VM1512:1
console.log(myArray);
(4) [25, 35, true, 'hello']
VM1570:1
undefined
myArray[2] = false
false
myArray
(4) [25, 35, false, 'hello']

```
> myArray
<· ▶ (4) [25, 35, false, 'hello']
> var myArray2 = [10, 20, "hi", false];
<· undefined
> myArray2
<· ▶ (4) [10, 20, 'hi', false]
> var myArray3 = new Array()
<· undefined
> var myArray4 = new Array(5)
<· undefined
> myArray2.length
<· 4
> myArray3.length
<· 0
> myArray2.sort()
<· ▶ (4) [10, 20, false, 'hi']
> myArray2.reverse()
<· ▶ (4) ['hi', false, 20, 10]
>
```

27. Introduction to Objects

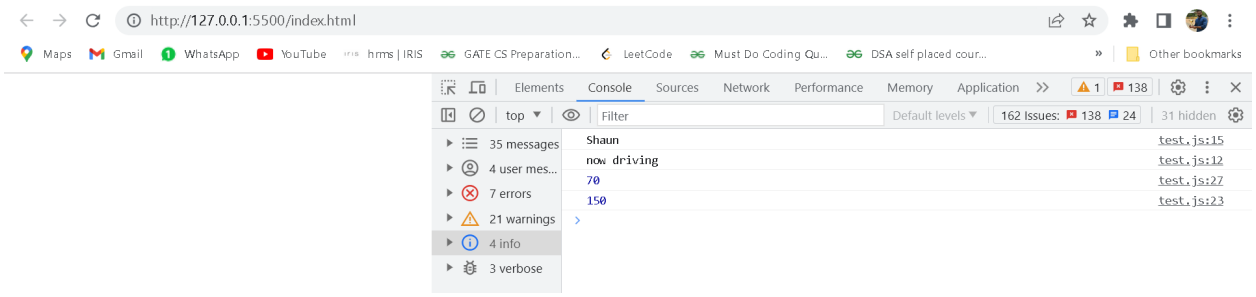


28. Creating a new JavaScript Object

test.js

```
var myArray = new Array();
myArray[0] = 8;
myArray[1] = "hello";

var myCar = new Object();
//properties of myCar object
myCar.maxSpeed = 50;
myCar.driver = "Shaun";

//methods
myCar.drive = function(){
    console.log("now driving");
};

console.log(myCar.driver);

myCar.drive();

var myCar2 = {
    maxSpeed: 70,
    driver: "Net Ninja",
    drive: function(speed, time){
        console.log(speed * time);
    }
};

console.log(myCar2.maxSpeed);
myCar2.drive(50, 3);
```

O/P:

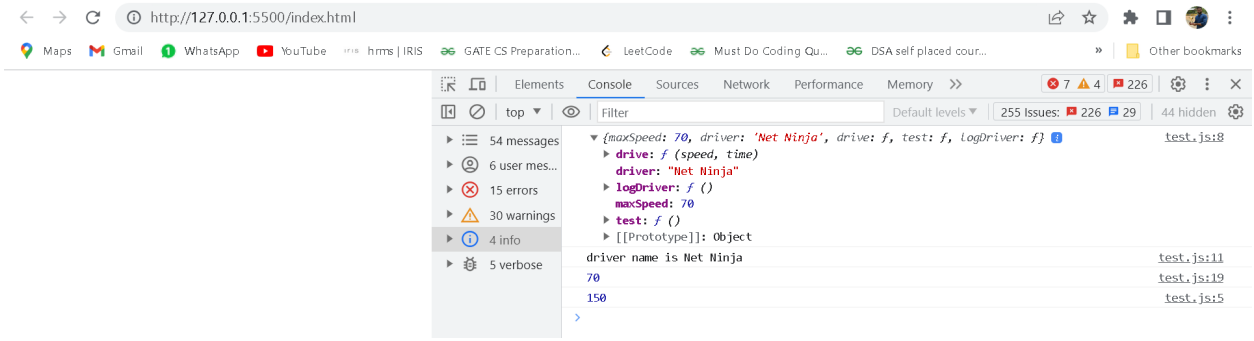


29. This KeyWord

test.js

```
var myCar2 = {
    maxSpeed: 70,
    driver: "Net Ninja",
    drive: function(speed, time){
        console.log(speed * time);
    },
    test: function(){
        console.log(this);
    },
    logDriver: function(){
        console.log("driver name is " + this.driver);
    }
};


myCar2.test();
myCar2.logDriver();
console.log(myCar2.maxSpeed);
myCar2.drive(50, 3);
```

O/P:

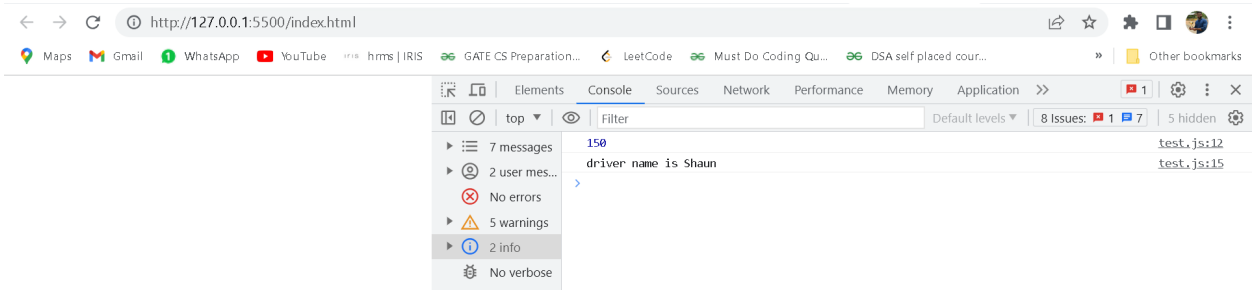

http://127.0.0.1:5500/index.html
Maps
Gmail
WhatsApp
YouTube
hrms | IRIS
GATE CS Preparation...
LeetCode
Must Do Coding Qu...
DSA self placed cour...
Other bookmarks
Elements
Console
Sources
Network
Performance
Memory
7 4 226
top
Filter
Default levels
255 Issues: 226 29
44 hidden
54 messages
6 user mes...
15 errors
30 warnings
4 info
5 verbose
{maxSpeed: 70, driver: 'Net Ninja', drive: ƒ, test: ƒ, logDriver: ƒ}
test.js:8
drive: ƒ (speed, time)
driver: "Net Ninja"
logDriver: ƒ ()
maxSpeed: 70
test: ƒ ()
[[Prototype]]: Object
driver name is Net Ninja
test.js:11
70
test.js:19
150
test.js:5

30. Constructor Functions

test.js

```
//Example of constructors
// var myArray = new Array();
// var myString = new String();
// var myCar = new Car();

//constructor function
var Car = function(maxSpeed, driver){

    this.maxSpeed = maxSpeed;
    this.driver = driver;
    this.drive = function(speed, time){
        console.log(speed * time);
    };
    this.logDriver = function(){
        console.log("driver name is " + this.driver);
    };

}

//calling constructor function
var myCar = new Car(70, "Net Ninja");
var myCar2 = new Car(40, "Humpy dumpty");
var myCar3 = new Car(10, "Shaun");
var myCar4 = new Car(90, "James Bond");

myCar.drive(30, 5);
myCar3.logDriver();
```

O/P:

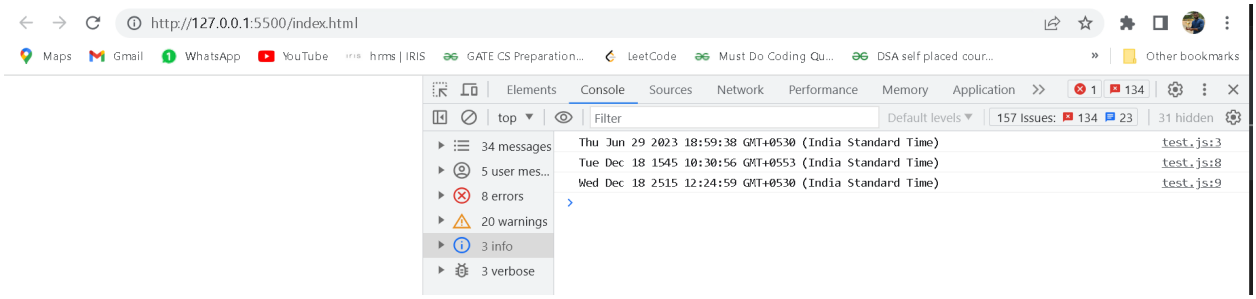


31. Date Object

test.js

```
//Inbuilt object date
var myDate = new Date();
console.log(myDate);

var myPastDate = new Date(1545, 11, 18, 10, 30, 56);
var myFutureDate = new Date(2515, 11, 18, 12, 24, 59);

console.log(myPastDate);
console.log(myFutureDate);
```

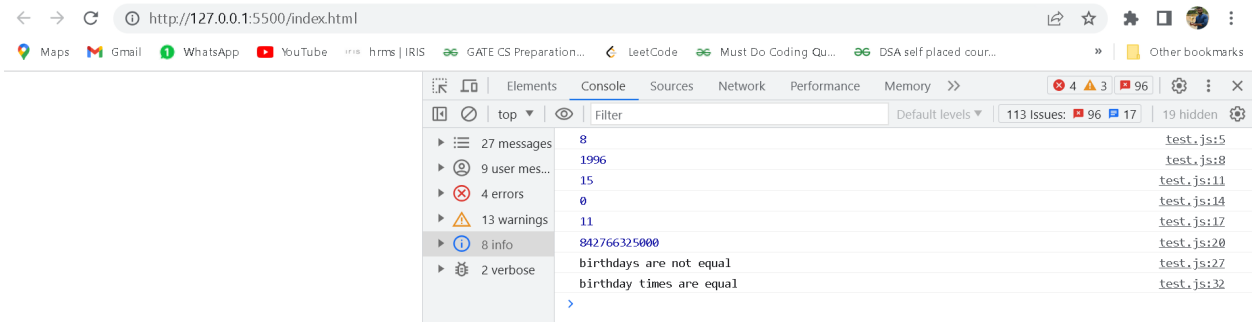
O/P:



test.js

```
var birthday = new Date(1996, 8, 15, 11, 15, 25);
var birthday2 = new Date(1996, 8, 15, 11, 15, 25);

//get the month of the date (0 - 11)
console.log(birthday.getMonth());

//get the full year (YYYY)
console.log(birthday.getFullYear());

//get the date of the month (1 - 31)
console.log(birthday.getDate());

//get the day of the week (0 - 6)  0 ->sun 1 -> mon
console.log(birthday.getDay());

//get the hour of the date (0 - 23)
console.log(birthday.getHours());

//get the number of milliseconds since 1st Jan 1970
console.log(birthday.getTime());

if(birthday == birthday2){  //both objects are not same

    console.log("birthdays are equal");
} else {

    console.log("birthdays are not equal");
}

if(birthday.getTime() == birthday2.getTime()){

    console.log("birthday times are equal");
} else {

    console.log("birthdays times are not equal");
}
```

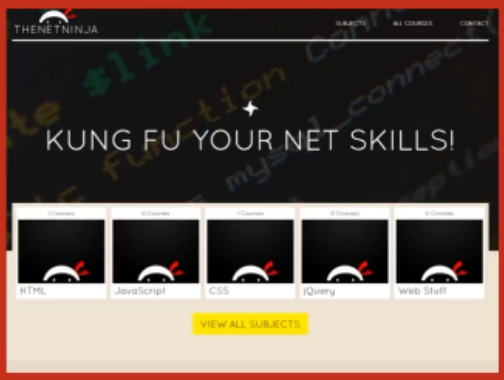
O/P:

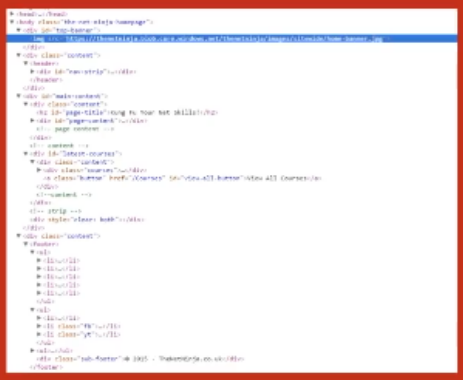


32. What is the DOM in JavaScript?

DOCUMENT

- The document is just the web page



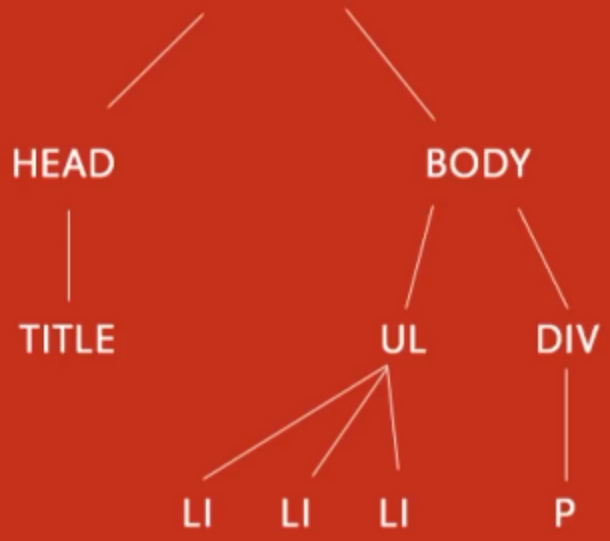


Objects Are Elements

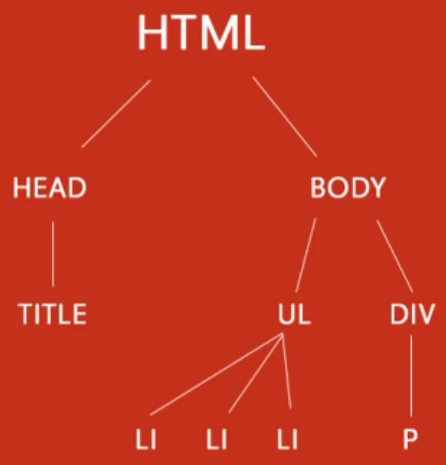
Every HTML element in the document is an object:

- <head></head> is an object
- <body></body> is an object
- is an object
- is an object

The Model



NODES

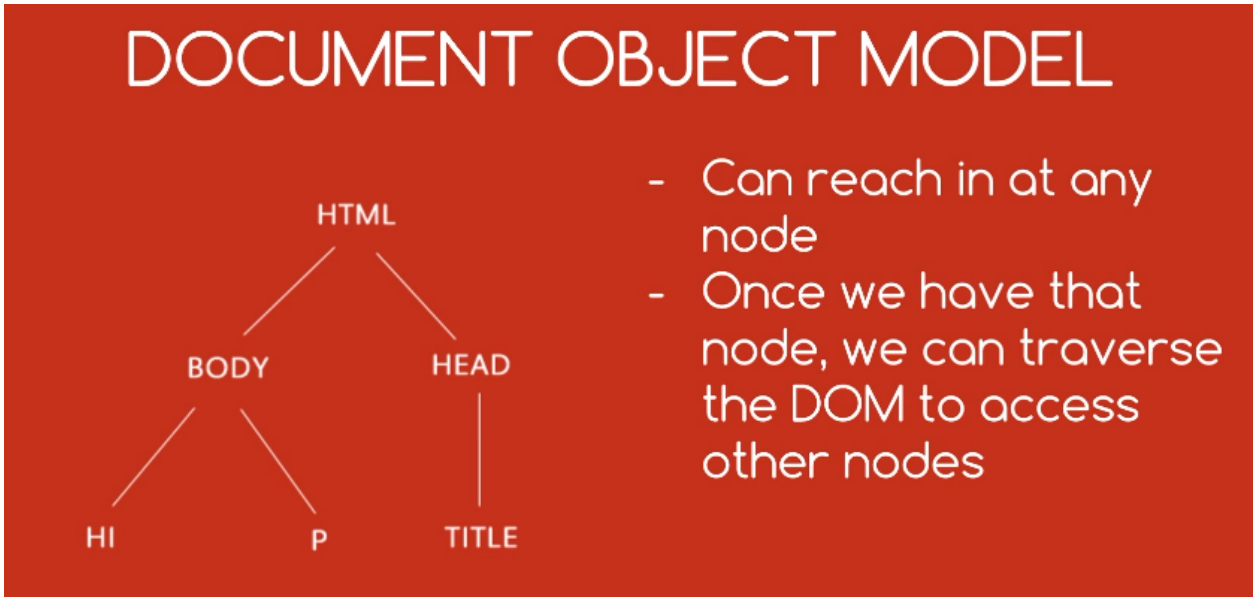


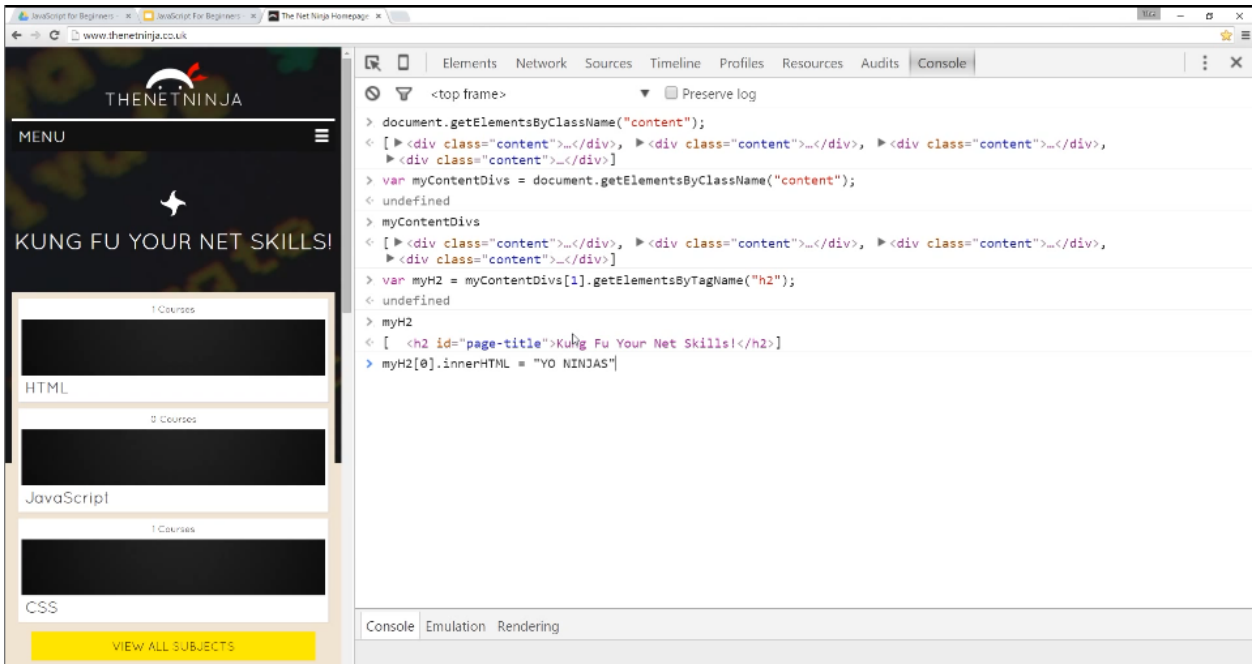
- Everything we can change in the document is a node:
  - Elements
  - Text within elements
  - HTML attributes

What Can We Do With The DOM?

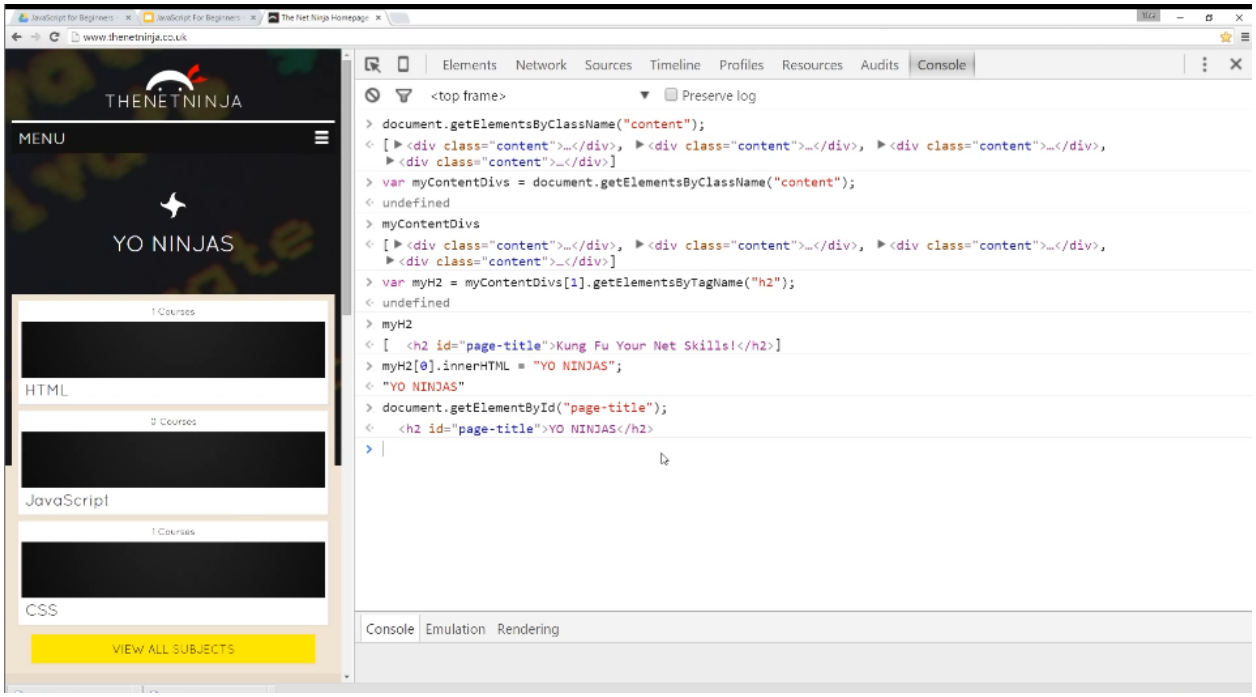
- Change the <h1> text node at the top of the page
- Change the background colour of an element node
- Animate the logo node from left to right
- Expand the height of an element node when you click on it
- PLUS MUCH MORE...

33. Traversing the DOM

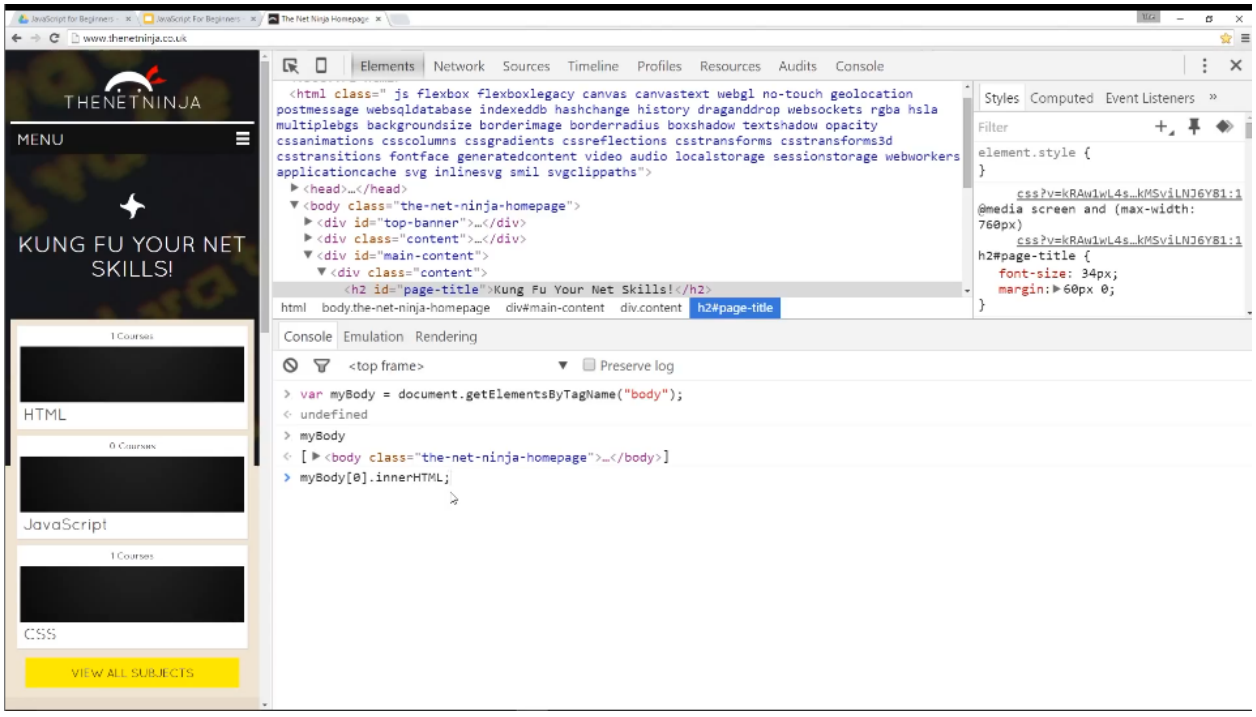




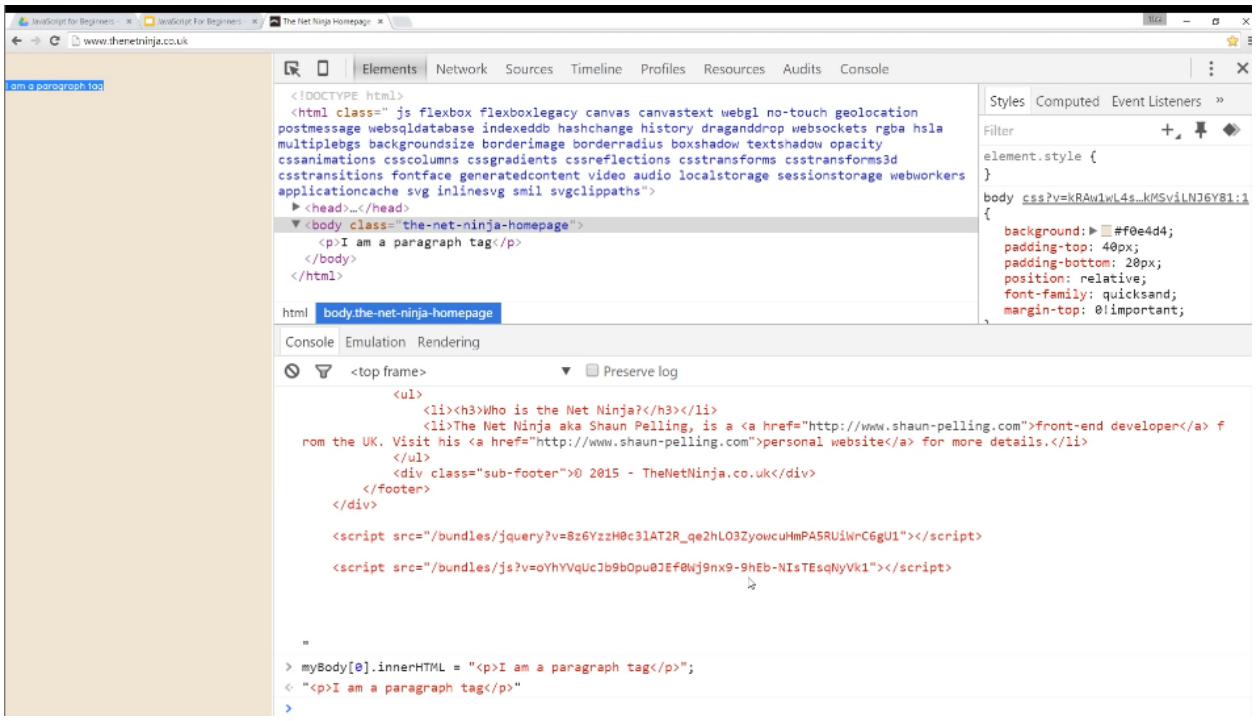
→Change the H2 title name

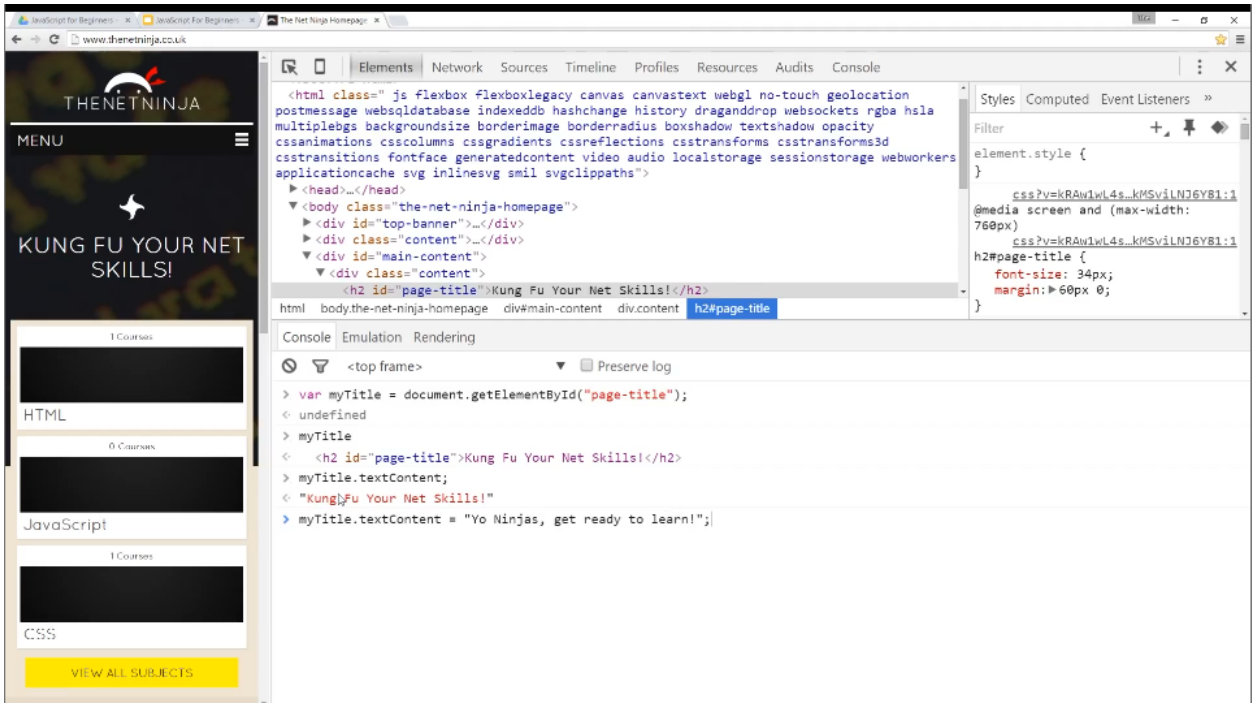


34. Changing Page Content

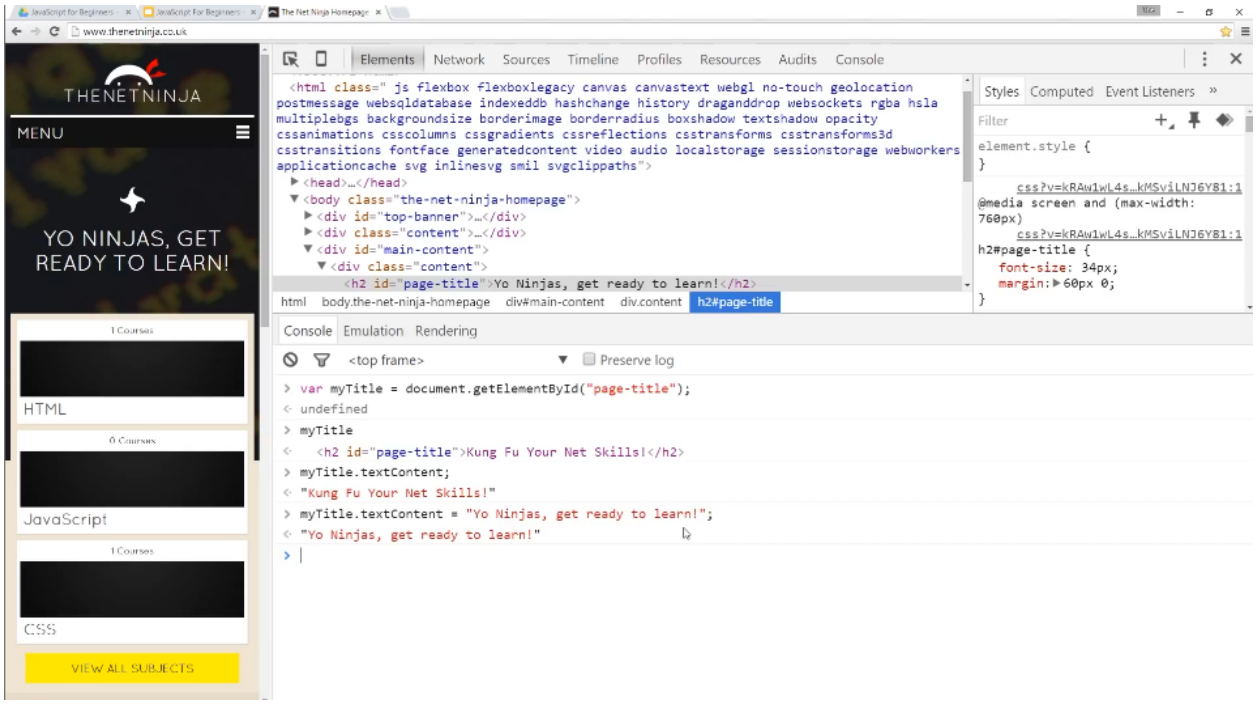


→How an element from HTML is changed/ Text on HTML Page is Changed

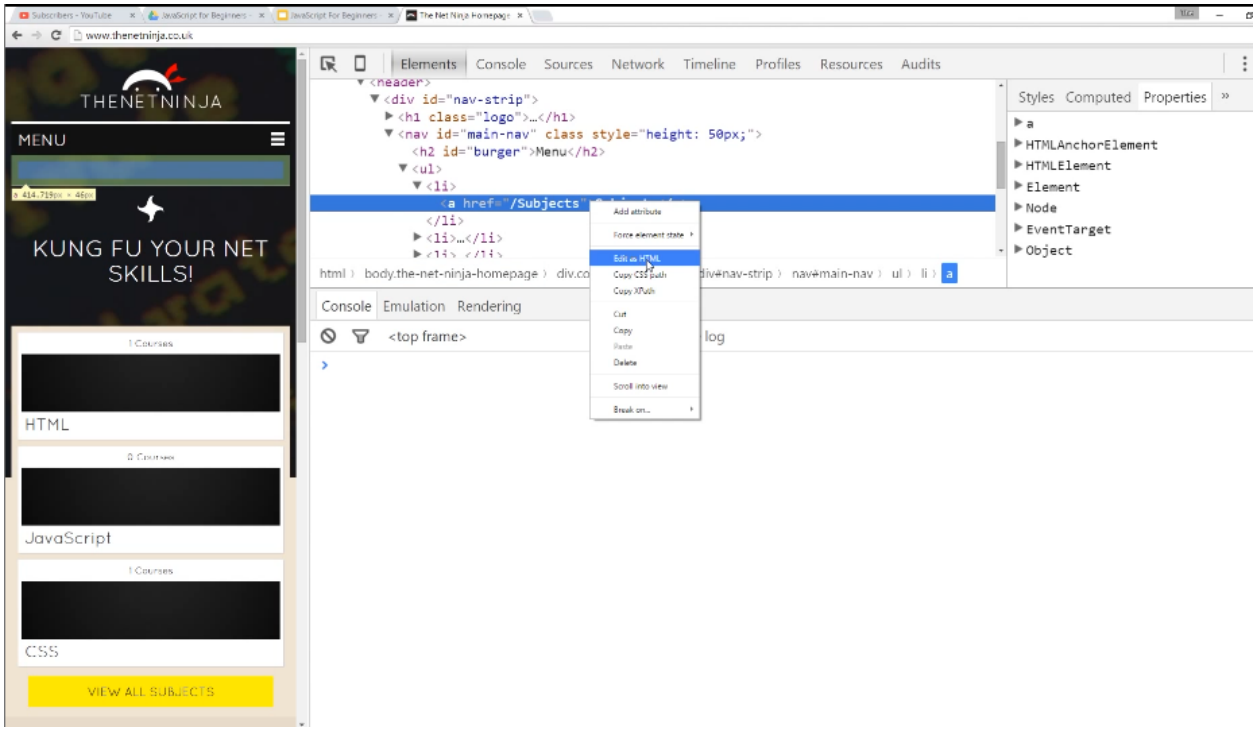


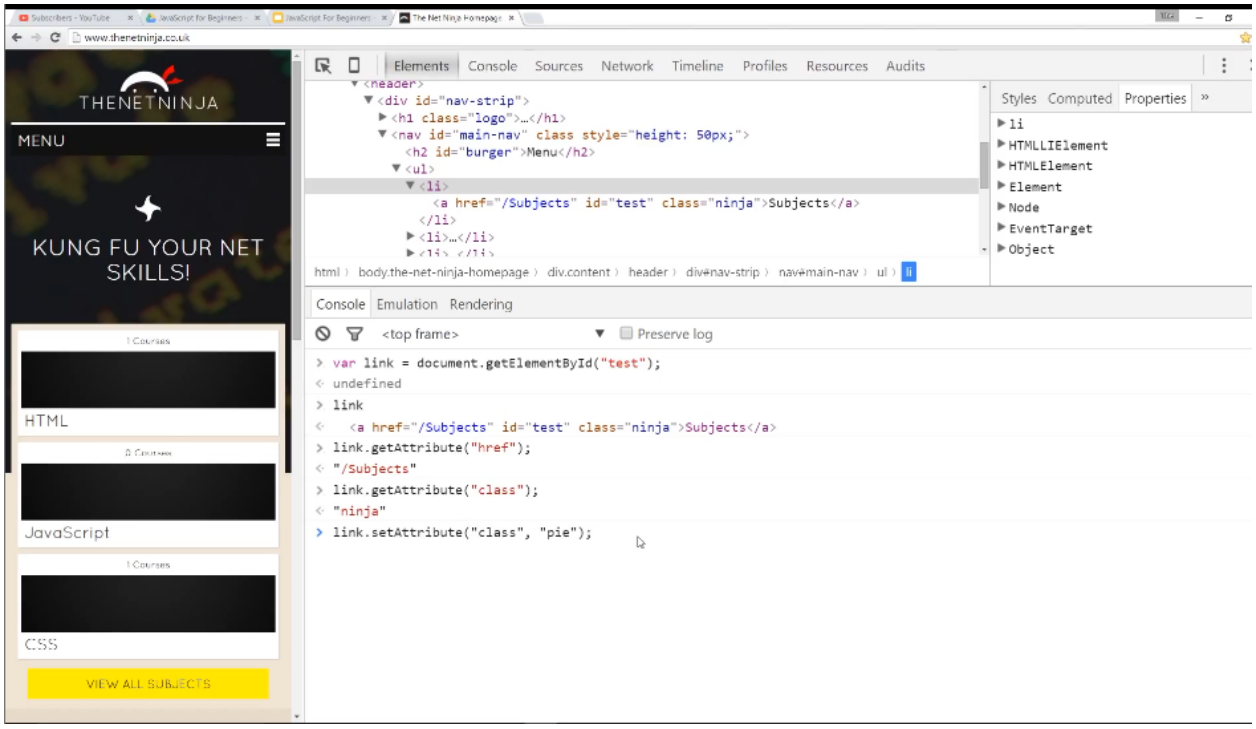


→Changed the “Title Text”

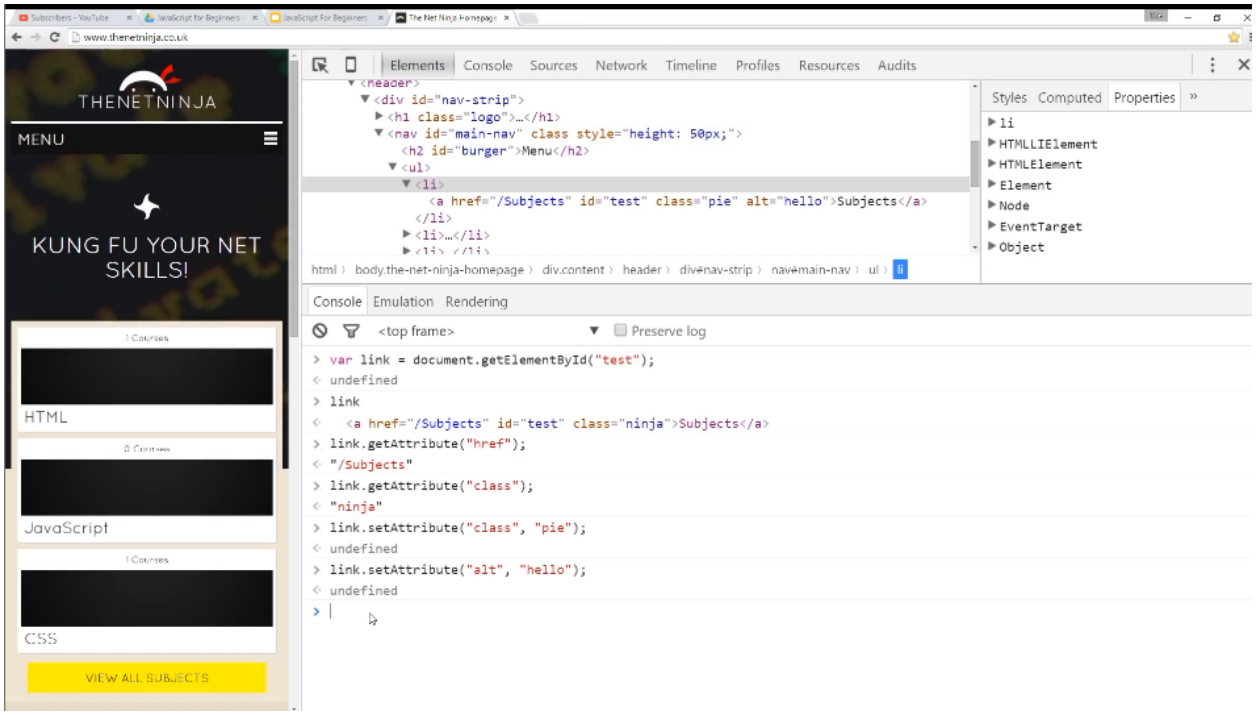


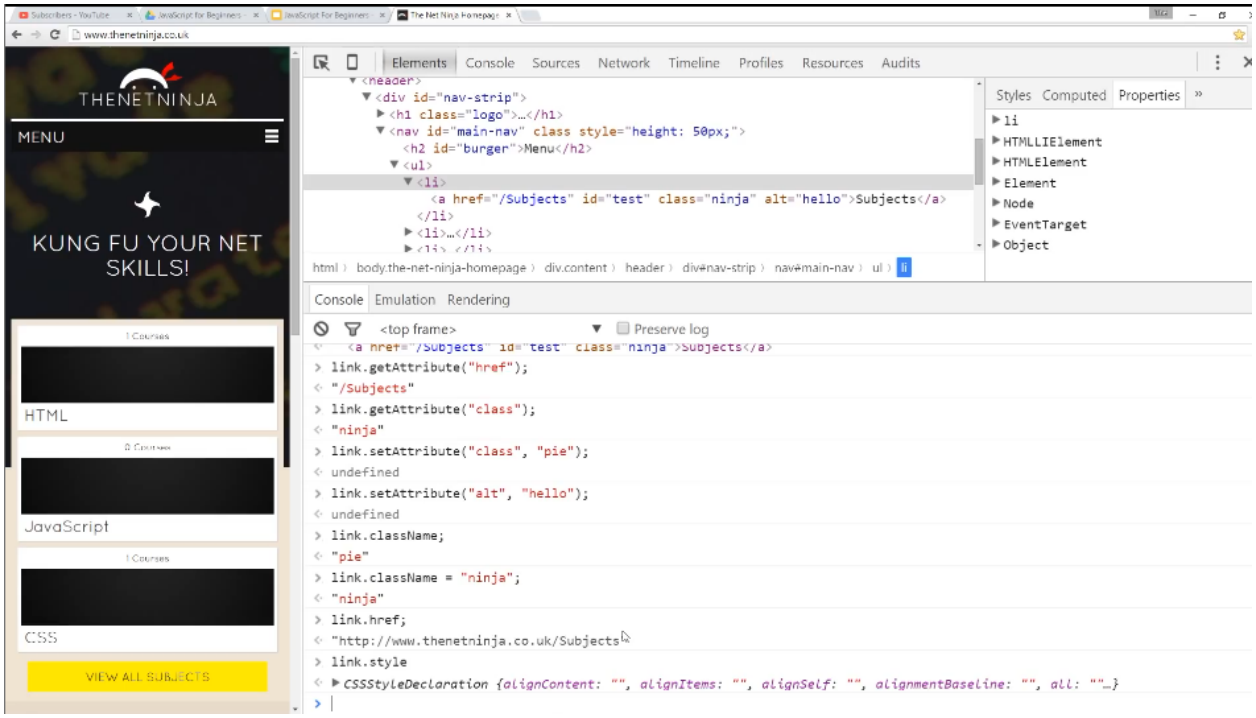
35. Changing Element Attributes



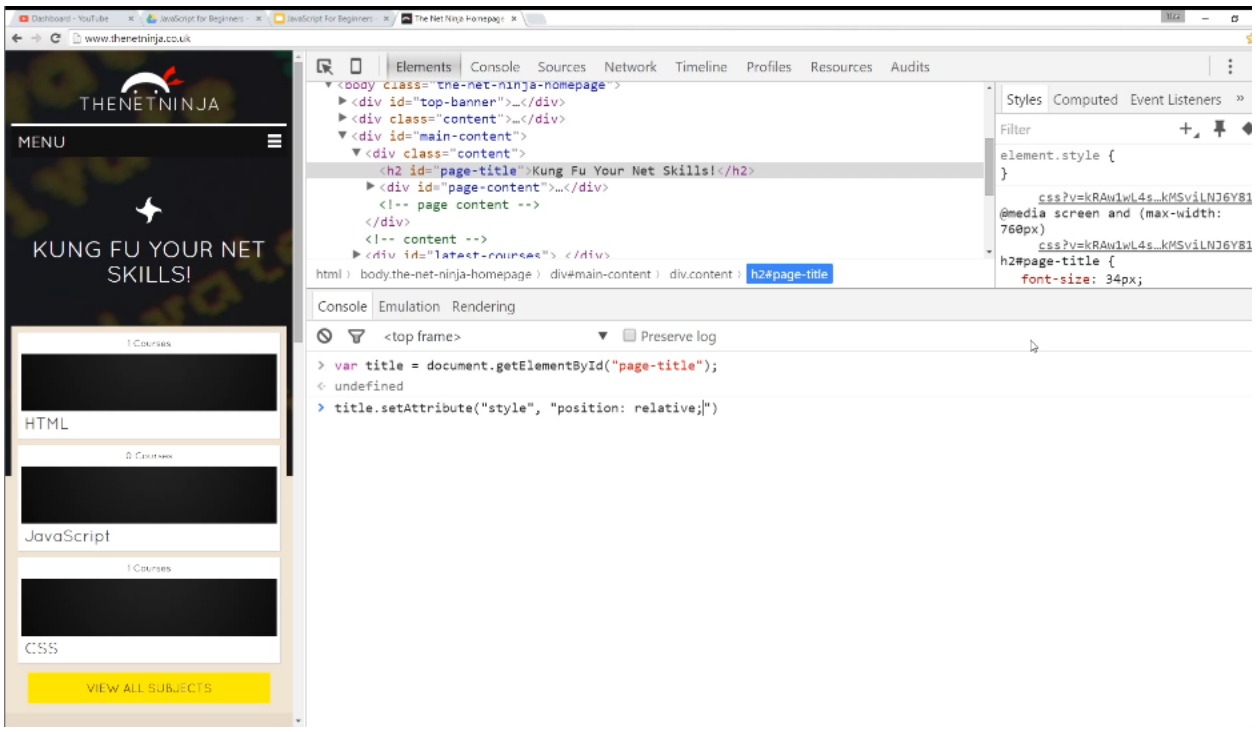


→Changing the “class” & adding the “alt” attribute

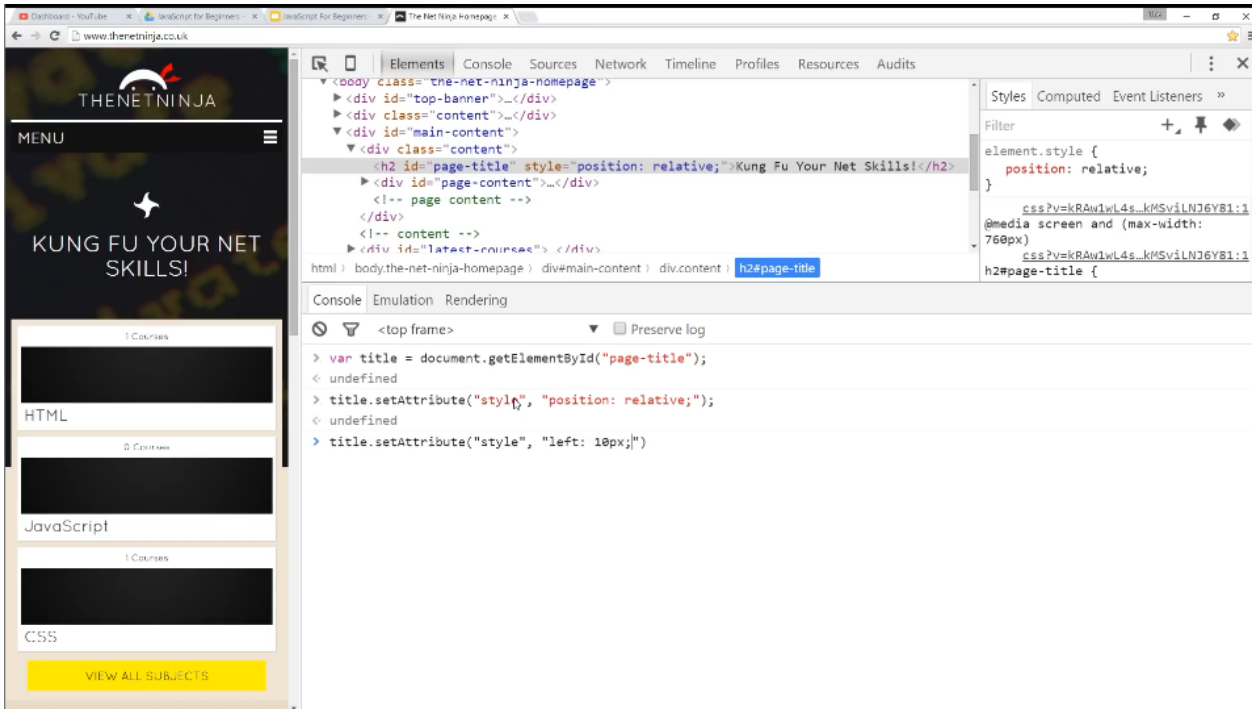


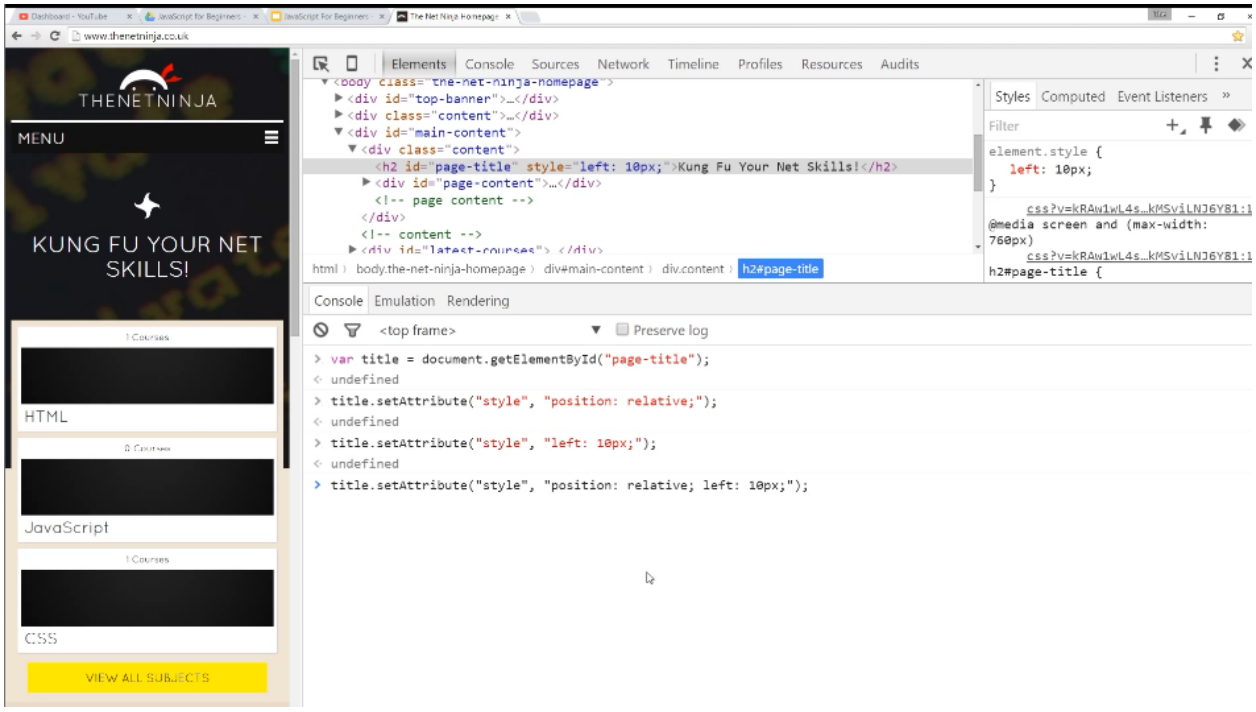


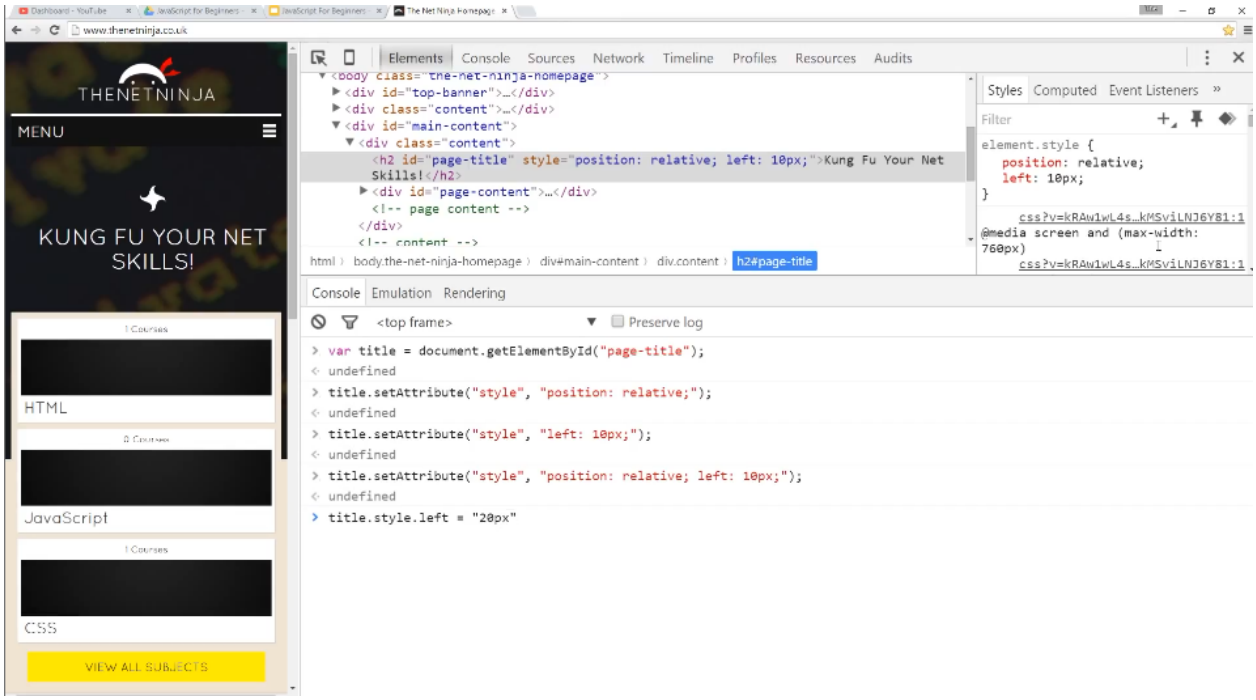
36. Changing CSS Styles

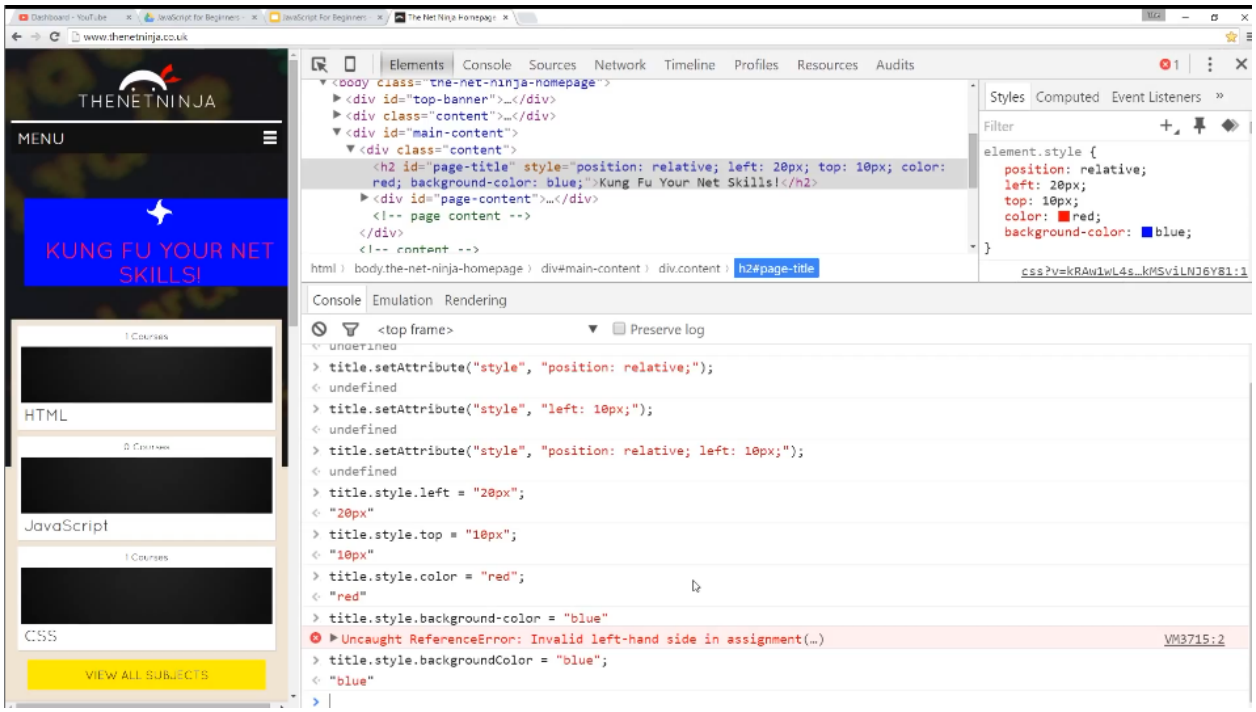


→Updating Style



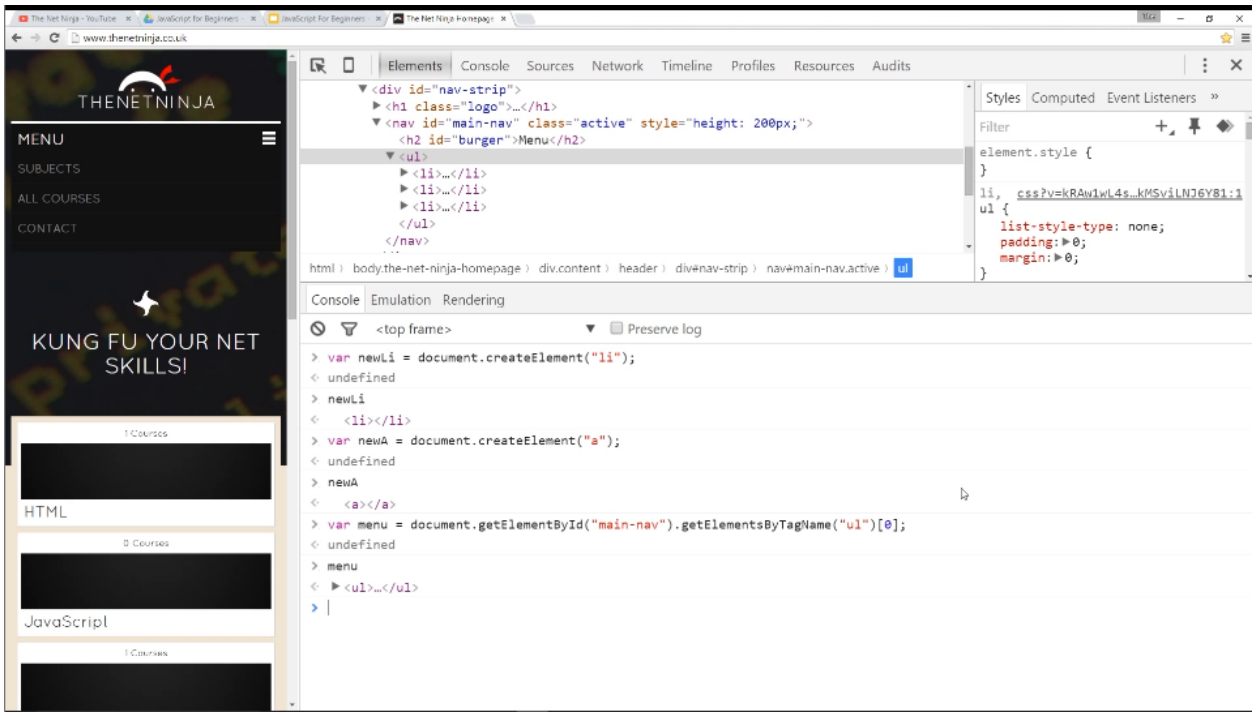



THENETNINJA
MENU
KUNG FU YOUR NET SKILLS!
HTML
JavaScript
CSS
VIEW ALL SUBJECTS
Elements Console Sources Network Timeline Profiles Resources Audits
<h2 id="page-title" style="position: relative; left: 10px;">Kung Fu Your Net Skills!</h2>
Styles Computed Event Listeners
element.style {
position: relative;
left: 10px;
}
@media screen and (max-width: 760px)
Console Emulation Rendering
<top frame>
Preserve log
var title = document.getElementById("page-title");
undefined
title.setAttribute("style", "position: relative;");
undefined
title.setAttribute("style", "left: 10px;");
undefined
title.setAttribute("style", "position: relative; left: 10px;");
undefined
title.style.left = "20px"

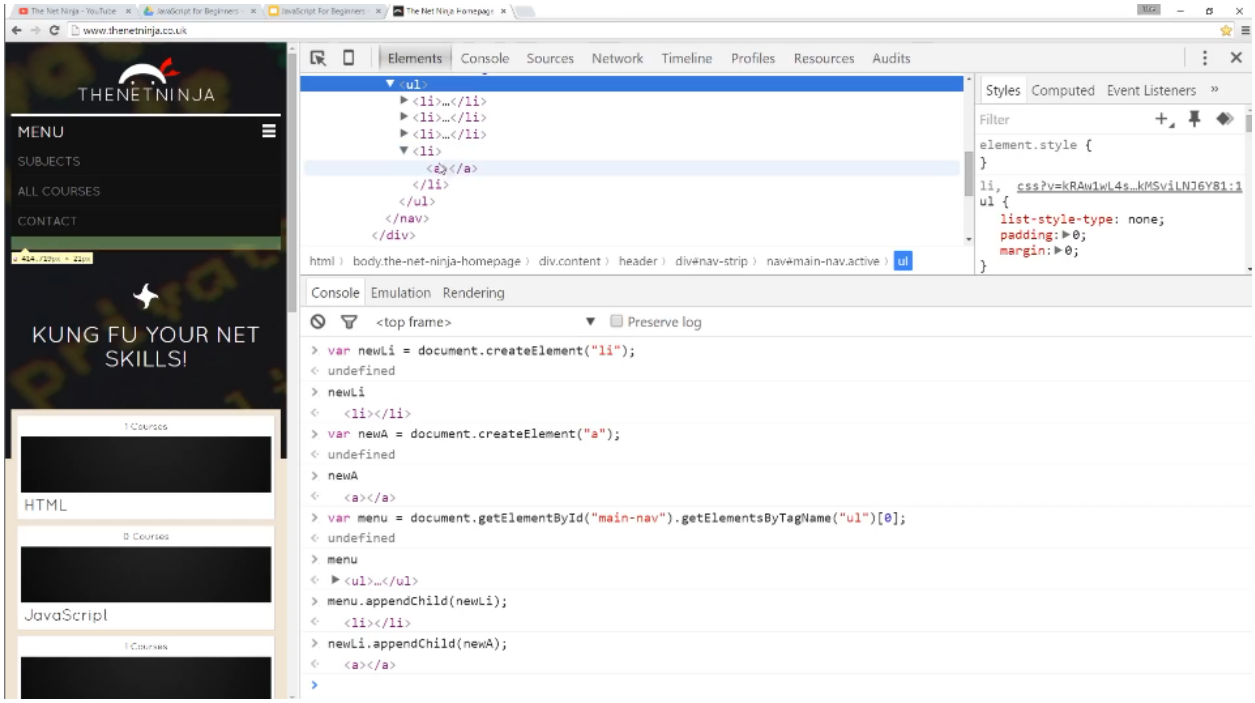

THENETNINJA
MENU
KUNG FU YOUR NET SKILLS!
HTML
JavaScript
CSS
VIEW ALL SUBJECTS
Elements Console Sources Network Timeline Profiles Resources Audits
<h2 id="page-title" style="position: relative; left: 20px; top: 10px; color: red; background-color: blue;">Kung Fu Your Net Skills!</h2>
Styles Computed Event Listeners
element.style {
position: relative;
left: 20px;
top: 10px;
color: red;
background-color: blue;
}
Console Emulation Rendering
<top frame>
Preserve log
title.setAttribute("style", "position: relative;");
undefined
title.setAttribute("style", "left: 10px;");
undefined
title.setAttribute("style", "position: relative; left: 10px;");
undefined
title.style.left = "20px";
"20px"
title.style.top = "10px";
"10px"
title.style.color = "red";
"red"
title.style.background-color = "blue"
Uncaught ReferenceError: Invalid left-hand side in assignment(…)
VM3715:2
title.style.backgroundColor = "blue";
"blue"

37. Adding Elements to the DOM

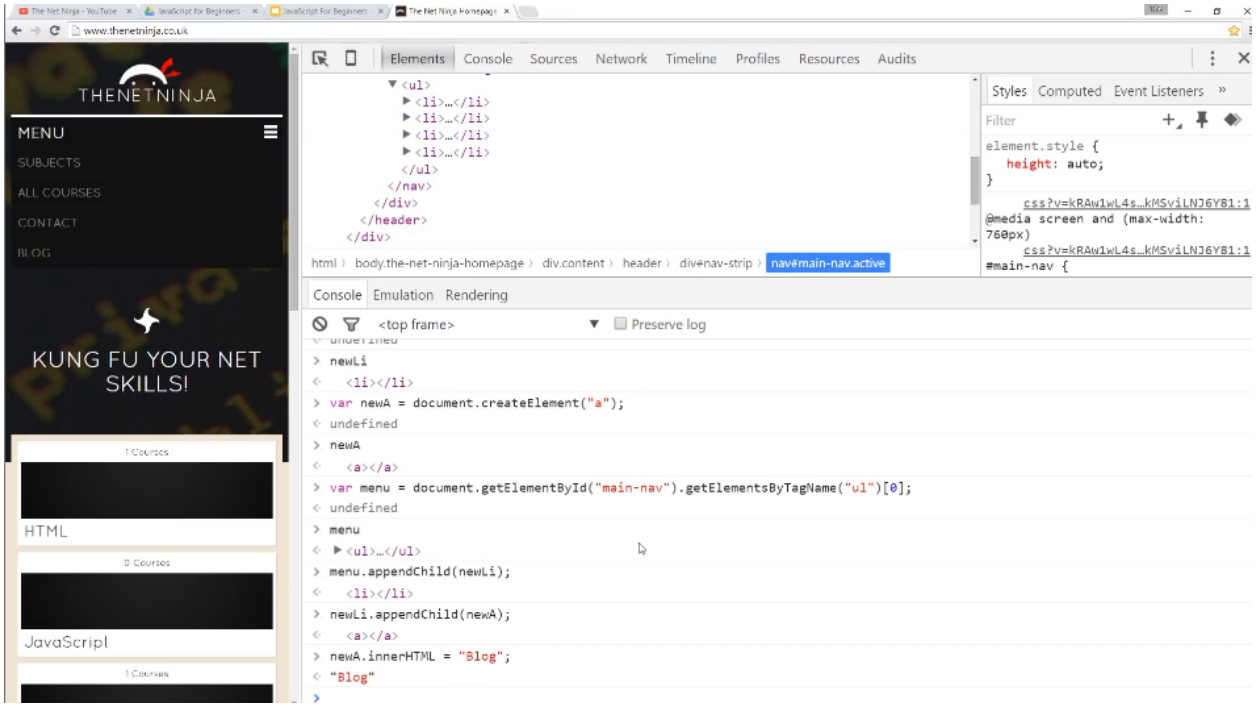
Step 1: Get the Data where to add “new Li”



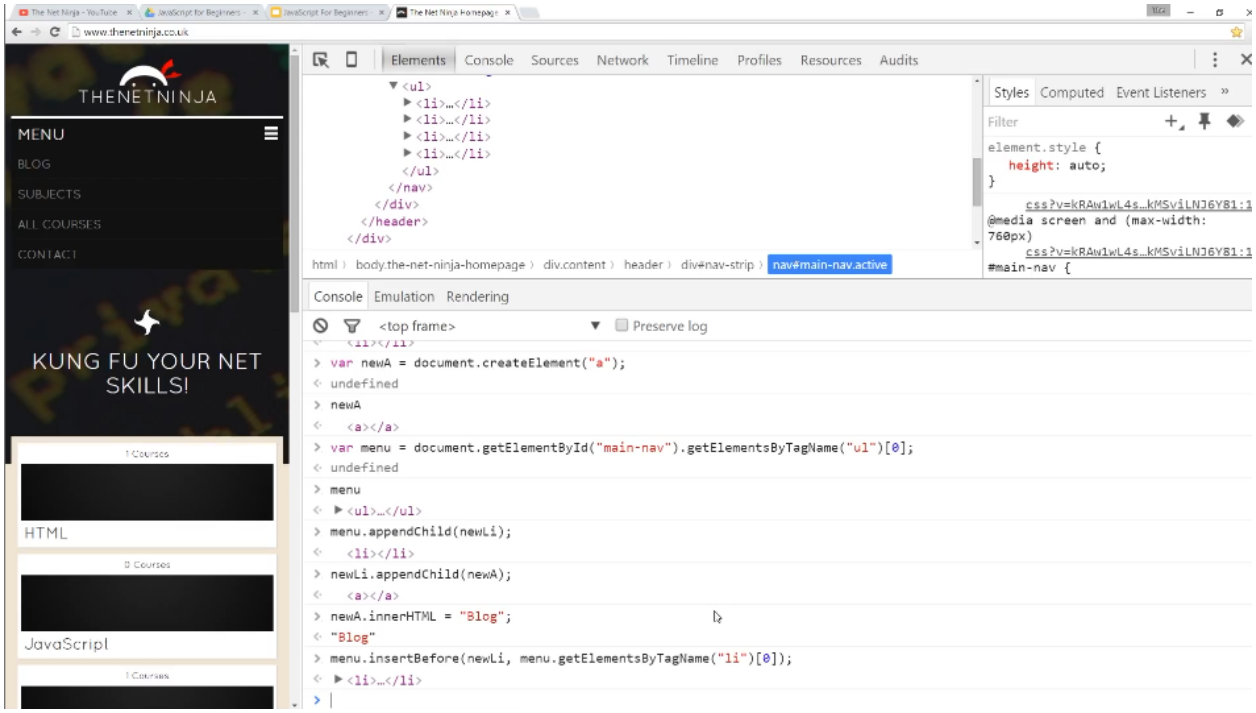
Step 2: Adding the “new link”



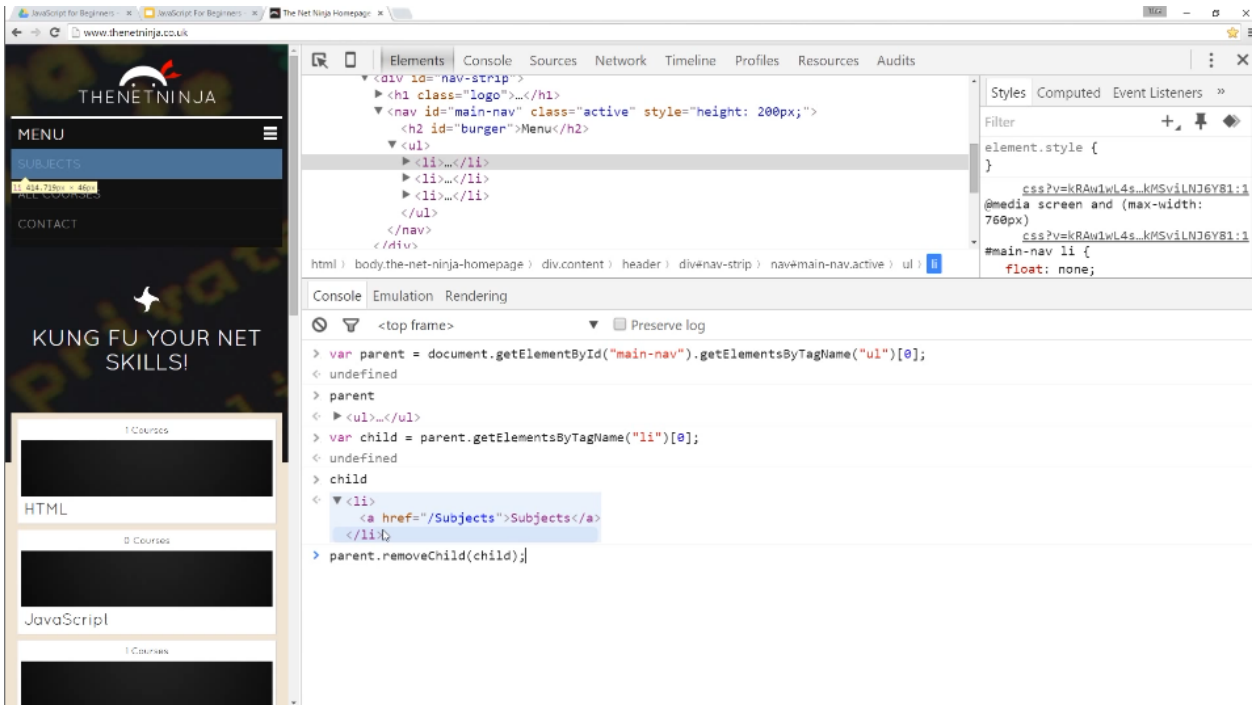
→Here “Li” is inserted at bottom & If the styles height is modified to “auto” it is visible in page



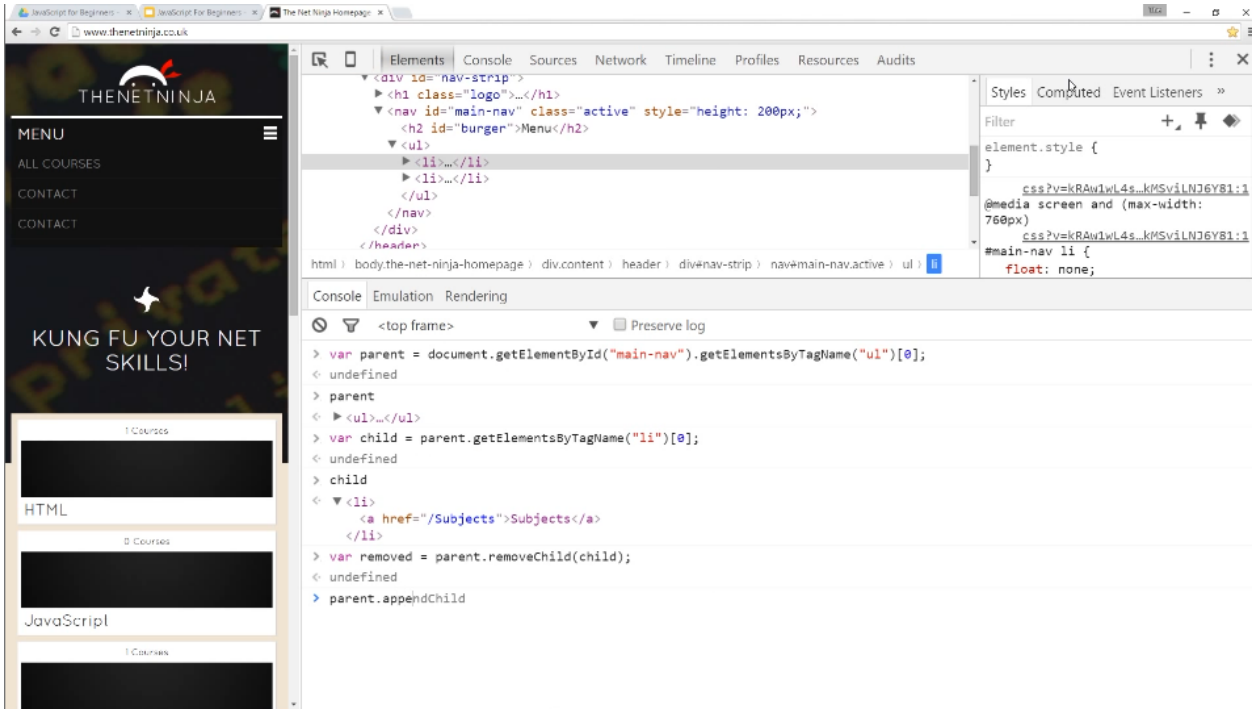
→Here “Li” is inserted at the Top



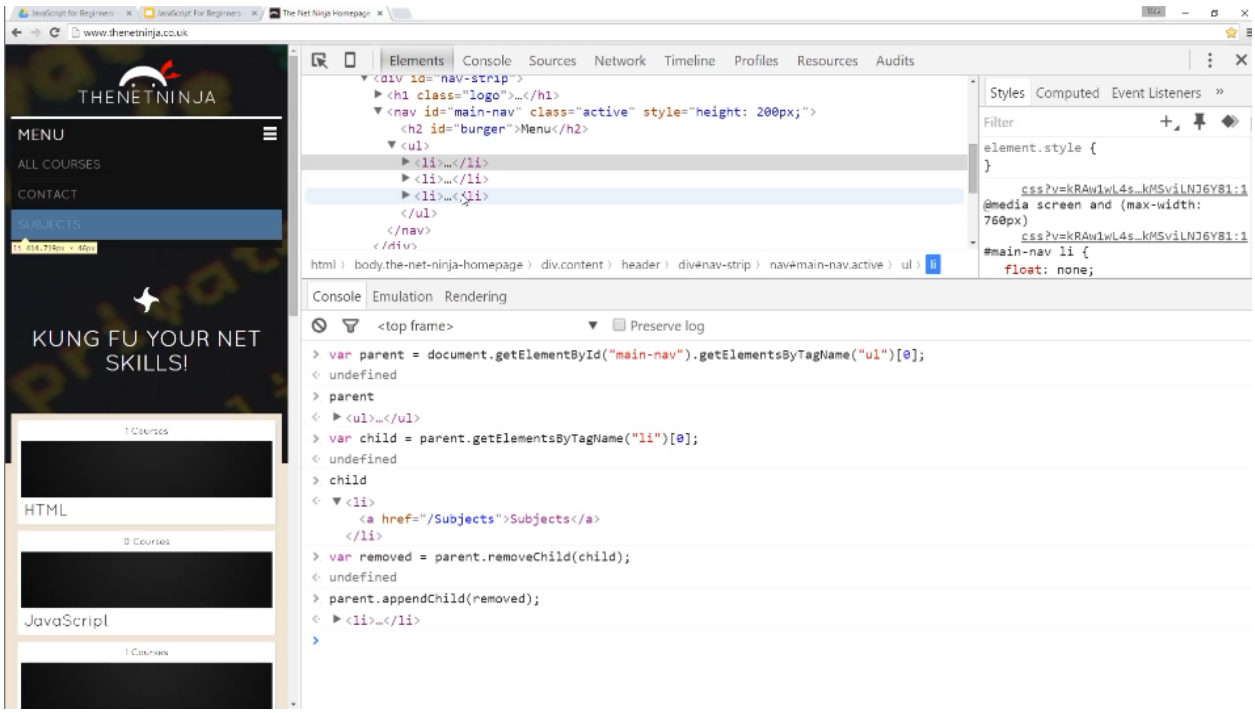
38. Removing Elements from the DOM



→Removed “Subjects”

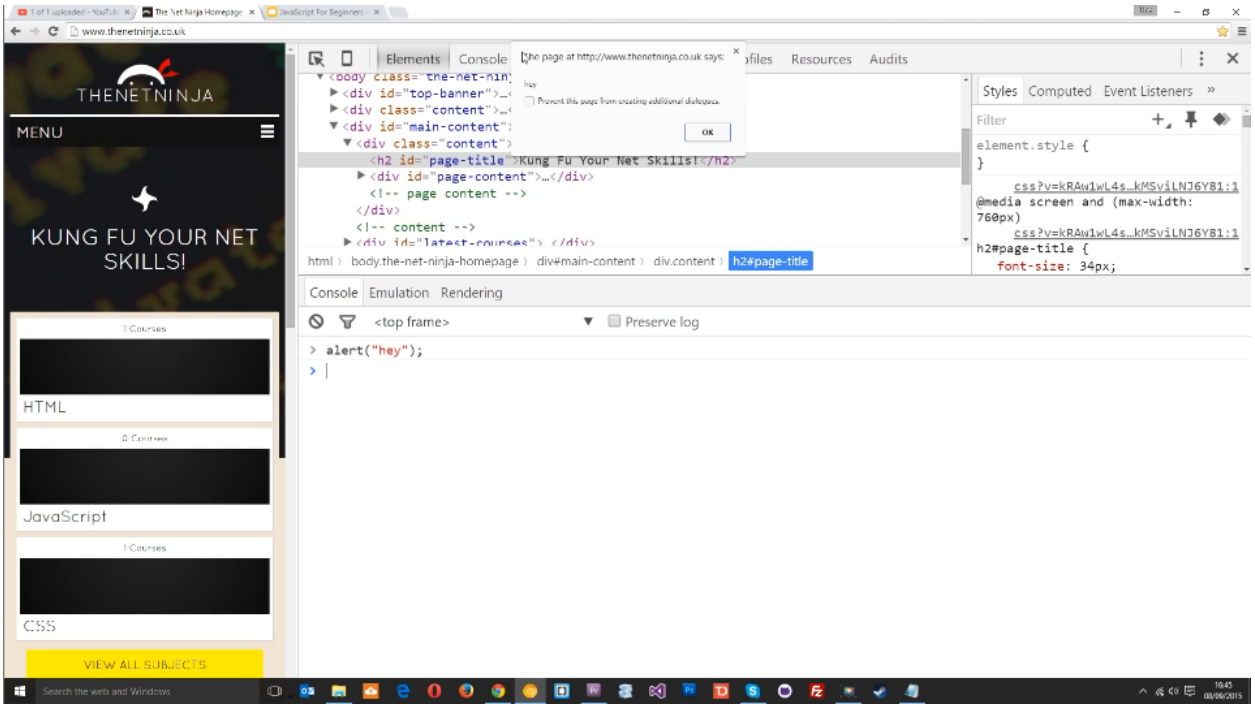


→added back to parent.

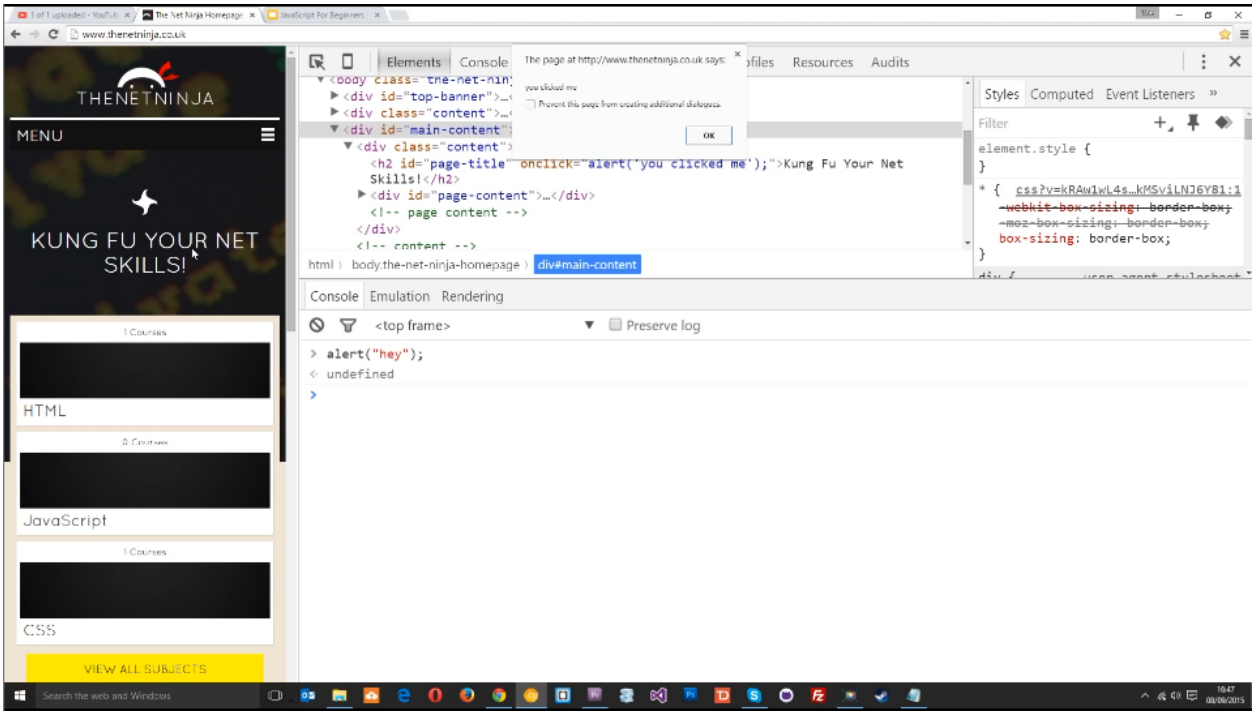


39. Introduction to JavaScript Events

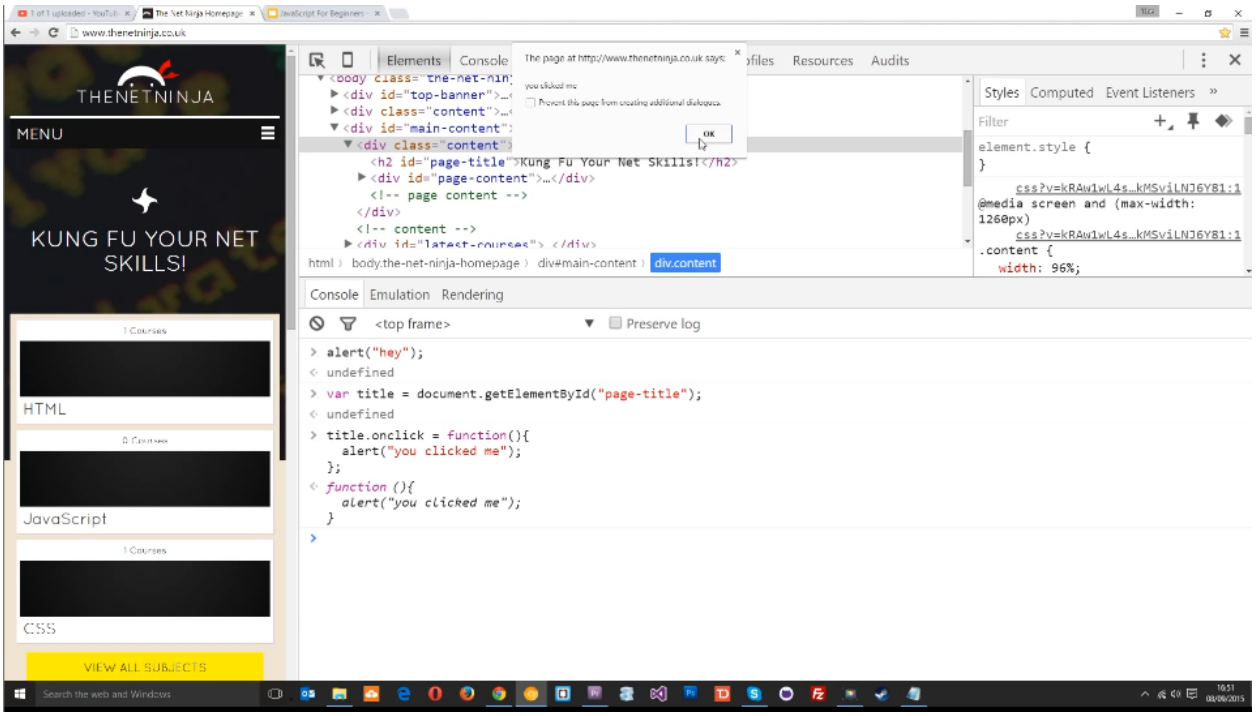
→Dialog box appear at Top of the page



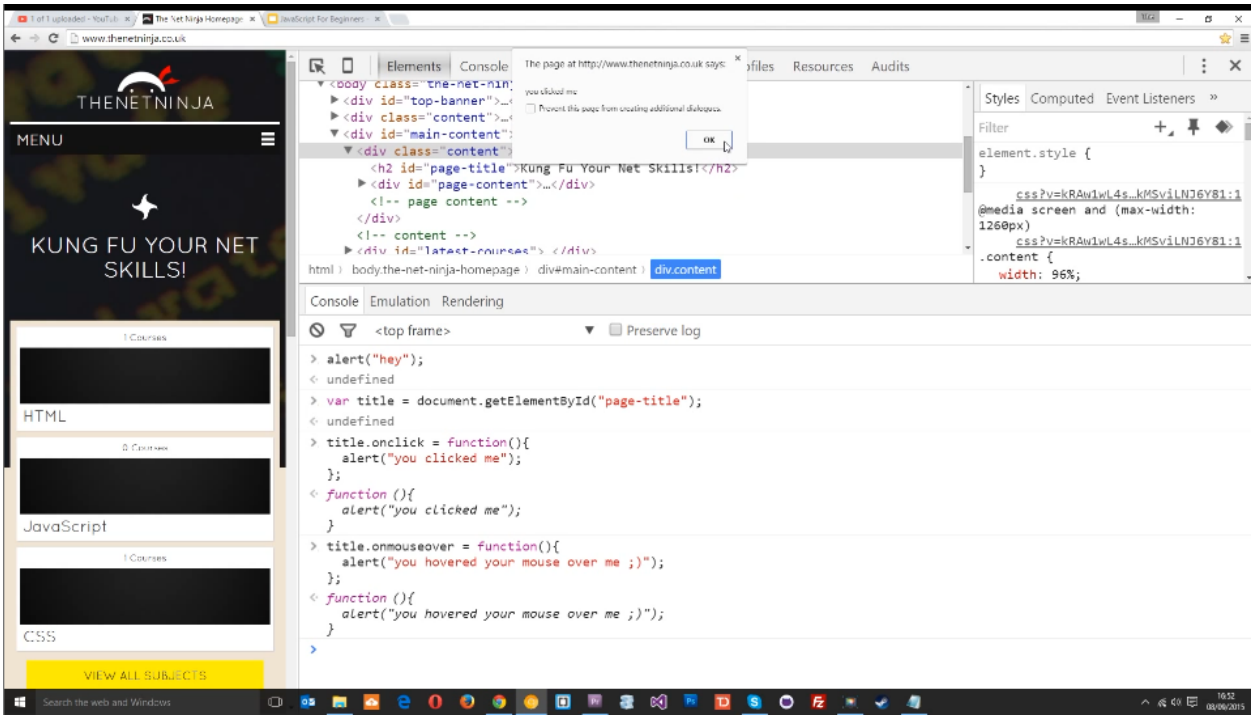
→Event is added (“onclick”)



→If I click on the “KUNG FU YOUR NET SKILLS!” it will show the dialog box



→If I drag the mouse on the “KUNG FU YOUR NET SKILLS!” it will show the dialog box



40. The onClick Event

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>JavaScript for Beginners</title>
    <link href="style.css" type = "text/css" rel = "stylesheet">
</head>
<body>

    <div id="content">
```

```
        <p>The ICC Cricket World Cup is an international cricket tournament held every four years. It is organized by the International Cricket Council (ICC) and features teams from around the world competing for the prestigious title. </p>

        <p>The Cricket World Cup 2023 is scheduled to be the 13th edition of the tournament. It is set to be hosted by India, with matches expected to be played in various cities across the country. The tournament will bring together the top cricketing nations to compete in a series of One Day International (ODI) matches</p>

        <p>The qualification process for the Cricket World Cup usually involves teams earning qualification spots through various regional tournaments and rankings. The final tournament typically consists of 10 teams competing in a round-robin format, followed by knockout stages leading up to the final match.</p>
    </div>

    <a id ="show-more">Show More</a>

    <script src="test.js"></script>

</body>
</html>
```

style.css

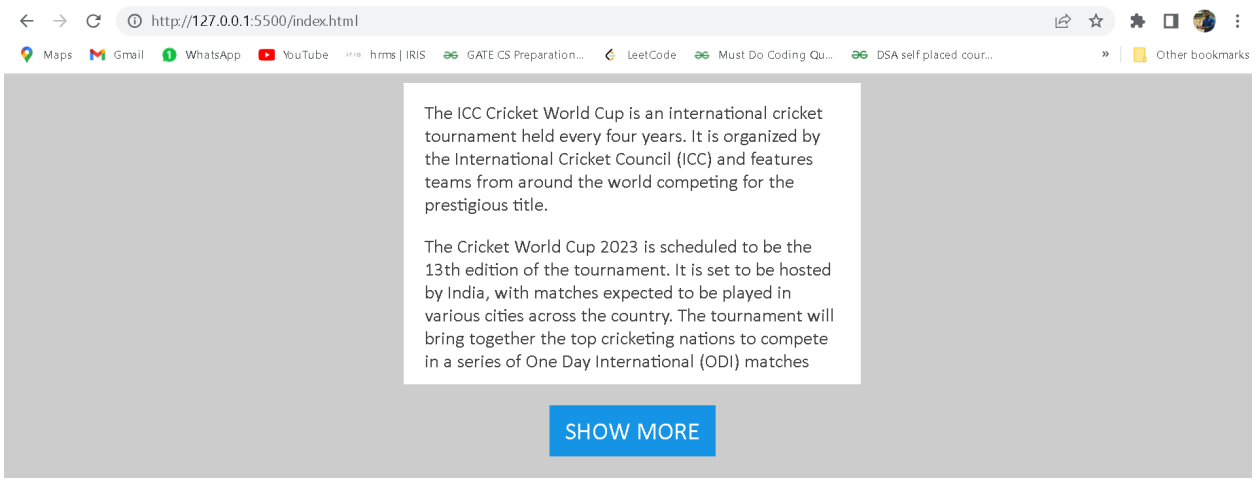
```
body{
    background: #ccc;
}

#content{
    width: 400px;
    background: #fff;
    padding: 20px;
    padding-top: 0;
    font-family: calibri;
```

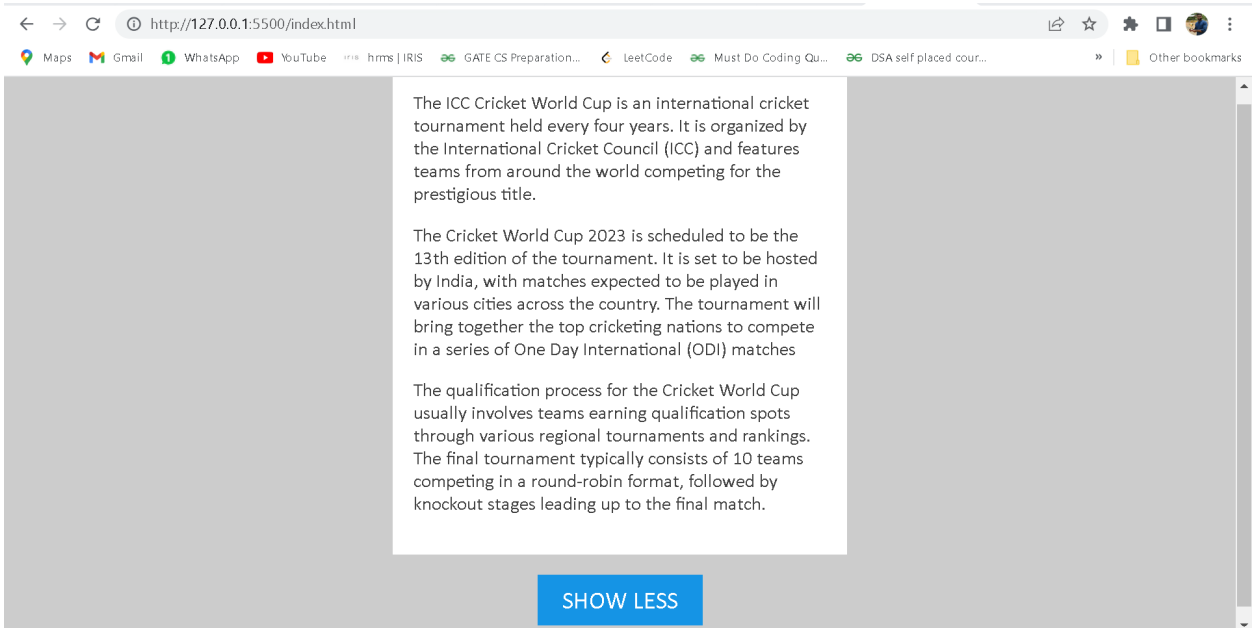
```
    font-size: 18px;
    color: #444;
    margin: 0 auto;

    max-height: 270px;
    overflow: hidden;

    /* set transitions up. */
    -webkit-transition: max-height 0.7s;
    -moz-transition: max-height 0.7s;
    transition: max-height 0.7s;
}

#content.open{
    max-height: 1000px;

    /* set transitions up. */
    -webkit-transition: max-height 0.7s;
    -moz-transition: max-height 0.7s;
    transition: max-height 0.7s;
}

#show-more{
    background: #1594e5;
    color: #fff;
    font-family: calibri;
    display: block;
    width: 140px;
    font-size: 24px;
    text-transform: uppercase;
    padding: 10px;
    text-align: center;
    margin: 20px auto;
    cursor: pointer;
}
```

test.js

```
var content = document.getElementById("content");
var button = document.getElementById("show-more");

button.onclick = function() {

    if(content.className == "open"){
        //shrink the box
        content.className = "";
        button.innerHTML = "Show More";
    }else{
        //expand the box
        content.className = "open";
        button.innerHTML = "Show Less";
    }
};
```

O/P:





41. Window onLoad Event

→If the **"script"** is placed at the head part then it won't load the script
That is why introduce **"window.onload"**

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>JavaScript for Beginners</title>
    <link href="style.css" type = "text/css" rel = "stylesheet">

    <script src="test.js"></script>

</head>
<body>
```

```
    <div id="content">
        <p>The ICC Cricket World Cup is an international cricket tournament
held every four years. It is organized by the International Cricket Council
(ICC) and features teams from around the world competing for the prestigious
title. </p>

        <p>The Cricket World Cup 2023 is scheduled to be the 13th edition of
the tournament. It is set to be hosted by India, with matches expected to be
played in various cities across the country. The tournament will bring
together the top cricketing nations to compete in a series of One Day
International (ODI) matches</p>

        <p>The qualification process for the Cricket World Cup usually
involves teams earning qualification spots through various regional
tournaments and rankings. The final tournament typically consists of 10 teams
competing in a round-robin format, followed by knockout stages leading up to
the final match.</p>
    </div>

    <a id ="show-more">Show More</a>



</body>
</html>
```

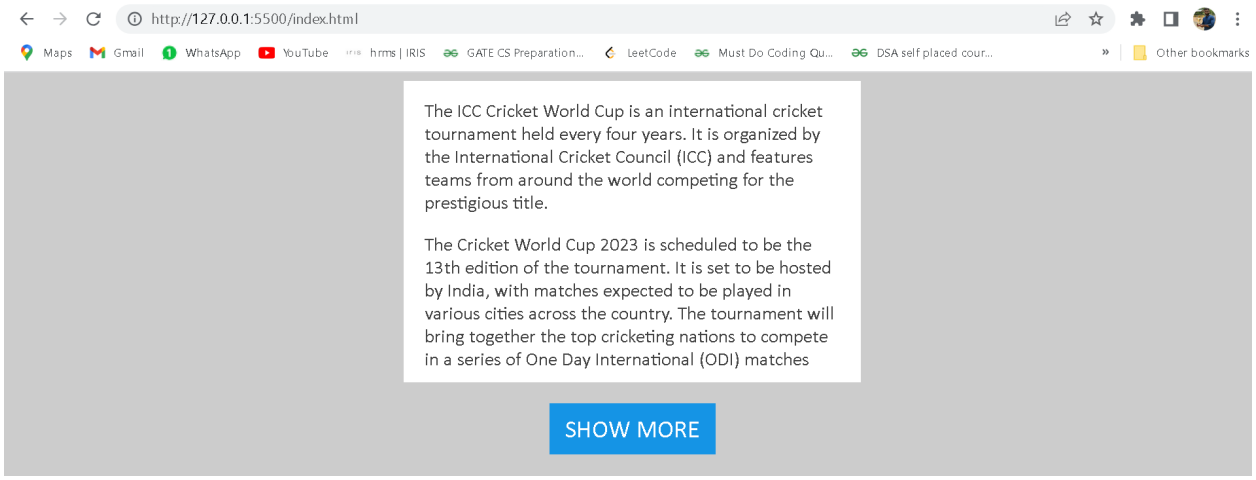
test.js

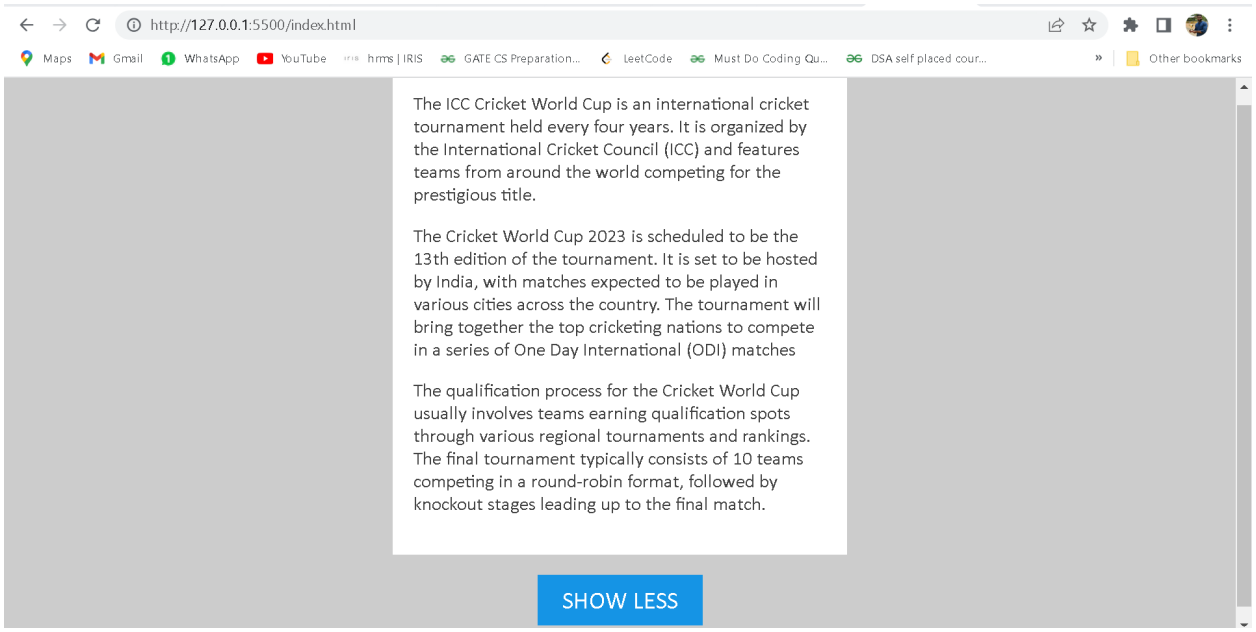
```
function setUpEvents(){

    var content = document.getElementById("content");
    var button = document.getElementById("show-more");

    button.onclick = function() {

    if(content.className == "open"){
        //shrink the box
        content.className = "";
```

```
        button.innerHTML = "Show More";
    }else{
        //expand the box
        content.className = "open";
        button.innerHTML = "Show Less";
    }
    };

}

window.onload = function(){

    setUpEvents();

};
```

→there is no change in the **"style.css"**

O/P:





42. JavaScript Timers

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>JavaScript for Beginners</title>
    <link href="style.css" type = "text/css" rel = "stylesheet">
</head>
<body>

    <div id="message">
        <p>THIS IS A REEEEEAAAALITY IMPORTANT NOTIFICATION</p>

        <P>PIE IS NICE</P>
    </div>

    <script src="test.js"></script>
```

```
</body>
</html>
```

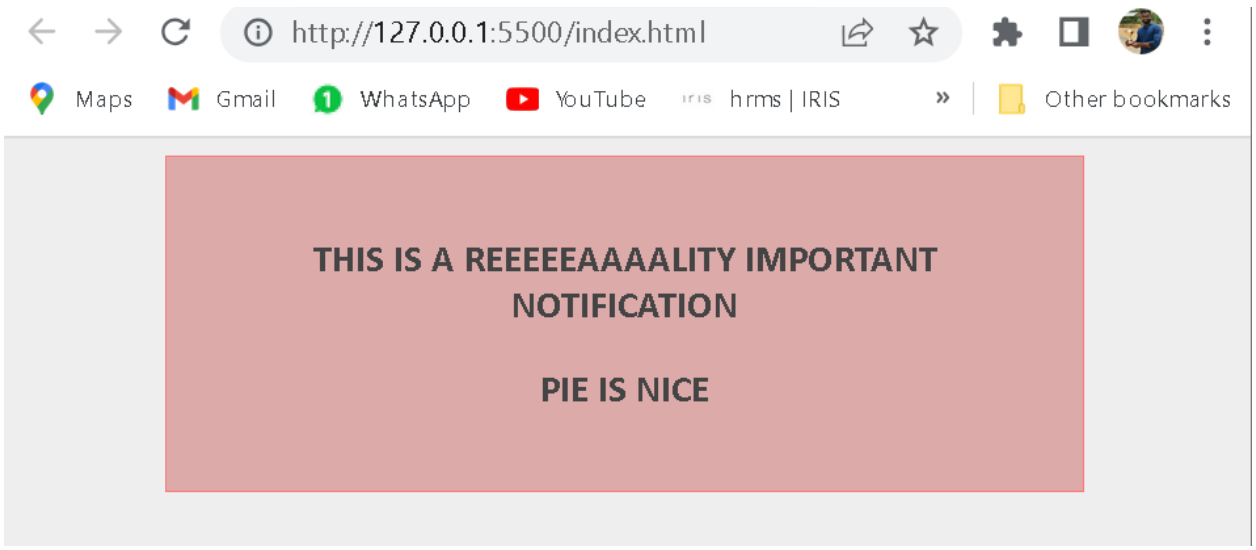
style.css

```
body{
    background: #eee;
}

#message{
    width: 400px;
    background: #ddaaaa;
    padding: 20px;
    font-family: calibri;
    font-size: 18px;
    color: #444;
    margin: 0 auto;
    text-align: center;
    border: 1px solid #ff7777;
    font-weight: bold;
    opacity: 0;

    /* set transitions up. */
    -webkit-transition: opacity 0.7s;
    -moz-transition: opacity 0.7s;
    transition: opacity 0.7s;
}

#message.show{
    opacity: 1;

    /* set transitions up. */
    -webkit-transition: opacity 0.7s;
    -moz-transition: opacity 0.7s;
    transition: opacity 0.7s;
}
```

test.js

```
var myMessage = document.getElementById("message");

function showMessage() {

    myMessage.className = "show";
}

setTimeout(showMessage, 3000);
```

O/P:



index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>JavaScript for Beginners</title>
    <link href="style.css" type = "text/css" rel = "stylesheet">
</head>
<body>
```

```
    <div id="colour-changer"></div>

    <script src="test.js"></script>

</body>
</html>
```

style.css

```
body{
    background: #eee;
}

#message{
    width: 400px;
    background: #ddaaaa;
    padding: 20px;
    font-family: calibri;
    font-size: 18px;
    color: #444;
    margin: 0 auto;
    text-align: center;
    border: 1px solid #ff7777;
    font-weight: bold;
    opacity: 0;

    /* set transitions up. */
    -webkit-transition: opacity 0.7s;
    -moz-transition: opacity 0.7s;
    transition: opacity 0.7s;
}

#message.show{
    opacity: 1;

    /* set transitions up. */
    -webkit-transition: opacity 0.7s;
```

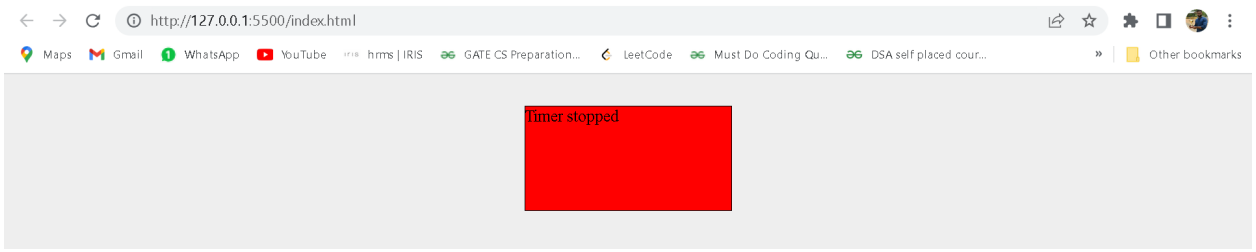
```
    -moz-transition: opacity 0.7s;
    transition: opacity 0.7s;
}

#colour-changer{
    width: 200px;
    height: 100px;
    margin: 30px auto;
    border: 1px solid #000;
    background: #fff;

    -webkit-transition: background 0.7s;
    -moz-transition: background 0.7s;
    transition: background 0.7s;
}
```

test.js

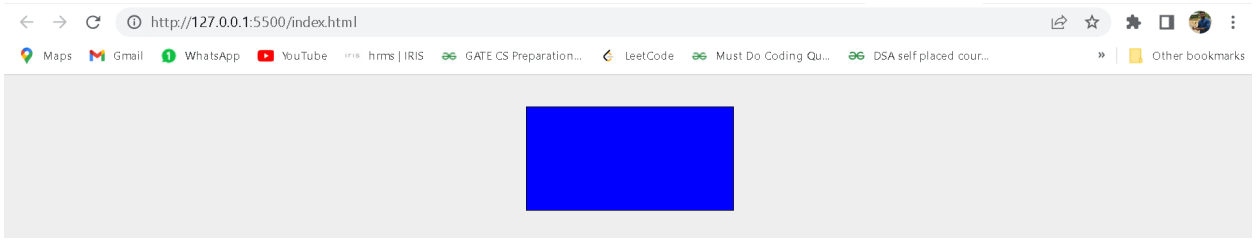
```
var colourChanger = document.getElementById("colour-changer");
var colours = ["red", "blue", "green", "pink"];
var counter = 0;

function changeColour() {

    if(counter >= colours.length){  //repeating the cycle
        counter = 0;
    }

    colourChanger.style.background = colours[counter];
    counter++;

}

var myTimer = setInterval(changeColour, 3000);  //3000ms or 3sec

colourChanger.onclick = function() {

    clearInterval(myTimer);
    colourChanger.innerHTML = "Timer stopped";
```

```
};
```

O/P:



→after every 3sec it will change the colour



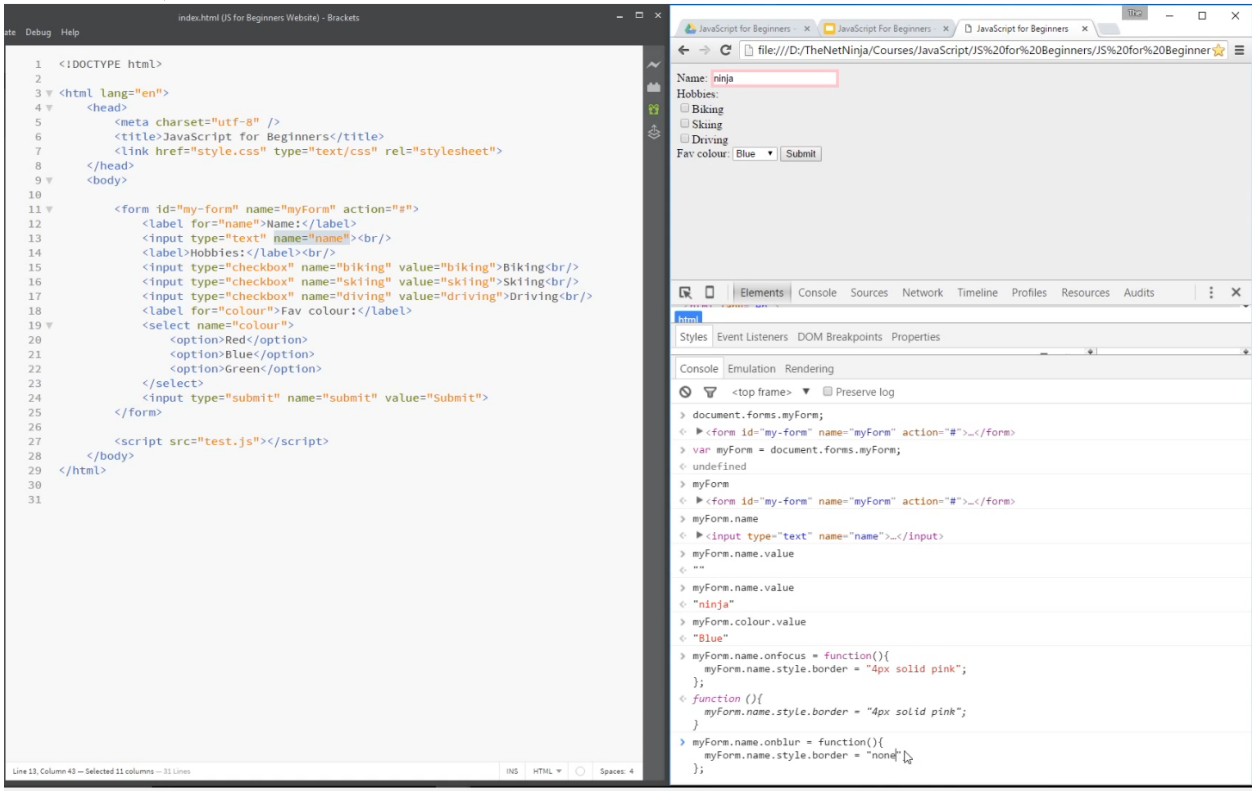
43. Accessing Form Elements

index.html

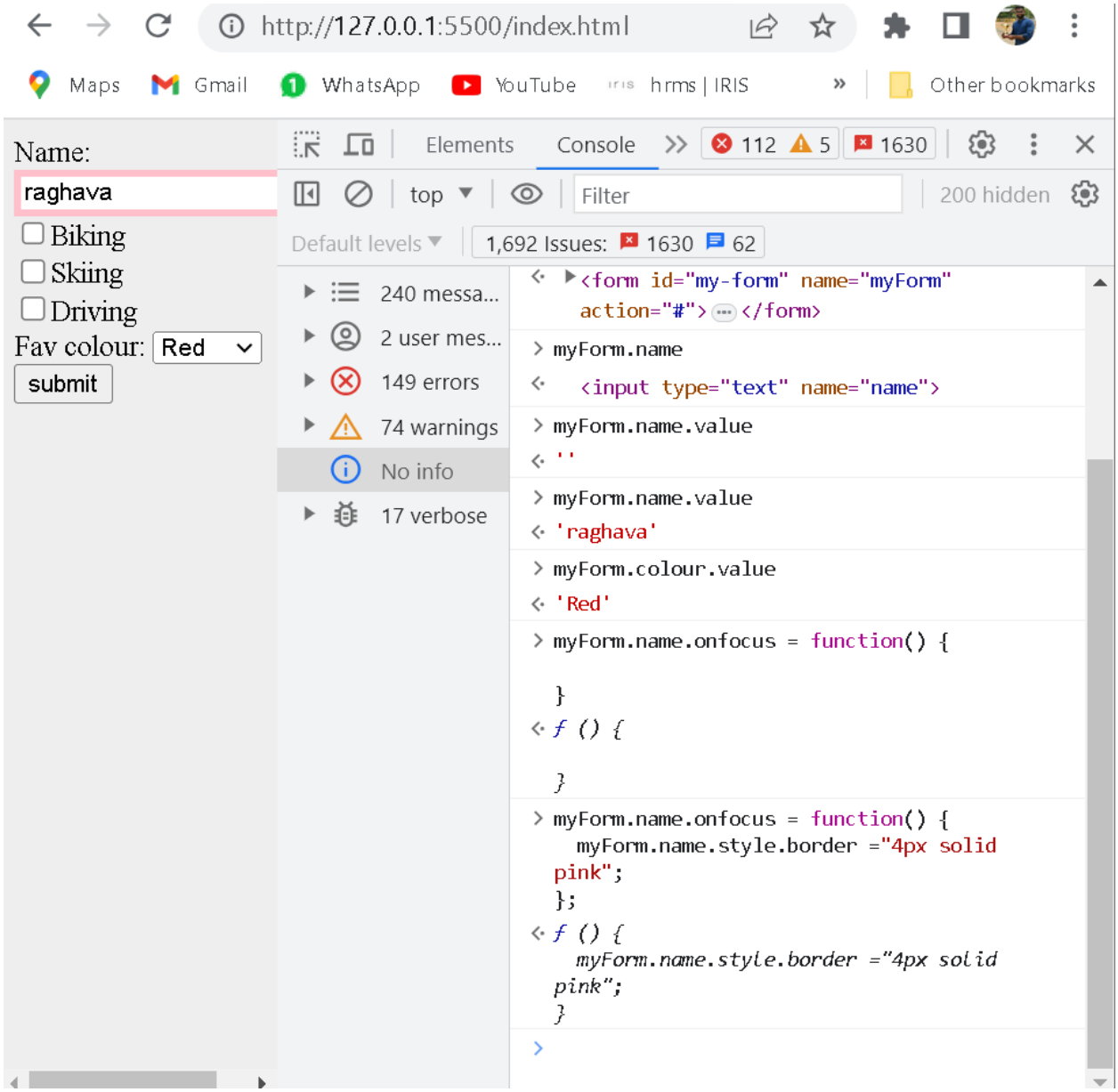
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>JavaScript for Beginners</title>
    <link href="style.css" type = "text/css" rel = "stylesheet">
</head>
<body>

    <form id="my-form" name="myForm" action="#">
        <label for="name">Name:</label>
```

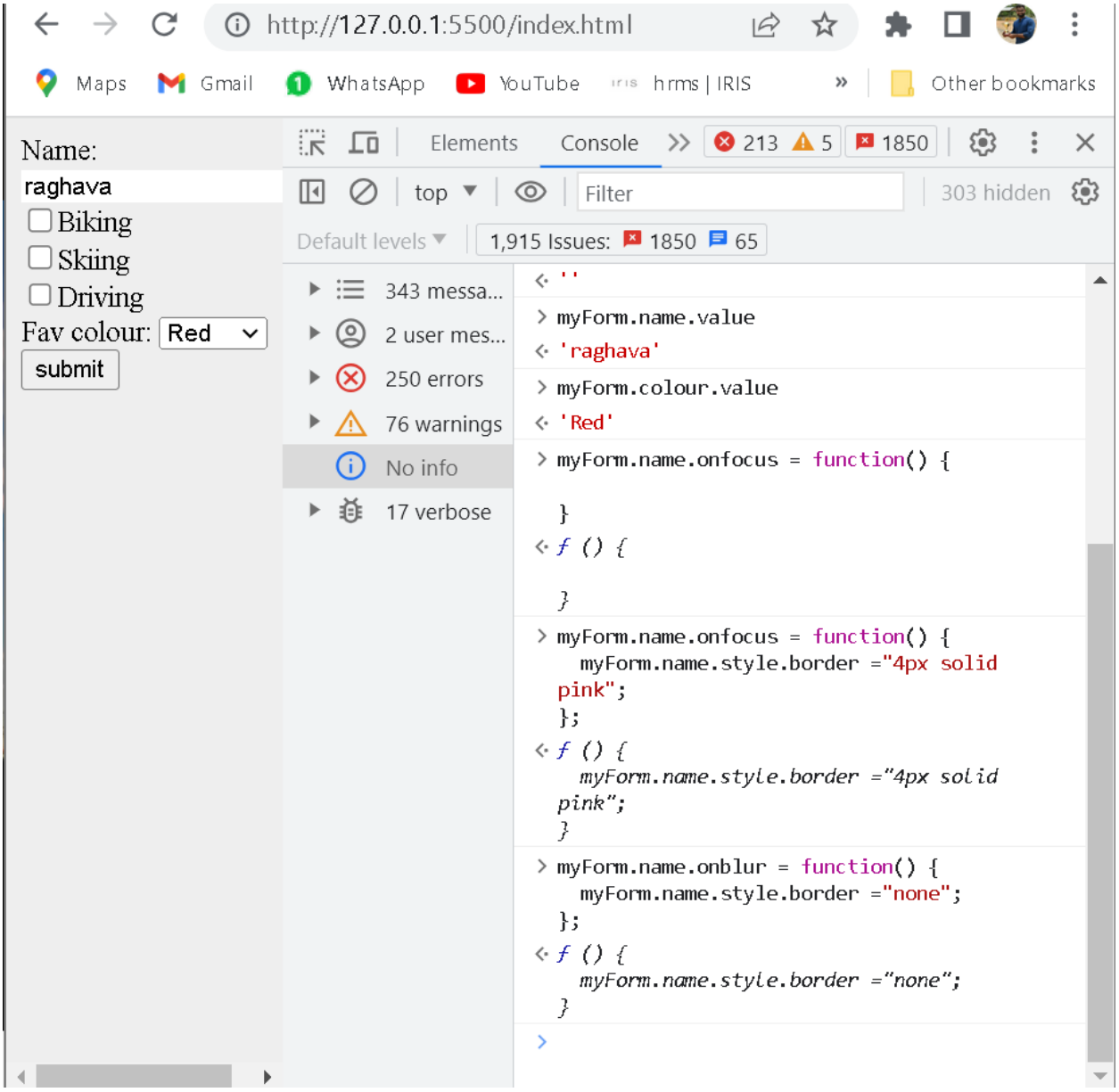
```
        <input type="text" name="name"><br>
        <input type="checkbox" name="biking" value="biking">Biking<br>
        <input type="checkbox" name="skiing" value="skiing">Skiing<br>
        <input type="checkbox" name="driving" value="driving">Driving<br>
        <label for="colours">Fav colour:</label>
        <select name="colour" id="">
            <option>Red</option>
            <option>Blue</option>
            <option>Green</option>
        </select>
        <input type="submit" name="submit" value="submit">
    </form>
    <script src="test.js"></script>

</body>
</html>
```

→there is no change in the test.js & style.css



O/P:


http://127.0.0.1:5500/index.html
Maps
Gmail
WhatsApp
YouTube
hrms | IRIS
Other bookmarks
Name:
raghava
Biking
Skiing
Driving
Fav colour: Red
submit
Elements
Console
112
5
1630
top
Filter
200 hidden
Default levels
1,692 Issues: 1630 62
240 messa...
2 user mes...
149 errors
74 warnings
No info
17 verbose
<form id="my-form" name="myForm" action="#"> </form>
myForm.name
<input type="text" name="name">
myForm.name.value
''
myForm.name.value
'raghava'
myForm.colour.value
'Red'
myForm.name.onfocus = function() {
}
ƒ () {
}
myForm.name.onfocus = function() {
myForm.name.style.border ="4px solid pink";
};
ƒ () {
myForm.name.style.border ="4px solid pink";
}


http://127.0.0.1:5500/index.html
Maps
Gmail
WhatsApp
YouTube
hrms | IRIS
Other bookmarks
Name:
raghava
Biking
Skiing
Driving
Fav colour: Red
submit
Elements
Console
213 5 1850
top
Filter
303 hidden
Default levels
1,915 Issues: 1850 65
343 messa...
2 user mes...
250 errors
76 warnings
No info
17 verbose
''
myForm.name.value
'raghava'
myForm.colour.value
'Red'
myForm.name.onfocus = function() {
}
ƒ () {
}
myForm.name.onfocus = function() {
myForm.name.style.border ="4px solid pink";
};
ƒ () {
myForm.name.style.border ="4px solid pink";
}
myForm.name.onblur = function() {
myForm.name.style.border ="none";
};
ƒ () {
myForm.name.style.border ="none";
}

44.Very Simple Form Validation

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>JavaScript for Beginners</title>
    <link href="style.css" type = "text/css" rel = "stylesheet">
</head>
<body>

    <form id="my-form" name="myForm" action="#">
        <label for="name">Name:</label>
        <input type="text" name="name"><br>
        <label>Hobbies:</label><br>
        <input type="checkbox" name="biking" value="biking">Biking<br>
        <input type="checkbox" name="skiing" value="skiing">Skiing<br>
        <input type="checkbox" name="driving" value="driving">Driving<br>
        <label for="colour">Fav colour:</label>
        <select name="colour">
            <option>Red</option>
            <option>Blue</option>
            <option>Green</option>
        </select>
        <input type="submit" name="submit" value="Submit">
    </form>

    <div id="message" style="color: red;"></div>

    <script src="test.js"></script>
</body>
</html>
```

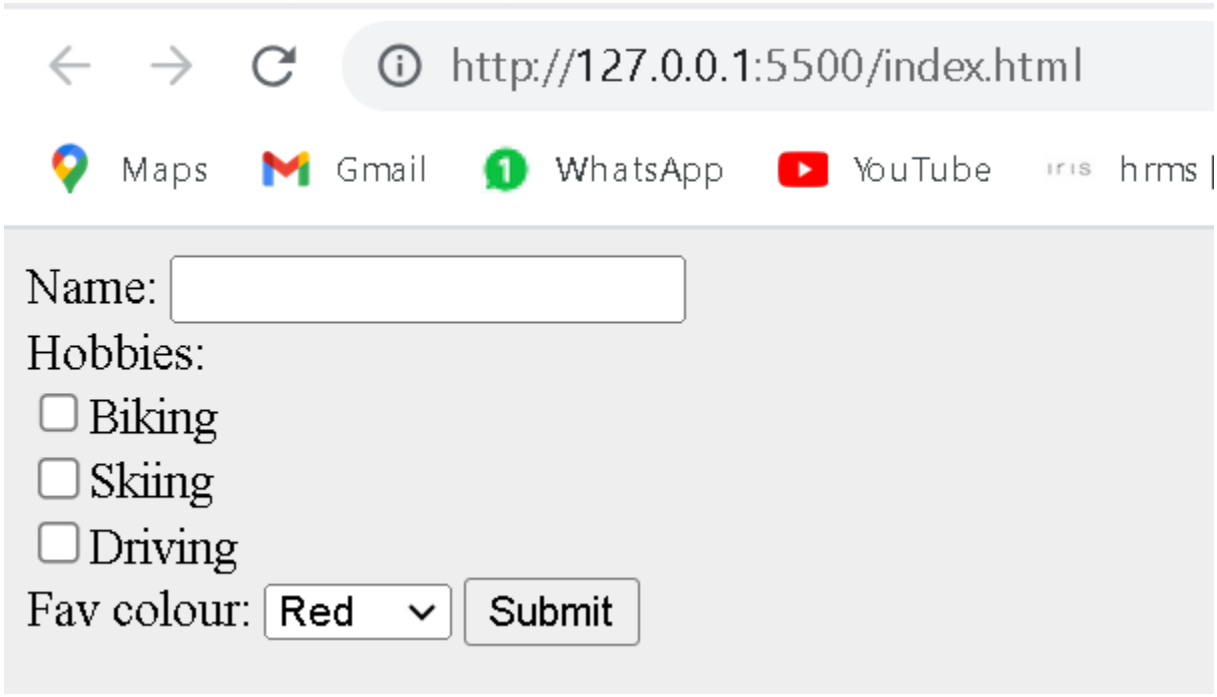
style.css

```css
body{
    background: #eee;
}

#message{
    width: 400px;
    font-family: calibri;
    font-size: 18px;
    color: #444;
    font-weight: bold;

    /* set transitions up. */
    -webkit-transition: opacity 0.7s;
    -moz-transition: opacity 0.7s;
    transition: opacity 0.7s;
}
```

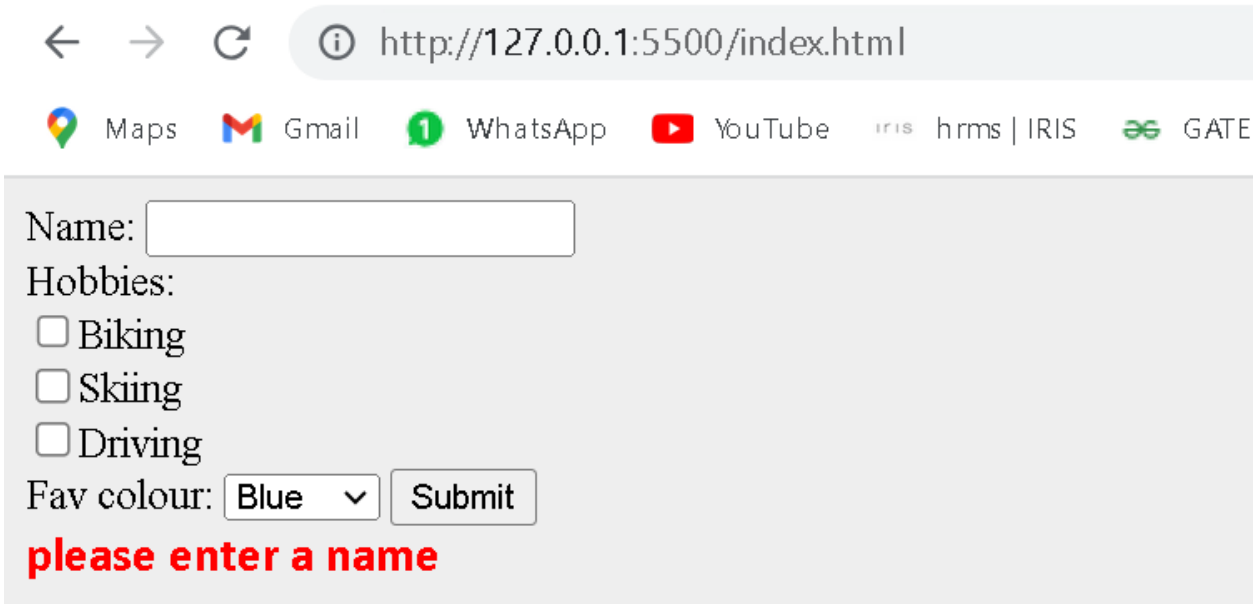
test.js

```javascript
var myForm = document.getElementById("my-form");
var message = document.getElementById("message");

myForm.onsubmit = function() {

    if(myForm.name.value == ""){
        message.innerHTML = "please enter a name";
        return false;
    } else{
        message.innerHTML = "";
        return true;
    }
};
```

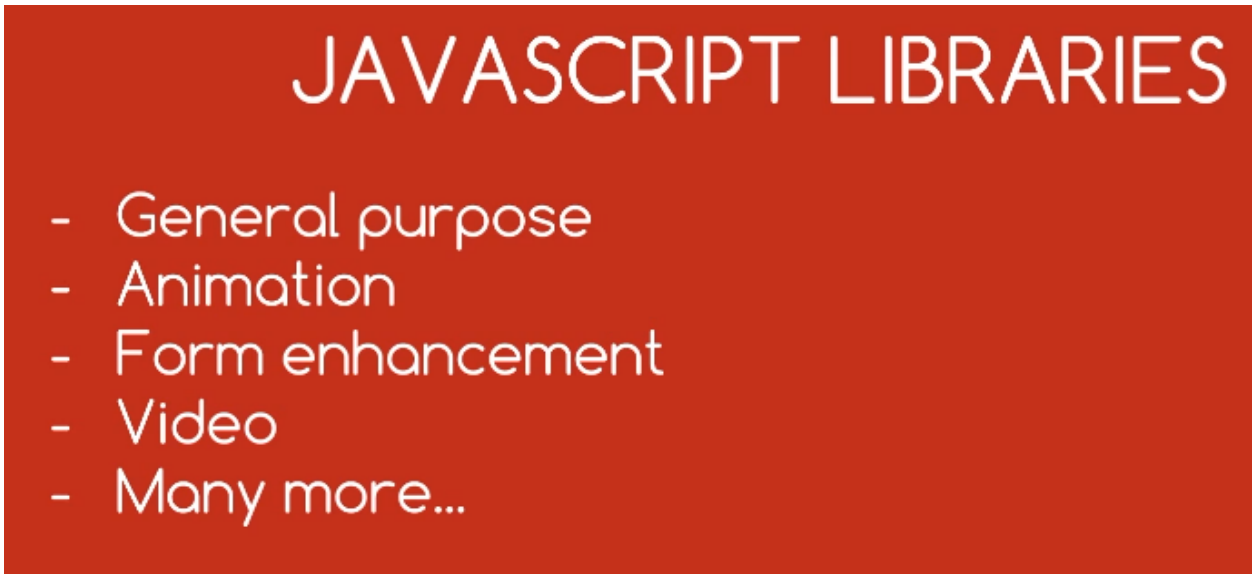
O/P:


http://127.0.0.1:5500/index.html
Maps
Gmail
WhatsApp
YouTube
hrms |
Name:
Hobbies:
Biking
Skiing
Driving
Fav colour:
Red
Submit


http://127.0.0.1:5500/index.html?name=Gudiwada+Raghava&colour=Blue&submit=Submit#
Maps
Gmail
WhatsApp
YouTube
hrms | IRIS
GATE CS Preparation...
LeetCode
Must Do Coding Qu...
Name: Gudiwada Raghava
Hobbies:
Biking
Skiing
Driving
Fav colour: Blue
Submit



45. JavaScript Libraries



https://jquery.com/ →Download jQuery click

Go to the "Using jQuery with a CDN"

https://releases.jquery.com/ →jQuery

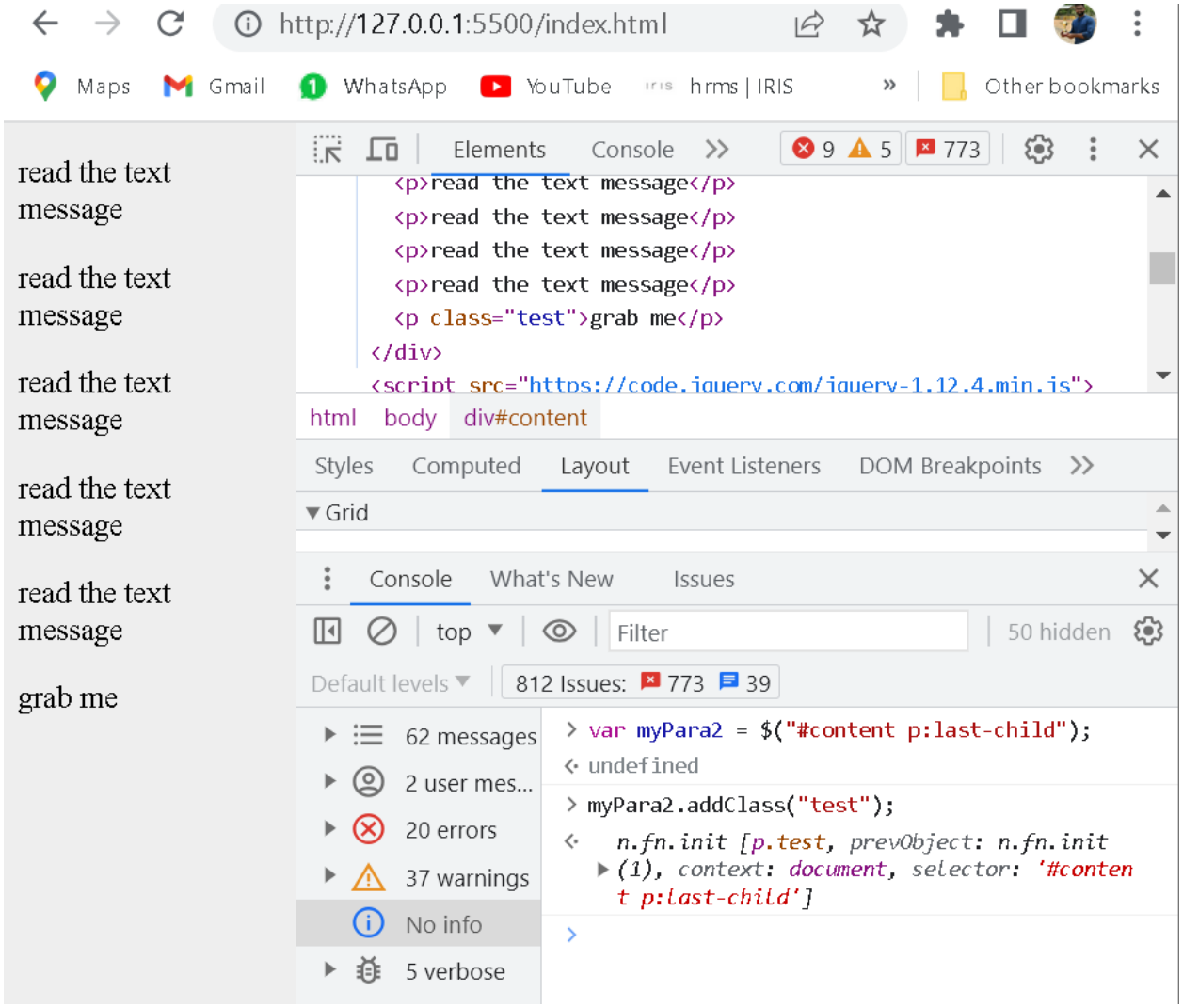
index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <title>JavaScript for Beginners</title>
    <link href="style.css" type = "text/css" rel = "stylesheet">
</head>
<body>

    <div id="content">
        <p>read the text message</p>
        <p>read the text message</p>
        <p>read the text message</p>
        <p>read the text message</p>
        <p>read the text message</p>
        <p>grab me</p>

    </div>

    <script src="https://code.jquery.com/jquery-1.12.4.min.js"></script>
    <script src="test.js"></script>
</body>
</html>
```

test.js

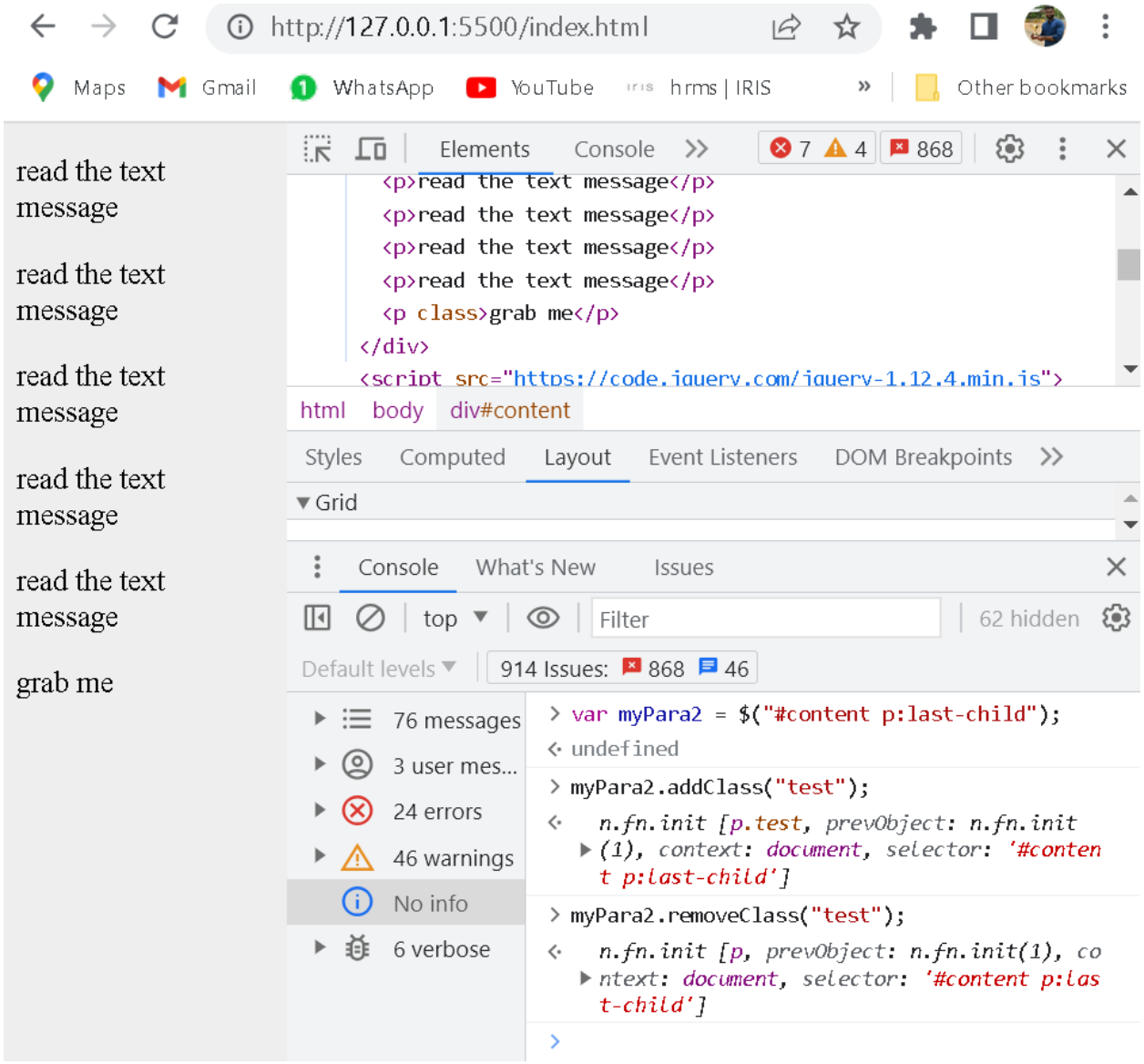
```
var myPara = document.getElementById("content").getElementsByTagName("p")[5];
var myPara2 = $("#content p:last-child");
```

→add class

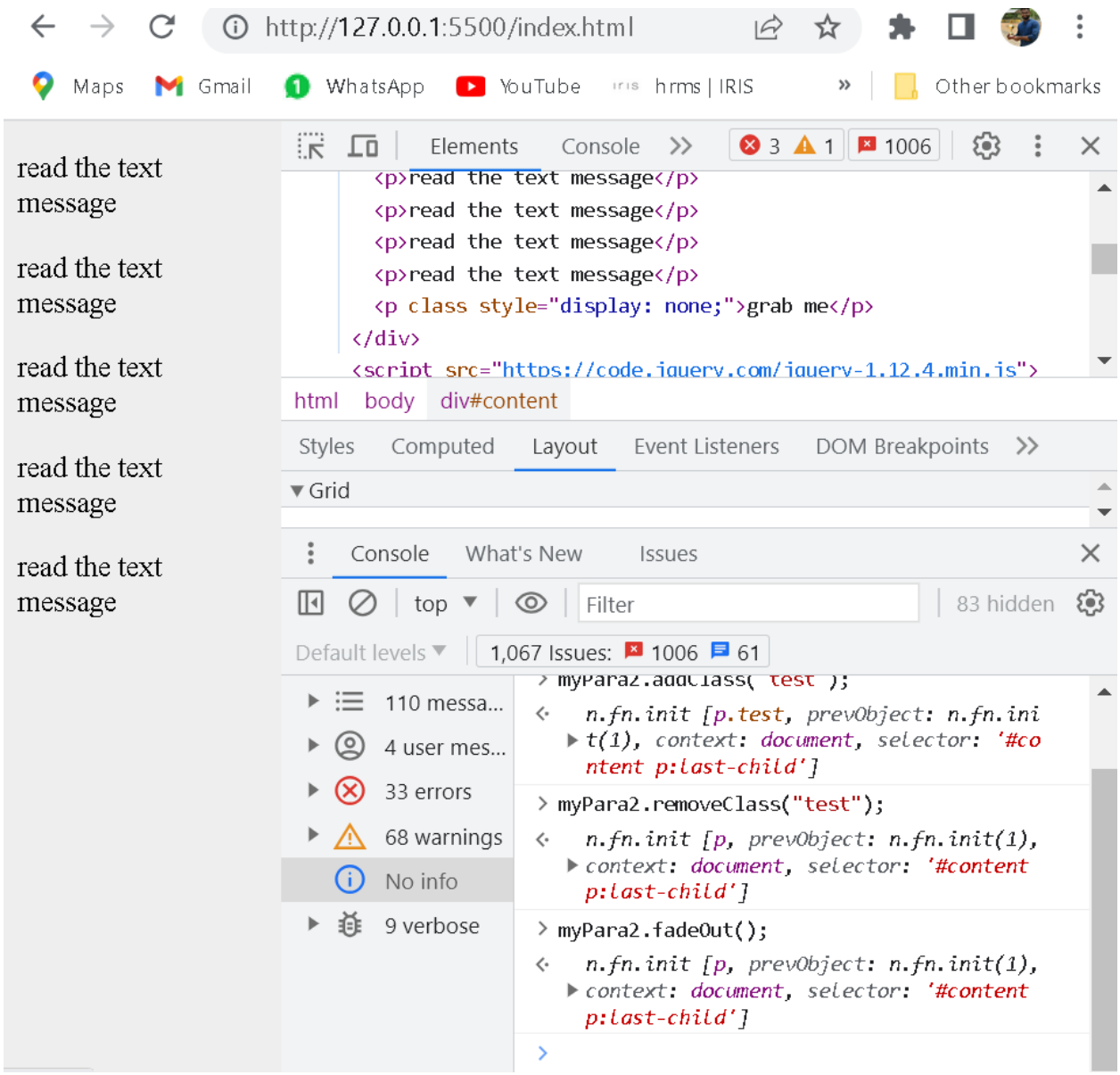

http://127.0.0.1:5500/index.html
Maps
Gmail
WhatsApp
YouTube
hrms | IRIS
Other bookmarks
read the text message
read the text message
read the text message
read the text message
read the text message
grab me
Elements
Console
9
5
773
read the text message
read the text message
read the text message
read the text message
grab me

<script src="https://code.jquery.com/jquery-1.12.4.min.js">
html
body
div#content
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Grid
Console
What's New
Issues
top
Filter
50 hidden
Default levels
812 Issues: 773 39
62 messages
2 user mes...
20 errors
37 warnings
No info
5 verbose
var myPara2 = $("#content p:last-child");
undefined
myPara2.addClass("test");
n.fn.init [p.test, prevObject: n.fn.init (1), context: document, selector: '#content p:last-child']

→remove class

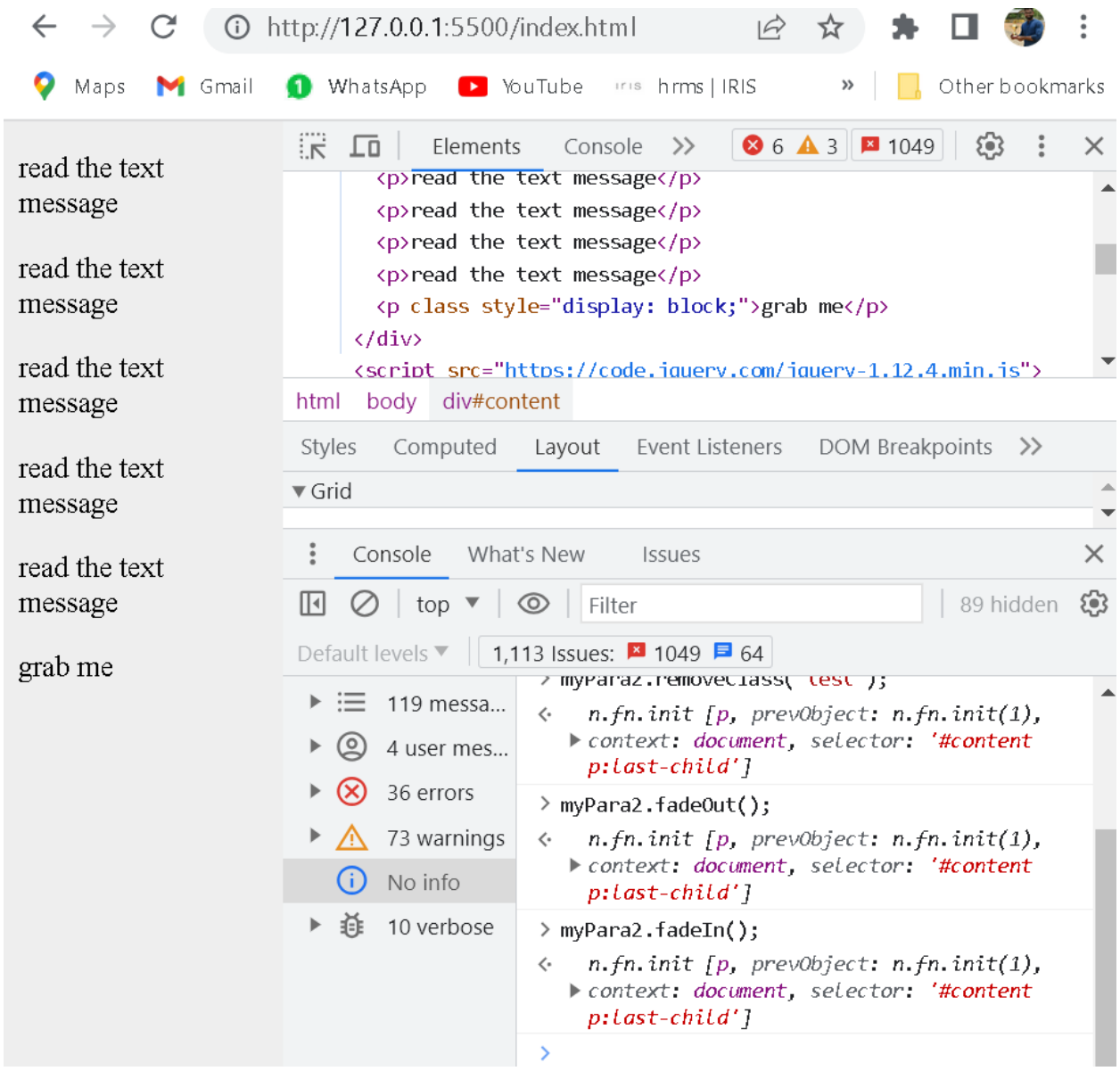


→FadeOut


http://127.0.0.1:5500/index.html
Maps
Gmail
WhatsApp
YouTube
hrms | IRIS
Other bookmarks
read the text message
read the text message
read the text message
read the text message
read the text message
Elements
Console
3
1
1006
read the text message
read the text message
read the text message
read the text message
<p class style="display: none;">grab me

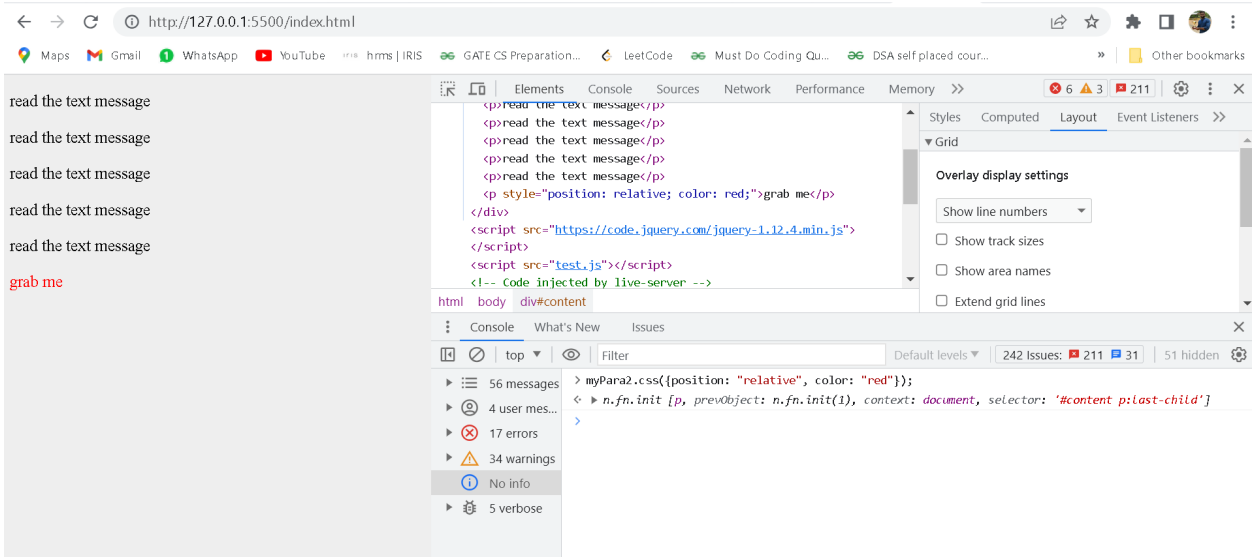
<script src="https://code.jquery.com/jquery-1.12.4.min.js">
html
body
div#content
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Grid
Console
What's New
Issues
top
Filter
83 hidden
Default levels
1,067 Issues: 1006 61
110 messa...
4 user mes...
33 errors
68 warnings
No info
9 verbose
myPara2.addClass(test);
n.fn.init [p.test, prevObject: n.fn.init(1), context: document, selector: '#content p:last-child']
myPara2.removeClass("test");
n.fn.init [p, prevObject: n.fn.init(1), context: document, selector: '#content p:last-child']
myPara2.fadeOut();
n.fn.init [p, prevObject: n.fn.init(1), context: document, selector: '#content p:last-child']

→FadeIn

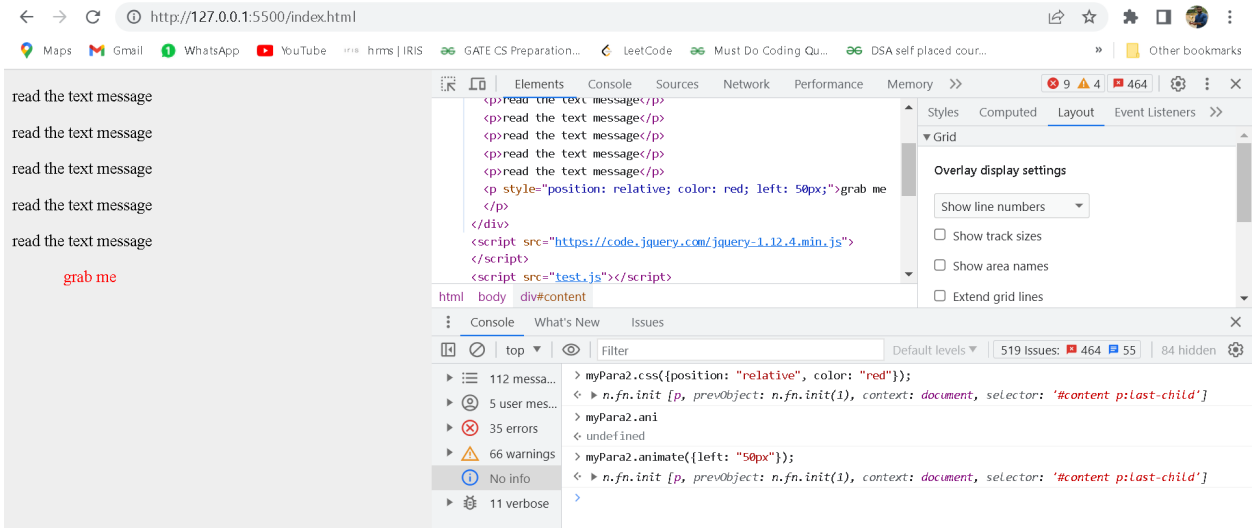

http://127.0.0.1:5500/index.html
Maps
Gmail
WhatsApp
YouTube
hrms | IRIS
Other bookmarks
read the text message
read the text message
read the text message
read the text message
read the text message
grab me
Elements
Console
6
3
1049
read the text message
read the text message
read the text message
read the text message
<p class style="display: block;">grab me

<script src="https://code.jquery.com/jquery-1.12.4.min.js">
html
body
div#content
Styles
Computed
Layout
Event Listeners
DOM Breakpoints
Grid
Console
What's New
Issues
top
Filter
89 hidden
Default levels
1,113 Issues: 1049 64
119 messa...
4 user mes...
36 errors
73 warnings
No info
10 verbose
n.fn.init [p, prevObject: n.fn.init(1), context: document, selector: '#content p:last-child']
myPara2.fadeOut();
n.fn.init [p, prevObject: n.fn.init(1), context: document, selector: '#content p:last-child']
myPara2.fadeIn();
n.fn.init [p, prevObject: n.fn.init(1), context: document, selector: '#content p:last-child']

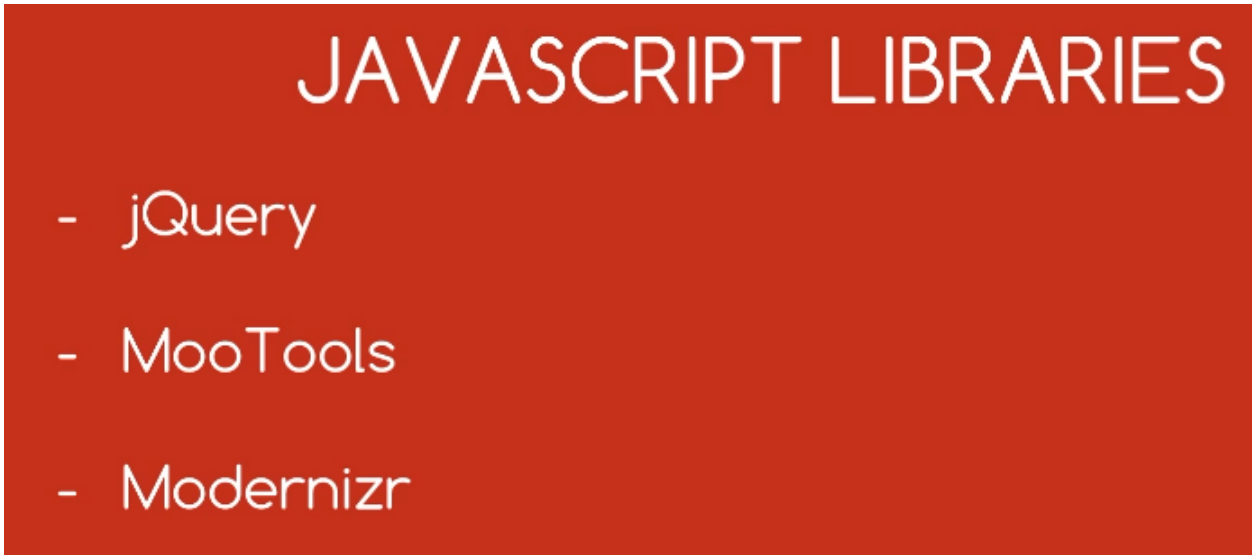
→add css

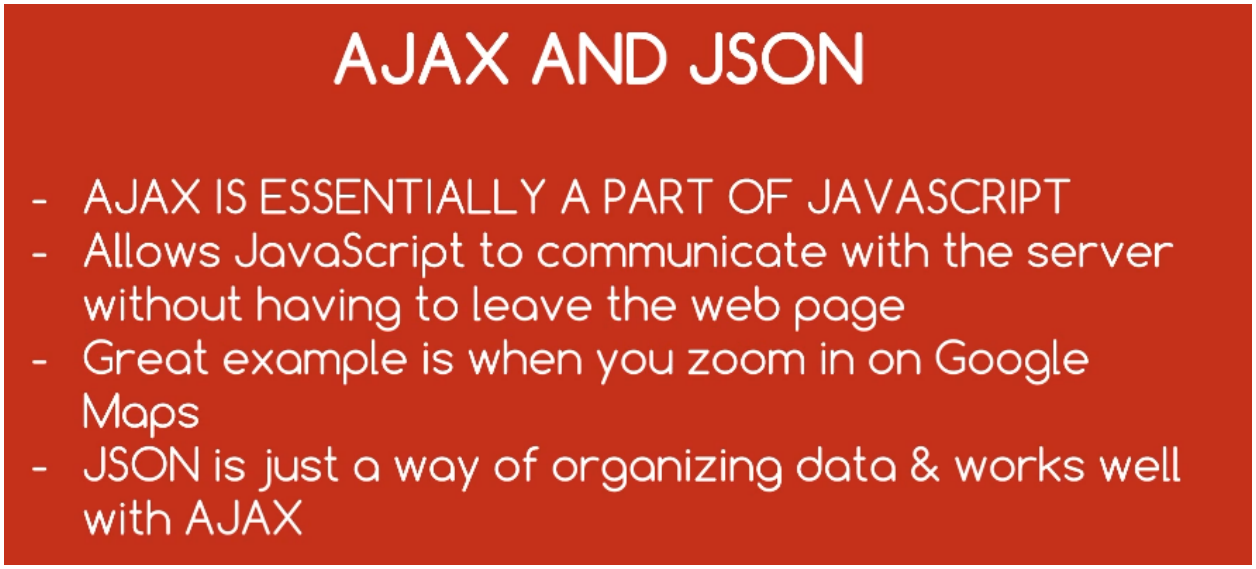


→Animate



46 What to Study Next





EXPLORING JAVASCRIPT FURTHER

- Lots in JavaScript that we haven't covered:
  - Regular expressions & advanced validation
  - OOP (object orientated programming)
  - Sliders, drag and drop, animated menu's