Web Application Development (WAD)

N Pratik
Patil

UNIT-I → Introduction to Web Technologies

Q1] What is HTML? Give reason why we use HTML.

→ HTML stands for Hypertext Markup Language.
It is the standard language used to create & design web pages. It provides the structure of web pages by using tags to define heading, images, link, tables and other content.

* Why HTML?

- HTML is very famous for creating pages that are displayed on World.Wide.Web (www)
- Browser friendly → HTML supported by all browsers eg- google chrome, opera mini.
- free to use → HTML is free and open source markup Language. you dont need to buy single line of code.
- Easy to learn → It is very basics of web that even school students also uses it.
- Simple Structure → Coding Structure is very simple to understand.
- Works with CSS & JS → works very efficiently with them.

Q] Explain the tags used in HTML with their attributes

→ 1) <a> (anchor Tag)

purpose :→ Create Hyperlink to navigate between web pages.

Attributes

href → Specify the URL of a Link

target → Define where to Opens the link.

eg → click here

2] <Img> (image tag)

purpose → Embeds image on web page

Attributes → src → specify the image URL

alt → provides alternative information if image is not displayed.

width & height → Image dimensions.

eg → <img src = " " alt = " " ></img>

3] <p> (paragraph tag)

purpose → Define a paragraph in HTML

Attributes → align → Specify text alignment (L, R, C)

] <h1> to <h6> (Heading tool)

purpose → Define heading from large to Small.

align : specifies text alignment.

5] <input> (input tag)

purpose → Create input field within a form

attributes → Type → define the input type (pass, email)

6] (division Tag)

purpose → Acts as a container for styling or layou changes.

Attributes → id → unique identifier

class → define a class of css style.

Q] Write a note on Bootstrap Grid System.

→ The Bootstrap Grid System has the capacity to generate 12 columns & countless rows. Also Known as twelve-grid system.

The Grid System is made of three components :→

1) Container → Acts as a wrapper for rows & columns. It ensures proper alignment.

2) Rows → Groups Columns together
- Ensures proper alignment and spacing between Columns.

3) Columns :→ Defines width of contentent within a row.

Working mechanism of Bootstrap Grid System

1] Container holds the Grid (.container or .container-fluid)
- The grid system must be wrapped inside a Container
- .container provides fixed-width layout.
- .container-fluid - provides full-width layout.

2] A Row (.row) groups the columns.
- Each row is divided into 12 equal columns by default.

3] Columns (.col-*) Define the layout
- Columns define how much space they occupy in 12 column layout.
- You can specify different column sizes for diffrent screen width.
- If no size is specified (.col) columns will share equal width automatically

8] Explain bootstrap Component.
→ Bootstrap provides ready-made UI component, that make web development faster & easier.

Some of the components of Bootstrap are

1] Alerts
- Used to display message like success, warning or error.

eg → Success

Type → alert-success, alert-danger, alert-info

2] Buttons
Used to create clickable buttons.

<button class="btn btn-primary">primary</button>
<button class="btn btn-danger">danger</button>

3] cards → A container for text, images & Links

This is card
.

~~4] Navbar - A navigation bar for menus~~

4] Forms → pre styled input fields & buttons.

<input type="text" class="form-control" placeholder="Name">

5] Badges → Small lable for notifications.

New.

6] Tables → Styled tables for displaying data.

<table class="table table-stripped">---</table>

Q] What is W3C? How W3C handles/supports Web Technologies? List advantages & disadvantages

→ World Wide Web consortium (W3C) is an organization that sets rules & standard for how website should be build & work on internet.

- It was founded in 1994 by Tim Berners-Lee. the founder of Web.
- The mission of W3C is to lead the World wide web to its full potential. that ensures the long term growth of web
- One of the primary Goal of W3C is to make it available for all people 'whatever their hardware, software, language or physical or mental ability is.

* How W3C handles/supports web Technologies

There are two main groups that handle the web standars → Community Group & Working Group

1) Community Group → It is such kind of group where the initial innovation happens around new web Technologies.. New web standards can be produced by community group but they are officially seen as pre-standard. Community groups are open for all peoples.

- For new ~~feeds~~ Standard the ~~new~~ dissussion stards in community group after the discussion has begun it moves to organization. This is repeated until dissussion becomes proposal, proposal becomes draft & draft become standard.

2] Working Group → It is a group where new standards are designed officially. Working group starts with a submission on a standards. After performing analysis & working, this standard goes in recommendation phase. in W3C.

Advantages

1] Standardization → Ensures all websites follows the same rules.

2] Cross-Brower Compatability - make websites work smoother on all browsers.

3] Improved Accessibity → Helps disabled user access web content easily.

4] Security & Standard → Safer web browsing

Disadvantages

1] Slow process

2] Complex.

3] Not all browser supports it.

Q] What are different Selectors in CSS explain with example

→ CSS selector are used to target HTML element & apply styles.

1] Universal Selector (*)

→ Select all element on the page

```
ex → *{ margin:0;
        padding:0;
      }
```

2] Element Selector

→ Target specify HTML Tags.

```
P {
    color: blue;
    font-size: 16px;
  }
```

3] Class Selector (.)

- Select element that has specific class.

```
.button{
  background-color: green;
}
```

4] ID selector (#)

- Target single element with specific ID

```
#header {
  background-color: black;
  color: white;
}
```

5] Group Selector (,)

- Styles multiple element at once.

```
h1, h2, h3 {
    font-family: Arial;
  }
```

6] descendent Selector

select an element inside another element

```
div P {
  color: red;
}
```

7] Child selector (>)

- Select direct children of element.

```
div > P {
  font-family: Arial;
}
```

8] Adjacent Sibling Selector (+)

- Selects the next element immediately.

```
h1 + P{
  color: orange;
}
```

9] General Sibling Selector (~)

- Selects all siblings of an element.

```
h1 ~ P {
  font-family: Arial;
}
```

Q] Differentiate between CSS & Bootstrap.

Feature	CSS	Bootstrap
1) Definition	A stylesheet language used to style HTML elements	A CSS Component with pre-defined style & component.
2) Complexity	Requires manual styling for each element	In is ready-to-use state.
3) Customization	Fully Customizable but requires more effort.	Customizable but come with predefined styles.
4) Ease of use	Need to write code from scratch	Predefined works make it easier.
5) File size	lighter : as only necessary style are written	Heavier due to the complete framework.
6) Learning	Requires Excellent Knowledge in CSS	Easier for beginner but Knowledge useful.
7) Usage	Used when Custom styling is needed.	for faster development with standard design.

Q] Explain the box modeling of CSS.

→ The CSS box model defines how every HTML element is structured in terms of content, padding, border & margin.

Every HTML element is treated as rectangular box with following Component.

1) Content → • Actual content inside the element (img, text, etc.)
• Defined by height & width.

ex→
```
div {
  width: 200px;
  height: 100px;
}
```

2] Padding → • Space inside the box, between the content & border.
• Add extra spaces without affecting other element.
• Controlled using padding properties.

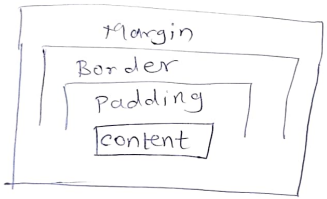
eg→
```
div {
  padding: 10 px;
}
```

3] Border → • Visible line that cover content & padding
• Defines the edge of element.
• solid, dashed, doted & double are common style.

eg→
```
div {
  border: 5 px solid black;
}
```

4] Margin → • Space outside the border, between the element & other element.
• Controlled using margin properties

```
div {
  margin : 20px;
}
```



Types of Box Models :-

1] Standard Box model

- It is a default box model.
- The width & heigh apply only on content.
- Padding & border are added outside. increases the total size of element.

eg →
```
div {
  width : 100px;
  height : 90px;
  padding : 10px;
  Border : 5px solid black;
}
```

2] Alternative Box Model

- Helps in maintaining fixed box size without increasing dimension
- The width & heigh includes padding & border.

```
div {
  width : 100 px
  height : 90px
  padding : 10px
  border : 5px solid black;
  box-sizing : border-box;
}
```

Q] Difference between checkbox Group & Radio button Group.

feature	checkbox group ☑	Radio Button Group ◉
1) Slection	Allows Multiple Selection	Allows only one selection.
2) Usage	Used for selecting multiple ~~only~~ options (eg hobbies)	· Used for choosing only one option (eg - gender)
3) HTML Inputtype	<input type = "checkbox">	<input type = "Radio">
4] Grouping	Each box works independently	All radio buttons with the same name work as group
5] Deselecting	User can check/uncheack anytime.	User connot uncheck once selected.

Q1] What is inline, embedded and external JavaScript? Explain the advantages & drawbacks

→ i) Inline Javascript

- Inline JavaScript is written directly within an HTML element's attribute, usually inside onclick, onmouseover, or other event attribute

```
<button onclick = "alert("Hello World!")>Clickme
</button>
```

Advantages →

1] Easy to use
2) Faster execution

Disadvantage.

1] Difficult to maintain: mixing JS with HTML makes it messy
2] Not reusable → Have to write same code multiple time.

2] Embedded (internal) Javascript.

In ~~embedded~~ Internal JS, the script is written inside the <script> tag within the HTML file, usually in the <head> or <body> section

```
<!Doctype Html>
<html>
<head>
    <script>
        fn showMessage() {
            alert ("Hello, World!");
        }
    </script>
</head>
<body>
    <button Onclick ="showMessage()">Click Me</button>
</body>
</html>.
```

Advantage

1] Easy to manage
2] Avoid need of multiple http requests

Drawback

- Complex
- duplication may occur.

3] External JS

External JS ~~is~~ is stored in seperate .js file and linked to the HTML file using <script> tag

```
<!Doctype html>
<html>
<head>
    <script src = "script.js"></script>
</head>
<body>
    <button onClick ="showMessage()"> Click Me</button>
</body>
</html>
```

script.js

```
fn showMessage() {
    alert ("Hellooo world");
}
```

Advantages

- Maintainability & scalability
- Reusability.
- Excellent page reload speed

Disadvantages

- Requires ~~external~~ External HTTP request to fetch the JS file.
- Security risk.

Q] Explain Javascript object in detail.

→ • An object is a JS is a collection of key-value pair, where key are properties (variable) & valve can be data (number, string, arrays, fn)

• Object allows us to group related data & fn together, making Javascript code more structured & organized.

There are three main categories in object
→ JS Buildin object, Browser Object, Dom object.

1] Buildin Objects

1) Number Object → Allows you to work with primitive numeric valves.
• A Number object are created using Number() (constructor.

~~2) Not Math Object →~~

2) Math Object → Used to perform mathematical task.
eg→ min(), max(), round(), sqrt(), tan(), cos().
→ Math.random() // returns random no. between 0 to 1.

3) Date Object → Used to handle dates & times
Syntax for creating date is using Date() fn
var d = new Date()

7 arguments in date
1) Year 2) month 3) Date 4) Hour 5) min 6) sec 7) mil-sec

methods → setMonth(), getMonth().

4] Boolean Object

Boolean represent valves either true or false.

5] String Object

Use to work with series of character. The string object of JS allows you to perform manipulations on a stored piece of text.

eg→ split(), search(), subString(), replace()

2] Browser Object

1) Window Object→
• window object is top level object in JS and contains itself several other object such as "document", "history" etc...

2) Location Object →
Contains info about the current URL of browser

3) History Object → • History of URL ~~visited~~ visited

Q] What is Async-Await

→ • Async & Await are modern JS features to handle asynchronous operation in simpler and readable way. They are used to work with promise but with a syntax that look synchronous.

• Before callback & promise worked to handle asynchronous event but the Callback hell & promise made the code difficult to understand & readable. However Async-Await handles it like synchronous code.

How does it work

- Async make a fn return a promise
- Await pauses the fn until the promise is resolved.

```
async function hello() {
    return "Hello World!";
}
hello().then(console.log);
```

o/p → Hello World!

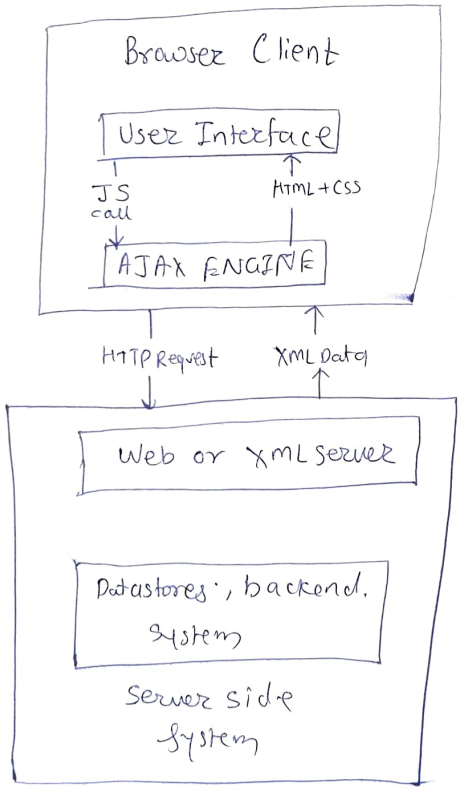
Q] Write JS that read an integer & display whether it is a prime no or not

→

```
let n = PaseInt(prompt("Enter a number"));
let isPrime = true;
if (n<=1) {
    isprime = false;
}
else {
    for(let i=2; i<n; i++) {
        if (n%1==0) {
            isprime = false;
            break;
        }
    }
}
alert (isPrime ? `${n} is a prime number` : `${n} is
    not a prime number`);
```

Q] Write short note on AJAX based, web application architecture.

→ AJAX → Asynchronous Javascript & XML is a technique that allows webpage to update asynchronously by exchanging data with a web server in the background. This means part of webpage can be updated without reloading the entire page.



Explaination of diagram

1] Browser Client
- UI interacts with the user
- JS makes a AJAX call to request data
- The UI updates dynamically with HTML + CSS Data

2] AJAX Engine
- Acts as intermediary betⁿ the UI & the server
- Send HTTP request to the server
- Receive XML or JSON data from the server

3] Server-Side (Backend System)
- Web / XML Server process request.
- Interacts with databases, backend
- Sends the requested data back to AJAX engine

4] Data Update in Browser.
- The AJAX engine update only the required part without ~~updating the system~~ refreshing whole page.

8] Explain disadvantage of using callback f^n & how it can overcome using promise with example.

→ Callback f^n's are commonly used in JS for handling Asynchronous operations but come with several disadvantages.

1] Callback ~~Hell~~ Hell

- when multiple nested callbacks are used, the code became hard ~~to~~ to read & maintain

example →

```
getuser(1, function(user){
    getuser(user.ID, f^n(posts){
        getcomments(posts[0].id, f^n
                    (comments){
            console.log(comments);
        });
    });
});
```

- As the nesting increases code become unreadable.

2] Difficult Error Handling.

- Handling error in deeply nested callbacks requires passing errors manually.

eg.

```
getuser(1, function(error, user){
    If(error){
        console.log("error");
    }
    else{
    getpost(user id, (error, posts){
      If (error){
          console.log("error");
      } else{
          console.log("posts");
      }
    });
});
```

3) Inversion of control

- In callback execution depends on another f^n.
- It means it have to be dependent to being executed correctly.
- If callback not executed, it may break the logic.

Overcomming callback hell using promices

. A promices in JS is an object that represent a future value (successful or failed) of an asynchronous operation. instead of nesting callbacks, we use then() chaining

Callback

```
fn task(callback){
    setTimeout(()=>{console.log("Task done");
      callback();},1000);
}
task(()=>console.log("New task"));
```

Sol^n Promise

```
fn task(){
    return new promise(resolve => setTimeout(()
      =>{console.log("Task done"); resolve();},1000));
task().then(()=>console.log("Next task"));
```

Q] What is JQERY? Explain with example. List & explain the form selector used & disadvantage of JQERY

→ JQERY. List advantages