

Scripting Languages

-deepankar sharma

Unit 01

scripting languages → { Bash, NodeJS, Ruby, Python, Perl }

- Scripts and Programs,
- Origin of Scripting,
- Scripting Today,
- Characteristics of Scripting Languages,
- Uses for Scripting Languages,
- Web Scripting,
- the universe of Scripting Languages.

Advantages of scripting languages

- ① easy learning
- ② fast editing
- ③ Interactivity
- ④ functionality

Scripts and Programs

A script is a type of program that is usually interpreted rather than compiled. This means scripts are often executed by another program, whereas the programs can be executed directly by the computer's operating system.

Scripts are often used for automation tasks or adding interactivity to web pages. Programs can be used for wide range of purposes including standalone applications.

Some common example of scripts include Javascript code that runs in a browser or shell scripts that automate the tasks of a computer.

Programs can be written in different programming languages and can range from small utilities to large applications like MS Word / Adobe Photoshop.

Origin of Scripting Languages

- ① Scripting Languages were developed in 1980s as a way to automate tasks and simplify programming. 1980 → first scripting language → TCL (Tool Command Language)
- ② Perl → 1987 by Larry Wall, designed basically for text processing but became famous for web development in 1990s
- ③ Python (1991)
Ruby (1995)
JavaScript (1995) → Netscape "very flexible & adaptable"
1996 → CSS (Cascading Style Sheets)
→ separate presentation of webpage from its content
- ④ PHP → originally interpreted but now compiled often.
- ⑤ **HTML** (Tim Berners-Lee) in 1993
HTML 2.0 (1995)
HTML 3.2 (1997) ↓
HTML 4.0 (1997) HTML5 (2008) → new semantic tags for content structure
→ support for graphics, video, audio
1999 → W3C → XHTML (XML + HTML)

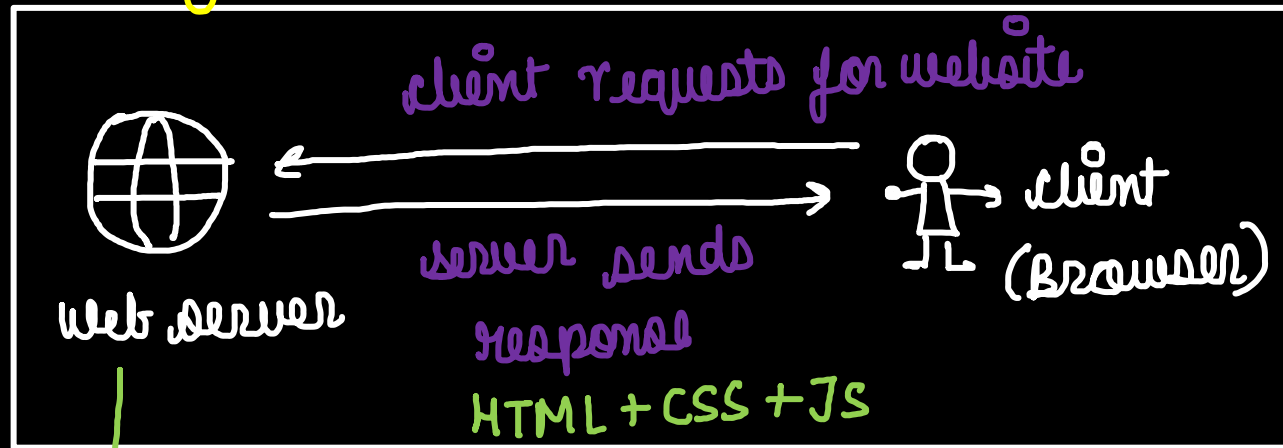
Unit 02

- Introduction: Concept of WWW, Internet and WWW,
- HTTP Protocol:
 - Request and Response,
 - Web browser and
 - Web servers,
- HTML Basic –
- Structure of HTML documents, HTML Elements, Linking in HTML, Anchor Attributes, Image Maps, Meta Information, Layouts, Tables, Audio and Video Support with HTML. Interactive Layout with Frames, FORMS, Form Control.

origin and characteristics of scripting languages (bash, ruby, nodejs, python)

" HTML + CSS + JavaScript "

body beauty brain



→ Backend

→ php

→ python

→ node JS

HTML (Hypertext Markup Language)

- standard markup language for giving a static skeleton to web application & websites.
- It is a well standardized system.

CSS (Cascading Style Sheets)

- style sheet language that is used to handle the presentation of webpage containing HTML.
- It makes our websites beautiful and modern looking.

Javascript (JS)

- high level dynamic interpreted programming language.
- allows client-side scripting to create completely dynamic web applications & websites.

meta tags → SEO optimization

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

```
<!-- head contains all the metatags -->
```

```
<head>  
  <meta charset="UTF-8">  
  <meta http-equiv="X-UA-Compatible" content="IE=edge">  
  <meta name="viewport" content="width=device-width, initial-  
scale=1.0">  
  <title>Lucky</title>  
</head>
```

Internet Explorer

```
<!-- body contains the content of the web page -->
```

```
<body>  
  Hello Lucky  
</body>  
</html>
```

Structure of HTML doc

head

meta tags

<!-- head contains all the metatags -->

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta name="description" content="This is description">

<meta name="keywords" content="html, python, web development">

<!-- <meta name="robots" content="NOINDEX, NOFOLLOW"> -->

<meta name="robots" content="INDEX, FOLLOW">

<!-- This is how we include external css -->

<link rel="stylesheet" href="temp.css">

<!-- This is how we include external js -->

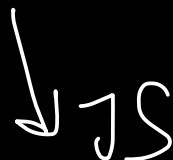
<script src="temp.js"></script>

<title>Lucky</title>

</head>



for browser



JS

Emmet ⇒ set of plugins for text editors for fast coding & editing HTML, XML etc

Heading Tag

```
<h1>Hello World</h1>  
<h2>Hello World</h2>  
<h3>Hello World</h3>  
<h4>Hello World</h4>  
<h5>Hello World</h5>  
<h6>Hello World</h6>
```

Paragraph Tag

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

Hello World

Hello World

Hello World

Hello World

Hello World

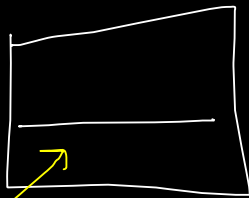
Hello World

`This is strong.` (default bold)

`This is emphasized.` (default italics)

`
` → break, new line

`<hr>` → horizontal rule



`<!-- not recommended after HTML5 -->`

`This is bold.`

`<i>This is italic.</i>`

use ``

use ``

```
<a href="https://google.com">Go to google</a>  
<a href="https://facebook.com">Go to facebook</a>  
<a href="https://twitter.com">Go to twitter</a>  
<a href="https://amazon.com">Go to amazon</a>
```

anchor tag
<a>

```
<a href="https://amazon.com" target="_blank">Go to amazon</a>
```

↗ internal document

↘ open in new tab

```
<a href="tut4.html">tut4</a>
```

image tag


```

```

↗ path to image
(url / url)

↘ if failed to load image, then show this

image tag

```
<img src = path alt = " " > </img>
```

Lists

↳ ordered list & unordered list

<ol type = '1'>

 This is first item

 This is first item

`<ul type="square">`

`This is the first item`

`This is the second item`

``

`This is first item of second-item list`

`This is second item of second-item list`

``

`This is the third item`

``

`<ol type="i">`

`This is the first item`

`This is the second item`

`This is the third item`

``

Square, disc, circle



} Nested list

I, l, a, A, i

- This is the first item
 - This is the second item
 - This is first item of second-item list
 - This is second item of second-item list
 - This is the third item
-
- i. This is the first item
 - ii. This is the second item
 - iii. This is the third item

Tables

```
<table>  
  <thead>  
  <tbody>  
</table>
```

HTML Table

Student Name	Student ID	University Roll No
Deepankar Sharma	20041299	2092014
Sooraj	20041245	2092045

```
<!-- Tables -->  
<h3>HTML Table</h3>  
<table border="1">  
  <thead>  
    <tr>  
      <th>Student Name</th>  
      <th>Student ID</th>  
      <th>University Roll No</th>  
    </tr>  
  </thead>  
  <tbody>  
    <tr>  
      <td>Deepankar Sharma</td>  
      <td>20041299</td>  
      <td>2092014</td>  
    </tr>  
    <tr>  
      <td>Sooraj</td>  
      <td>20041245</td>  
      <td>2092045</td>  
    </tr>  
  </tbody>  
</table>
```

Forms

Forms

Name:

Role:

Email:

Date:

Number:

Write about yourself:

Gender: Male ☐ Female ☐ Other ☐

Education: 10 ☒ 12 ☐ Graduation ☐

submit

reset

```
<h3>Forms</h3>
<form action="backend.php">
  <div>
    Name: <input type="text" name="myName">
  </div><br>
  <div>
    Role: <input type="text" name="myRole">
  </div><br>
  <div>
    Email: <input type="email" name="myEmail" id="">
  </div><br>
  <div>
    Date: <input type="date" name="myDate" id="">
  </div><br>
  <div>
    Number: <input type="number" name="" id="">
  </div><br>
  <div>
    Write about yourself: <br>
    <textarea name="" id="" cols="30" rows="5"></textarea>
  </div><br>
  <div>
    Gender: Male<input type="radio" name="" id=""> Female<input type="radio" name="" id="">
    Other<input type="radio" name="" id="">
  </div><br>
  <div>
    Education: 10<input type="checkbox" name="" id="" checked=""> 12<input type="checkbox" name="" id="">
    Graduation<input type="checkbox" name="" id="">
  </div><br>
  <div>
    <input type="submit" value="submit">
    <input type="reset" value="reset">
  </div><br>
</form>
```

Favourite TV Show:

```
<div>
  <label for="show_id">Favourite TV Show: </label>
  <select name="" id="show_id">
    <option value="1">The Flash</option>
    <option value="2">Arrow</option>
    <option value="3">Peaky Blinders</option>
    <option value="4" selected>The Witcher</option>
  </select>
</div><br>
```

Label & select

Block & Inline Elements

→ full block → content wrap

<p>

<p style="border: 2px solid red;">This is a paragraph</p>
<p style="border: 2px solid blue;">This is a paragraph</p> } Block Element

This is a span
This is a span } Inline Element

This is a paragraph

This is a paragraph

This is a span This is a span

ID and classes

```
<div id="mainbox" class="redBg blackBorder">  
  <!-- id (identifier) -->  
  This is mainbox  
</div>
```

```
<!-- Emmet -->  
<!-- . is for class and # is for id -->  
<span class="redBg"></span>  
<!-- span#mainId + <Tab> -->  
<span id="mainId"></span>  
<!-- div.hello.world.class4 + <Tab> -->  
<div class="hello world class4"></div>
```

```
<!-- Emmet takes div tag as default -->  
<!-- .blackBaground + <Tab> -->  
<div class="blackBackground"></div>
```

```
<!-- Multiple of the tags -->  
<!-- .class1.class2*4 + <Tab> -->  
<div class="class1 class2">hello</div>  
<div class="class1 class2">hello2</div>  
<div class="class1 class2">hello3</div>  
<div class="class1 class2">hello4</div>
```

HTML Entities

* Reserved **

This is a paragraph

This is a paragraph

paragraph tag is <p>

pound is written like this £

copyright is written like this ©

this is written like this →

this is written like this ¼

Empty character is written like this

```
<div class="container">
  <p>This is a paragraph</p>
</div>
<div class="container">
```

Non breaking space

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

```
<!-- <> -->
```

```
<p>paragraph tag is &lt;p> </p>
```

```
<p>pound is written like this &pound; </p>
```

```
<p>copyright is written like this &copy; </p>
```

```
<p>this is written like this &rarr; </p>
```

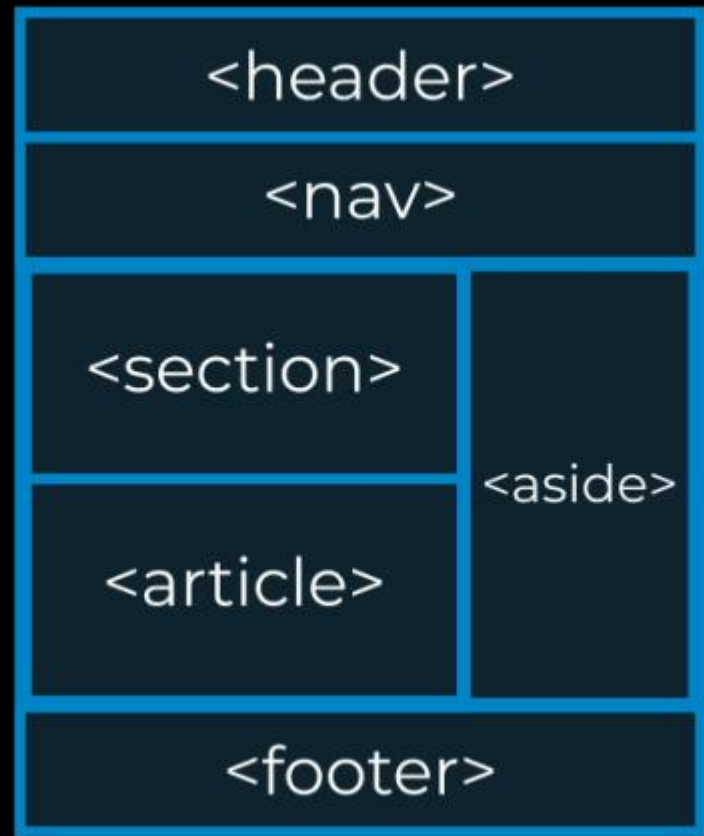
```
<p>this is written like this &frac14; </p>
```

```
<p>Empty character is written like this &#8203; </p>
```

```
</div>
```

```
<h3>Semantic tags</h3>
  <details>
    <summary>Details</summary>
    Lorem ipsum dolor sit amet consectetur
    adipisicing elit. Reprehenderit, ipsum esse enim
    tempore dignissimos vero maiores deleniti a
    obcaecati neque maxime adipisci est nostrum, magnam,
    sed consequatur quibusdam quasi mollitia. Aut nobis
    reiciendis porro consequuntur exercitationem tempore
    quod corporis assumenda numquam. Nesciunt explicabo
    accusantium sequi, minima laborum saepe illo
    accusamus est. Beatae mollitia, alias maxime
    voluptate provident dicta quod quae.
  </details>
```

Semantic Tags



Semantic tags

▼ Details

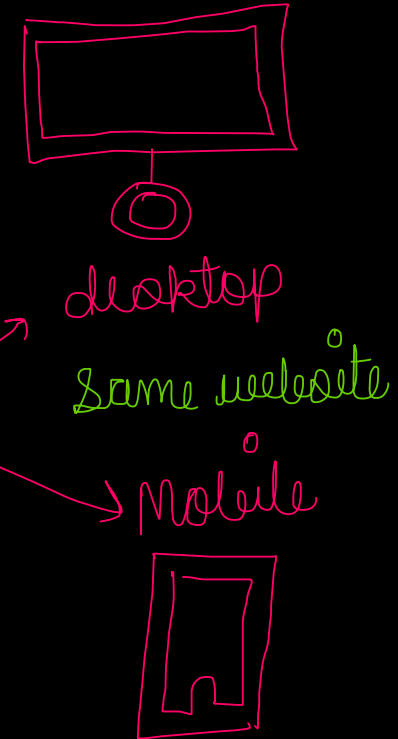
Lorem ipsum dolor sit amet consectetur adipisicing elit. Reprehenderit, ipsum esse enim tempore dignissimos vero maiores deleniti a obcaecati neque maxime adipisci est nostrum, magnam, sed consequatur quibusdam quasi mollitia. Aut nobis reiciendis porro consequuntur exercitationem tempore quod corporis assumenda numquam. Nesciunt explicabo accusantium sequi, minima laborum saepe illo accusamus est. Beatae mollitia, alias maxime voluptate provident dicta quod quae.

Unit 03

- CSS:
- Introduction,
- Benefits of CSS,
- types of CSS,
- Selector and types,
- text formatting properties,
- Box Model concept,
- CSS Border,
- margin properties, Positioning, color properties, Classes in CSS, concept of Ids pseudo classes.

What is CSS? → used to handle presentation of web page containing HTML
→ makes our websites beautiful & modern looking.

- ① CSS gives style to raw HTML
- ② CSS stands for Cascading Style Sheets
- ③ CSS is used to give style to our web pages
- ④ CSS is used to make websites responsive.
- ⑤ HTML is used to structure a website
- ⑥ CSS takes responsibility of design



Syntax

paragraph tag ← ^{selector} P { ^{property} color : ^{value} blue ; }

↑ where ↑ this ↑ set this value

selector

• class name

id name

tag name

header tag ← ^{paragraph with class intro} header, p.intro {

group of selectors ↓

background-color : red ;

border-radius : 3px ;

} Declaration Block

3 ways to add CSS to the MARKUP

`<p style="color: red;" > ... </p>`

↓ HTML code

① Inline CSS CSS is added to the elements directly using style attribute.

② Internal CSS CSS is kept inside the head tags in `<style>` tags

③ External CSS CSS is kept separately inside a .css style sheet.

- ① create a separate .css file
- ② include that in the markup

```
p{  
  color: aqua;  
  background: blue;  
}
```

```
<link rel="stylesheet" href="tut13.css">
```

Adding CSS to HTML: Inline & Internal

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Tutorial</title>
  <style>
    p{
      color: purple;
    }
  </style>
</head>
<body>
  <h3>CSS Tutorial</h3>
  <p style="color:red;background-color: yellow;">This is a great CSS Tutorial.</p>
  <p>This is another paragraph</p>
</body>
</html>
```

Internal CSS

Inline CSS

CSS Tutorial

This is a great CSS Tutorial.

This is another paragraph

Adding CSS to HTML: External

tut13.css

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Tutorial</title>
  <!-- <style>
    p{
      color: purple !important;
      /* !important-> wants to take precedence otherwise the inline CSS will take precedence */
      background-color: black;
    }
  </style> -->
  <link rel="stylesheet" href="tut13.css">
</head>
<body>
  <h3>CSS Tutorial</h3>
  <!-- <p style="color:red;background-color: yellow;">This is a great CSS Tutorial.</p> -->
  <p>This is another paragraph</p>
</body>
</html>
```

link

p{
color: aqua;
background: blue;
}

Unit 01

Unit 01

Unit 01

Unit 01

