

# Programming for Cloud

Dr. R. Karthi

# Overview of the Lecture

- HTML
- CSS
- Java Script
- Node.js
- Express

# HTML

- HTML (Hypertext Markup Language) is the markup language that is used to structure a web page and its content.
- HTML consists of a series of elements or Tags.
- HTML Tags tells the browser how to format and display the web contents.
- World Wide Web Consortium (W3C) is an international community that work together to develop standard for web and Open Web Platform.
- HTML5 is the latest version or specification of HTML.

# Anatomy of an HTML document

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>My test page</title>
```

```
</head>
```

```
<body>
```

```
<h1>This is a Heading</h1>
```

```
<p>This is a paragraph.</p>
```

```

```

```
</body>
```

```
</html>
```

```
<!DOCTYPE html>
```

- ✓ It is an instruction to the web browser about the version of HTML the page is written.
- ✓ It ensures that the web page is parsed the same way by different web browsers.

```
< HTML >
```

- ✓ Element represents the root element of an HTML document.
- ✓ All other elements must be descendants of this element

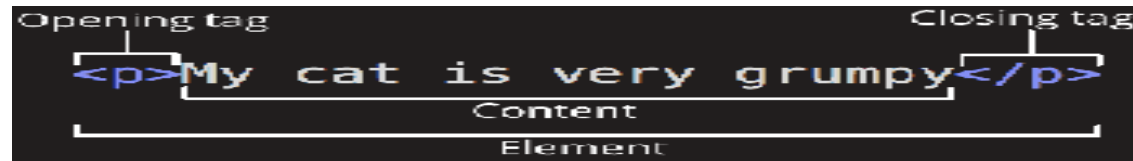
```
<HEAD>
```

- ✓ Element contains machine-readable information , meta data about the document, like its [title](#), [scripts](#), and [style sheets](#).

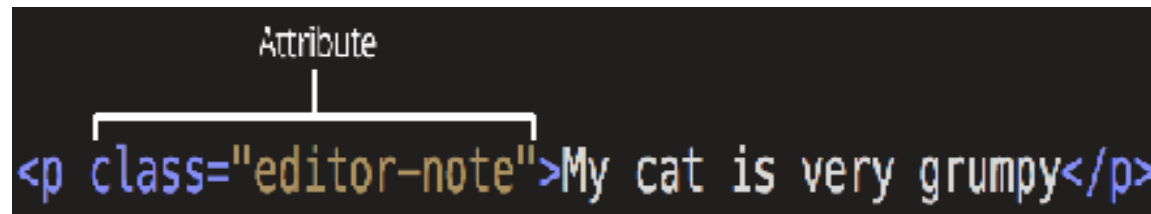
```
<BODY>
```

- ✓ Element represents the content of a HTML document.
- ✓ It is a is a container for all the visible components.

# HTML Tags



```
<p>My cat is <strong>very</strong> grumpy.</p>
```



```

```

src – is an attribute for img tag.

## Common HTML Tags

<code>&lt;title&gt;</code>	<code>&lt;h1&gt; ... &lt;h6&gt;</code>	<code>&lt;p&gt;</code>	<code>&lt;em&gt;</code>	<code>&lt;b&gt;</code>	<code>&lt;ol&gt;</code>	<code>&lt;ul&gt;</code>	<code>&lt;li&gt;</code>
<code>&lt;a&gt;</code>	<code>&lt;img&gt;</code>	<code>&lt;video&gt;</code>					
<code>&lt;table&gt;</code>	<code>&lt;tr&gt;</code>	<code>&lt;th&gt;</code>	<code>&lt;td&gt;</code>	<code>&lt;div&gt;</code>			
<code>&lt;form&gt;</code>	<code>&lt;script&gt;</code>	<code>&lt;textarea&gt;</code>					

# Form in HTML

- **HTML <form> element** represents a document section containing interactive controls for collecting and submitting information.
- Data can be send to the server for processing.
- Form contains elements which are different types based on the type of input elements.
  - Buttons, Checkbox, date, image , submit, text, password
    - `<input type="text" id="fname" name="fname" value="John">`
    - `<input type="submit" value="Submit">`
    - `<input type="email" name="email" id="email" required>`
    - **value** attribute specifies an initial value for an input field
    - **required** attribute specifies that an input field must be filled out before submitting the form.
    - **id** attribute specifies a unique id for an HTML element.

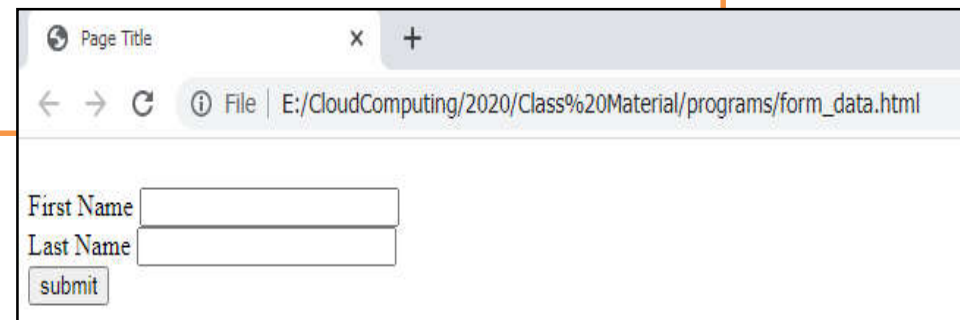
# Form in HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Page Title</title>
</head>
<body>
  <form action="http://127.0.0.1:8000/data" method="GET">

    <label for = "fname" > First Name </label>
    <input type = "text" id = "fname" name = "fname" > <br>
    <label for = "lname" > Last Name </label>
    <input type = "text" id = "lname" name = "lname" > <br>
    <input type = "submit" value = "submit" > <br>
  </form>
</body>
</html>
```

**action** - action attribute specifies where to send the form-data when a form is submitted.

**Method** - method attribute specifies how to send form-data



The screenshot shows a web browser window with a single tab titled "Page Title". The address bar displays the file path "E:/CloudComputing/2020/Class%20Material/programs/form\_data.html". The rendered form contains two text input fields labeled "First Name" and "Last Name", and a "submit" button.

# Cascading Style Sheets

- CSS(Cascading Style Sheets) is used to style web pages.
- A selector *selects* the HTML element that is to be styled. `<h1>`
- Style is applied to the selector using `property : value` pair.
  - `h1 { color: red; font-size: 20px; }`
    - h1 is the selector tag and color / font-size are properties
  - `p { color: red; text-align: center; }`
- CSS style using class definition
  - `.bg_clr { background-color: yellow; }`
    - `<p class="bg_clr">This paragraph refers to classes.</p>`
    - P is the selector tag which is assigned the bg\_clr class style
- CSS is applied on a web page using
  - Inline CSS
  - Internal or Embedded CSS
  - External CSS



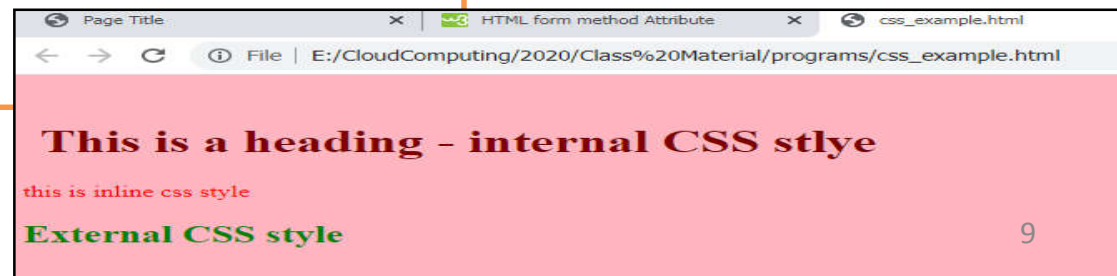
# Cascading Style Sheets

## Css\_example.html

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" type="text/css" href="mystyle.css">
  <style>
    h1{
      color:maroon;
      margin-left:10px;
    }
  </style>
</head>
<body>
  <h1>This is a heading - internal CSS stlye </h1>
  <p style="color:red"> this is inline css style </p>
  <h2> External CSS style </h2>
</body>
</html>
```

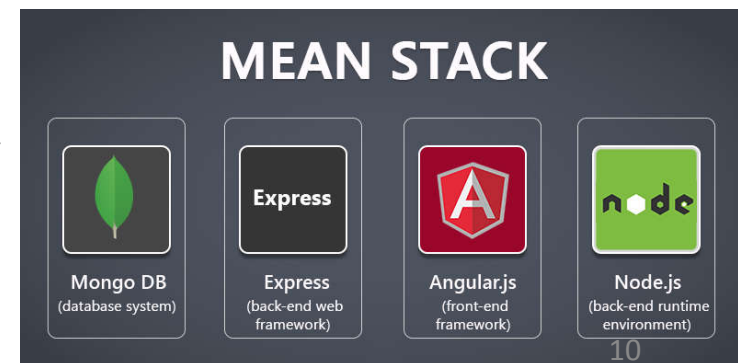
## mystyle.css

```
body
{
  background-color:lightpink;
}
h2{
  color:green;
}
```



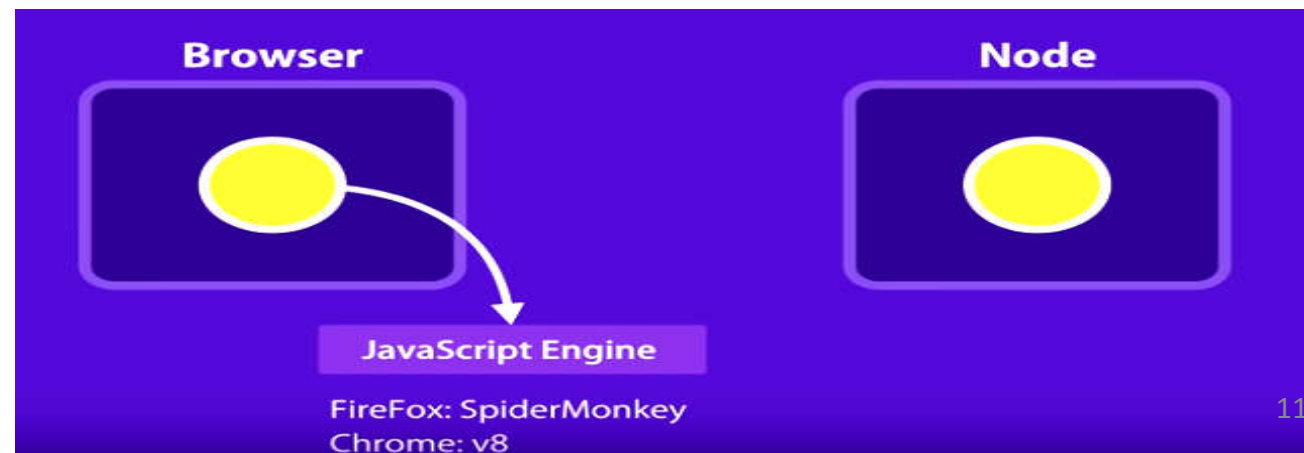
# Java Script

- JavaScript is a lightweight, interpreted, just-in-time compiled programming language that conforms to the ECMA Script specification.
- Need for JavaScript
  - HTML - define the content of web pages
  - CSS - specify the layout of web pages
  - **JavaScript - programs the behaviour of web pages and helps build web applications**
- Java scripts are used in designing of Web pages, desktop applications and server programs.
  - Example : Node.js , MongoDB and CouchDB use JavaScript as their programming language.
  - Netflix (Java Script with Node.js), wordpress Desktop (JavaScript with React )
- **Build a whole web application (from the UI to the data layer)** with a stack technology based only on JavaScript.
  - **MEAN** stack (MongoDB, Express.js, AngularJS, and Node.js)
    - MongoDB is a NoSQL database,
    - Express is a Node.js backend framework,
    - AngularJS is a front-end framework to create interfaces,
    - Node.js provide a JavaScript runtime Environment.



## Java Script Engine and Runtime Environment

- ❑ **Java Scripts** are run using browser or Node (JS runtime environment).
- ❑ **JavaScript engine** is a program responsible for translating source code into machine code and executing the translation result on a computer's central processing unit (CPU).
  - Chrome V8 - Google Chrome, Opera, NodeJS, and Couchbase.
  - SpiderMonkey - Firefox
  - Nitro - Safari
  - Chakra - Edge
- ❑ JavaScript engine works inside an **runtime environment**, which provides **additional features** to your scripts that you can use at runtime. (utility libraries or APIs)
- ❑ Runtime environment for executing java scripts



# Java Script Language Elements

- **Variables** – `var, let` - declare a variable (local / global)
- **Primitive data type** - JavaScript defines 5 types of primitive data types:
  - `String`        `"Hello Dave"`
  - `number`       `17 123.45`
  - `Boolean`      `true false`
  - `null`
  - `Undefined`
- **Operators** - `Arithmetic, comparison, assignment, boolean` - `Boolean(10 > 9)`
- **Control Structures** - `if if-else ?: switch`
  - `for while do-while`
  - `for (var in object)`
- **Functions** The keyword function is used to define a function.

```
function add(x,y) {  
    return(x+y);  
}
```

  - No type is specified for arguments
- **Objects** - The values are written as **name : value pairs** (name and value separated by a colon).
  - `var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};`
- **Array** - **variable which can hold more than one value at a time.**
  - `var cars = ["Saab", "Volvo", "BMW"];                      // literal type`
  - Methods – can be accessed using **dot** Operator (`sort, pop, push , length` )    `cars.length`
- **Built-in functions** – `Math.abs(x)`

# JavaScript

Javascript1.html

```
<!DOCTYPE html>

<html>

<head>

    <title> Document </title>

</head>

<body>

    <h1> My Page </h1>

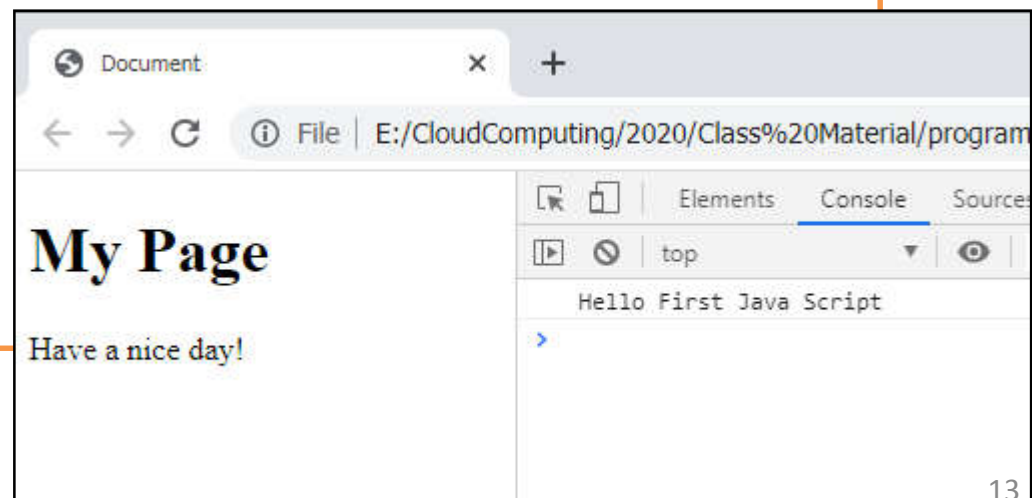
</body>

<script>

    console.log(" Hello First Java Script");
    document.write("Have a nice day!");

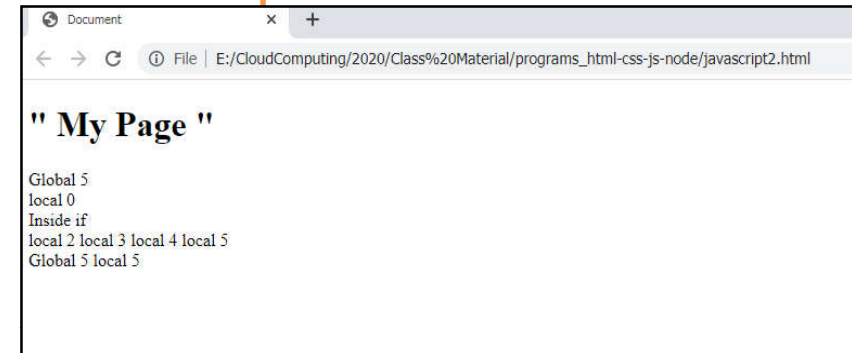
</script>

</html>
```



# Java Script

```
<!DOCTYPE html>
<html >
<body>
<h1> " My Page " </h1>
<script>
let globalVar = 5;
function fun() {
    let localVar = 0;
    document.write(" Global " + globalVar + " <br> ");
    document.writeln(" local " + localVar);
    if(localVar<globalVar)
    {
        localVar++;
        document.writeln( " <br> " + "Inside if " + " <br> ");
    }
    while(localVar<globalVar)
    {
        localVar++;
        document.writeln( "local " + localVar);
    }
    document.writeln("<br>" + " Global " + globalVar);
    document.writeln("local " + localVar);
}
fun(); // Function Call
</script>
</body>
</html>
```



# Declaring classes and constructors

Javascript3.html

```
<!DOCTYPE html>
<!DOCTYPE html>
<html>
  <head>
    <title> Constructor examples</title>
  </head>
  <body> <h1> Constructor </h1></body>
  <script type = "text/JavaScript" src = "construct.js">
  </script>
</html>
```

Construct.js

```
class student{
  constructor( nameval, ageval )
  {
    this.name = nameval;
    this.age = ageval;
  }
  display()
  {
    document.writeln("Name "+this.name + "
      Age " + this.age);
  }
}

var stud = new student("lee",23);
stud.display();
```



# Validation in forms

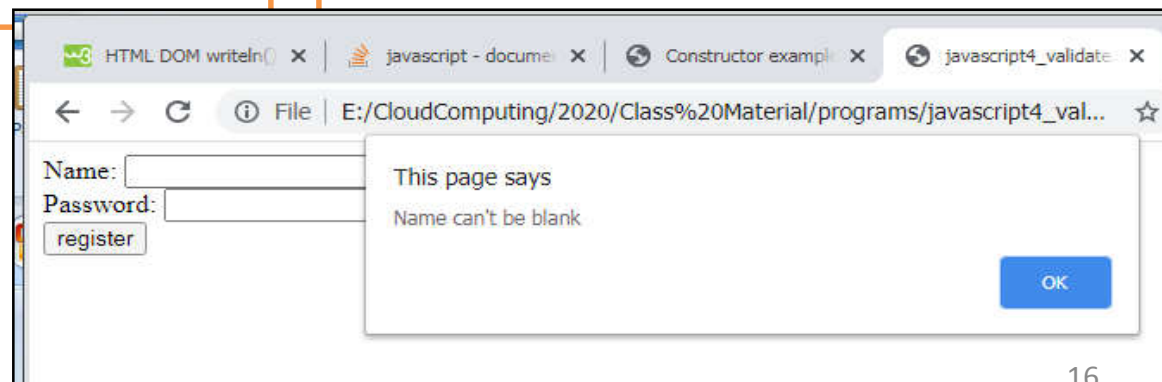
Javascript4\_validate.html

```
<DOCTYPE html>

<html>
<script type="text/javascript" src="validate_script.js">
</script>
<body>
  <form name="myform">
    Name: <input type="text" name="name"><br/>
    Password: <input type="password" name="password"><br/>
    <input type="submit" value="register" onclick = validateform()>
  </form>
</body>
</html>
```

validate\_script.js

```
function validateform(){
var name = document.myform.name.value;
var password = document.myform.password.value;
if (name == null || name == ""){
    alert("Name can't be blank");
    return false;
}
else if(password.length<6){
    alert("Password must be 6 char long.");
    return false;
}
}
```





# Running Java Script in Nodejs

- Node.js is an **open-source** and **cross-platform** JavaScript **runtime** environment.
- Node.js runs the **V8 Chrome** JavaScript engine.
- Node Installation
  - Step 1: Download Node.js Installer and run the installer.  
: <https://nodejs.org/en/download/>
  - Step 2: Verify installation : **node -version**
  - Step 3: Running Javascript from terminal: **node p1.js**
- Install **Visual Studio Code editor** for running your programs

# Java Script in Nodejs

ex1.js

```
class Rectangle {  
  constructor(height, width) {  
    this.height = height;  
    this.width = width;  
  }  
  area() {  
    return this.calcArea();  
  }  
  calcArea() {  
    return this.height * this.width;  
  }  
}  
  
const square = new Rectangle(10, 10);  
console.log(square.area()); // 100
```

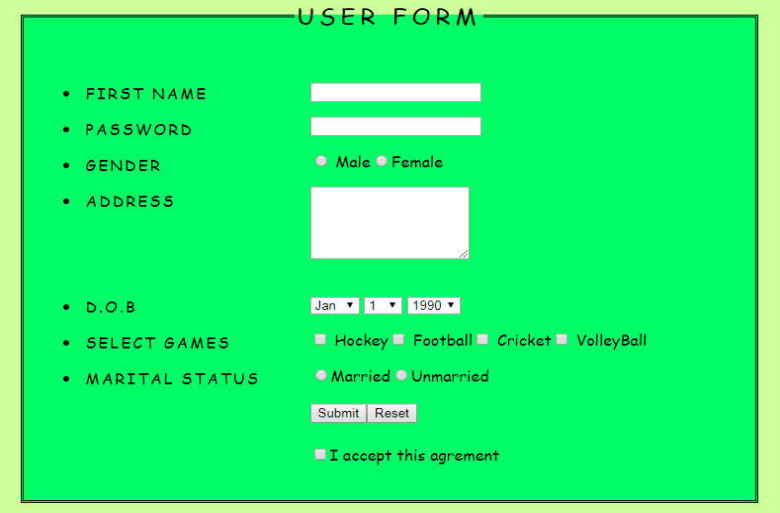
```
E:\CloudComputing\2020\JavaScript\Node-javascript>node ex1.js  
100
```

# Java Script Lab exercise

- 1) Create an array `arr1 = [1,3,5,10,1,3,2,9]`
  - a. Display the elements of the array.
  - b. Create an array `arr2` with the first three sorted elements of `arr1`.
  - c. Insert 15 into the array `arr1` at position 5.
- 2) Create a student record using `class : student { name, age, city, branch, batch}` that can hold 10 records.
  - a. Write function that will display students with `age > 19`.
  - b. Display students who are from `blr / che`.
  - c. Sort the students record based on branch and display them.

# Lab Assignment

- Create a web pages to display your personal information.
  - First pages display your personal information.
  - Second page display details of your school, college and other places of your interest, hobbies etc.
  - Use atleast 15 HTML tags & 5 CSS styles elements during creation.
- Prepare a web page of your Company
  - Basic Introductory Information of the company ( Text / images / hyperlinks), Contact Details with Address information
  - Menu / Tabs to other Information Like Employees / Products
  - Create a Link to **Form** that collect details from User on products they wish to order / feedback on product etc. and validate the form data
    - Sample form :



The image shows a sample web form titled "USER FORM" with a light blue background and a dark blue border. The form contains the following fields and options:

- FIRST NAME**: A single-line text input field.
- PASSWORD**: A single-line text input field.
- GENDER**: Two radio buttons labeled "Male" and "Female".
- ADDRESS**: A multi-line text area.
- D.O.B**: Three dropdown menus for month (Jan), day (1), and year (1990).
- SELECT GAMES**: Four checkboxes labeled "Hockey", "Football", "Cricket", and "VolleyBall".
- MARITAL STATUS**: Two radio buttons labeled "Married" and "Unmarried".
- Submit / Reset**: Two buttons at the bottom.
- I accept this agrement**: A checkbox at the bottom.

# Quiz

- <https://www.mentimeter.com/s/595f4d48cf62c138f9b6d1b8f4698db9/da681b2eec4c>