



HTML

- Hyper Text Markup language
- It is used to create web pages & web applications
- It is widely used language on the web
- We can create a static website by HTML only
- It is a markup language.

<!DOCTYPE html>

<html>

<head>

<title> web page title </title>

</head>

<body>

<h1> heading 1</h1>

<p> paragraph </p>

</body>

</html>

HTML tags

- tags are like keywords which defines that how web browser will format & display the content.

DATE / /



Eg:- ① unclosed html tags.

↳
 - break

↳ <hr> - horizontal rule

② Meta tags

↳ DOCTYPE,

↳ title

↳ link

↳ meta & style

③ Text tags

↳ <p>, <h1>, <h2>, <h3>. -

↳

↳ <var>

④ Link tags

<a>, <base>

CSS

↳ cascading style sheet

↳ used to design html tags.

↳ CSS is a widely used language on web

↳ It allows to apply the style on html tags.



CSS3 with bootstrap

combining CSS3 & bootstrap allows developers to create visual appealing, responsive websites with ease.

how to use.

1. Setting up bootstrap

including bootstrap.css &.js file in our html document. we can use a CDN (content delivery network) for simplicity.

2. Creating a responsive layout with bootstrap grid

```
<div class = " container ">  
  <div class = "row">  
    <div class = "col-md-4"> column 1 /div>  
    <div class = "col-md-4"> column 2 /div>  
    <div class = "col-md-4"> column 3 /div>  
  </div>  
</div>
```

DATE / /

State Bank of India
THE BANKER TO EVERY INDIAN

3) Using CSS3 for custom styles.

Create a custom CSS file to add additional styles using CSS3.

Eg:- /* styles.css */

body {

font-family: Arial;

background-color: #ffffff;

}

custom-box {

font-size: 10px;

font-color: red;

border-radius: 10px;

}

JS ES6 getting started with Node.js

Node.js allows us to run JS on the server.



Creating a simple Node.js application

1. Initialize a new Node.js project.
* create a new directory for our project

mkdir my-node-app

cd my-node-app

npm init -y

→ this will create a package.json file with default settings.

2. Create an entry point

Create a file named app.js

~~code~~ → touch app.js

3. Write a basic Node.js Server

Open app.js in your text editor
and add

```
const http = require('http');
```

```
const server = http.createServer()
```

```
((req, res) => {
```

DATE

/ /



```
res.statusCode = 200;  
res.setHeaders({'content-type':  
'text/plain'});  
res.end('Hello, world.');
```

```
});  
  
const port = 3000;  
server.listen(port, () => {  
  console.log('Server running at  
  http://localhost:' + port + '/');  
});
```

4. Run your server

```
node app.js
```

Open a web browser & go to
local host to see
Hello World!



Angular JS

- ↳ It is a open source JS framework that is used to build web applications.
- ↳ It is developed by google.

Advantages

- ↳ Dependency injections specifies a design patterns
- ↳ Two way data binding.
It creates a two way data-binding b/w the selected element & the OrderProp model
- ↳ Testing
it is easy to test any of its components through unit testing & end-to-end testing
- ↳ Model View controller

AngularJS module

- It defines an application.
- It is a container for the different parts of your application like
 - ↳ controllers
 - ↳ services,
 - ↳ Filters
 - ↳ directives etc.

How to create a module

```
<div np-app="myApp">..</div>
<script>
var app = angular.module("myApp",[])
</script>
```

AngularJS directives

- ↳ these are nothing but html with new attributes, these attributes are called AngularJS directives.

↳ Build-in-directives - which adds functionality to our application

↳ user-defined - directives.



Directives

↳ `np-app` → starts an angularJS app

↳ `np-init` - initializes app's data

↳ `np-model` - defines the model

↳ `np-repeat` → repeats html elements
for each item in a collection.

Ex :-

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<title> AngularJS directive </title>
```

```
</head>
```

```
<body>
```

```
<h1> Sample app </h1>
```

```
<div np-app="" np-init="countries =  
[ { locale: 'IND', name: 'India' },  
{ locale: 'PAK', name: 'Pakistan' } ]">
```

```
<p> Enter your name <input type="text"  
np-model="name"> </p>
```

```
<p> hello <span np-bind="name"> </span>  
! </p>
```

```
<p> list of countries with locale </p>
```

DATE

/ /

State Bank of India
THE BANKER TO EVERY INDIAN

 np-repeat = "country in countries">
 {{'country': ' + country.name + ', 'local': ' +
 country.locale}}

</div>

<script> src = "http://googl.com"></script>

</body>

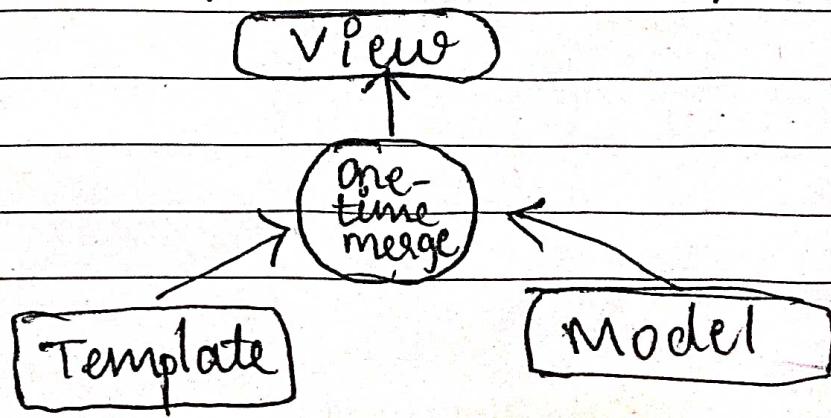
</html>

Binding database

- It acts as a bridge b/w the view and business logic of the appn
- It is useful & powerful.

There are

↳ One-way - data binding.





- * The one-way-data binding is an approach where a value is taken from the data model & inserted into an html element.
- * These systems bind data in only one direction.

Two-way-data binding.

