1) INCEPTION

React is a library.

```
const heading=React.createElement('h1',{id:"title"},"Hello World!");
const root=ReactDOM.createRoot(document.getElementById{"root"));
root.render(heading);
```

React elements are objects. React backside takes arguments and converts it into Objects.
We inject React inside the root.
(generally, we have 1 root and 1 render method)

Render will manipulate DOM.
React overrides whatever is inside root div.

# Mostly,we say <div id="root">Not rendered</div>, so that if react isn't able to load we will see Not rendered.
Script tag is inside the body tag.

There is a very little lag in loading react, because browser takes some time to load websites, and then starts reading code.

React came with the idea of writing html and css inside JS file only. So that we don't have to go to different files.

1) Read about Emmet
2)Read about CDN
3)Cross-origin
4)DOM
5) Difference between async and defer
6)check for classname,id, style, null
7)

EMMET

It is a web development tool that helps in writing easy, quick, efficient code.
For example , pressing ! in the html file in vscode, gives us a dummy html code.

LIBRARY vs FRAMEWORK

Both are reusable code written by someone else.

In Libraries , developer has inversion of control. We control the flow of code/application. We decide when to call certain functions. Libraries are flexible, we can use only some functions as per our requirement. We don't need to use entire library Libraries are typically smaller in scope and size. They focus on specific functionalities	Framework dictates flow of control. It come with a structure when we just have to inject our code. We don't control when to call functions. Framework has a set structure, it is not flexible as we can not choose to use only some component. It is more extensive and provide a broader set of tools and conventions for building applications.

CDN : Content Delivery Network

Distributed network of servers around the world. Jab user internet pe kuch search karta hai toh information server se internet pe travel karke aati hai uspe, agar server bahot hi door hoga toh delat hota hai response aane mei. Toh ye servers geographically placed hain aur ispe website content served hai, taki used ko paas ke server se jaldi data mil jaye.

> **Faster Loading Times:** CDNs reduce latency by serving content from servers closer to the user, resulting in faster loading times for websites and web applications.
> **Scalability**: CDNs distribute the load across multiple servers, making it easier to scale resources and handle increased traffic, especially during peak periods or traffic spikes.
> **Reliability**: By having multiple servers geographically distributed, CDNs provide redundancy and ensure that content is still accessible even if one server goes down.
> **Cost Savings**: CDNs can help reduce the load on your origin server, potentially lowering hosting costs by offloading the delivery of static assets to the CDN.
> **Security**: Some CDNs offer security features, such as DDoS protection and SSL/TLS support, helping to enhance the overall security of web applications.

Instead of downloading and hosting the React library locally, you can include it directly from a CDN.

```
<!-- React -->
<script src="https://unpkg.com/react@17/umd/react.production.min.js"></script>
<!-- React DOM -->
<script src="https://unpkg.com/react-dom@17/umd/react-dom.production.min.js"></script>
```

● Why is React known as React?

React is called react because it reacts rapidly. It reacts as soon as there is some change in data or state. Change dekhke react karti hai aur UI pe efficiently dikha deti hai.

● What is crossorigin in script tag?

The crossorigin attribute is typically used with scripts fetched from different domains, especially when using a Content Delivery Network (CDN) to host scripts. It helps prevent certain security vulnerabilities, such as cross-site request forgery (CSRF) attacks, by controlling how the browser handles cross-origin requests.

● What is diference between React and ReactDOM ?

REACT is a library jisse UI based app banate hain.
Class, element, ye sab use karte hain toh React use hoti hai.

REACTDOM bhi ek complimentary library hai jisse DOM se interact karte hain hai.It is responsible for rendering DOM components in browser.ReactDOM.render() is a main function.

● What is difference between react.development.js and react.production.js files via CDN?

Development wali development ke liye use hoti hai.

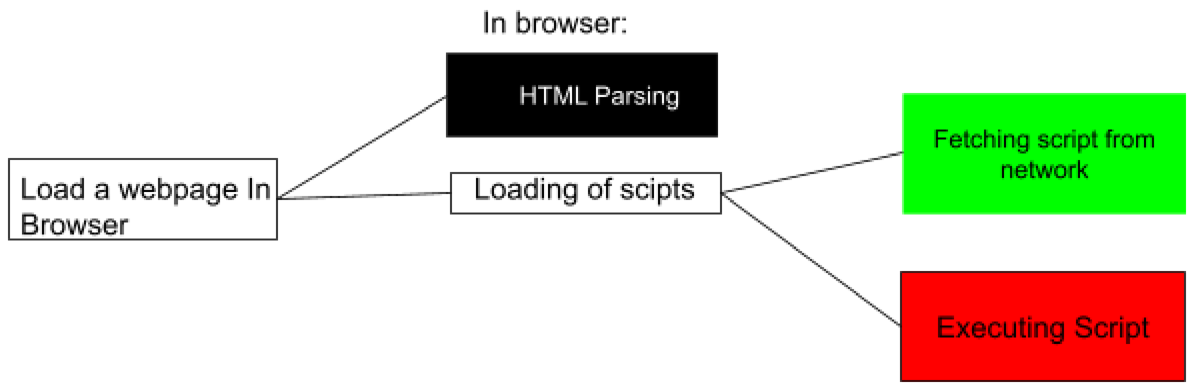
- Size bada hota hai. Performance utni fast nhi hoti
- Debugging info, checks, warnings hoti hai.
- Mostly local mei use krte hain.

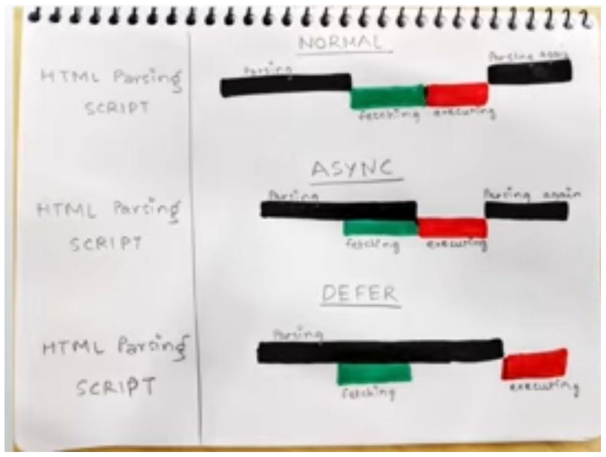
Production wali production mei use hoti hai.

- Minified size hota hai. Better performance.
- Faltu ki info, checks, warning nahi aaati.

What is async and defer?

Both are boolean attributes used with script tag to load external scripts efficiently to our webpage.



 NORMAL HTML Parsing SCRIPT parsing fetching executing Parsing again ASYNC HTML Parsing SCRIPT Parsing fetching executing Parsing again DEFER HTML Parsing SCRIPT Parsing fetching executing	If scripts are interdependable, its best to use Defer tag, as it executes scripts only after html is loaded.

2) IGNITING OUR APP

Bundler

Props are properties/attributes.
Production Ready: minify, bundle things, need server,optimize, caching, etc….
We can use many libraries in react. It's very flexible.

Bundler: webpack, vite, parcel.
Very helpful.
Create react app uses webpack.
Parcel,vite is a package. Package is a module of JS. So, we need package manager.
We will use npm. We can also use yarn.
NPMis used to manage our packages. We use it because we require many packages.
npm init -> to initialise npm (npm init -y) (-y to skip configuration) Gives package.json. (make it compatible with git)
npm i **-D** parcel _____ -> -D means dev dependency. We want only on our machine not in production. (-D is same as --save-dev). Means, our project is dependent on this package.

Dependency vs Dev Dependency:
Dependency production time pe chahiye hoti hain project run karke ke liye.
Dev Dependency, bas hume local mei chahiye hoti hai development phase mei bs.

Symbol before version in package.json:
~tilde - allows minor updates (4.0.1 -> 4.0.3 but not 4.1.0)
^caret - allows major updates (4.0.1 ->4.1.0 but not 5.0.0)
No sign before version - No updates allowed

package-lock.json - tells the exact versions used by our project.
Its a very important file. Never keep in gitignore.

Node_modules is database of npm. Everything is stored in it. It has helper functions.It is very heavy.Keep it in gitignore. Package.json, package-lock.json have full information to regenerate node_modules.
We put it in gitignore because its very huge, so our git generates it .
Node_modules is always there with our git.

Dist folder mei distribution/production ready code hota hai. Mostly js+css. Optimized code hai. Ready for deployment wala.

Parcel needs some space.
.parcel-cache: parcel uses this to do all things. This is a space
Dist folder: creates a dev build for us and hosts it on server.
Parcel uses many things to minify our app. Keep in gitignore. Because can be generated.
It is a folder that the Parcel bundler uses to store temporary files and information while it's building your web project.
Mtlb agar koi change pahle hi ho chuka hai toh ye cache use store krega aur jaise wo dobara karna hoga hume toh ye cached wala output dega, toh ye faster hota hai.

To execute project-> npx parcel entrypoint.file
Npx means execute using npm. Npx is a package runner tool that comes with npm. Bina download ke bhi easily run karne ke ke liye kaam aata hai npm.js wale packages ko. Ya fir temporarily store karke run karta hai

Parcel automatically refreshes page.
In js file, to use react we need to import react and reactDOM.

WHAT PARCEL DOES?
Hot Module Reload (HMR)
File Watcher Algorithm (written in c++)
Bundling
Minimizing
Cleaning our code
Super fast build algorithm
Dev and production build
Image optimization
Caching while development : takes time first time, then caches)
Compression
Compatible with older versions of browsers.
Https on dev (npm parcel index.html --https)
Consistent hashing algo
Zero config bundler (no configuration required, just need to install and give entry point, it will figure out everything on its own)

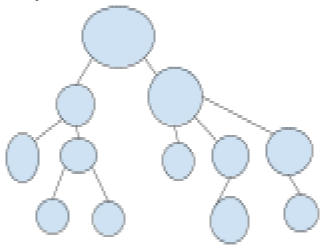
Media takes most space

Transitive dependencies (package manager takes care of all transitive dependency)
#**ReactIsFAST** (React can't create performant apps alone)
(react use bundler, bundler use dependencies to make it fast and efficient)
(parcel uses Packages in node_modules are used).
React→bundler —->packages, dependencies—-> use more dependencies —->more dependencies.

Read parcel documentation. (features)

Script types (module)

There is a package-lock.json in node_modules which has versions of all these dependencies.



-----------> Transitive dependencies

Babel is also a dependency used by parcel

To make app compatible with all browsers, we use browserlist. Parcel has it.
We write it in package.json. →

```
"browserList":[
    "Last 2 versions"
  ]
```

So now, all browser's last two versions will be supported.

● - What is Tree Shaking?

Nundlers hamari js file se Dead, unwanted aur unused code hata dete hain taaki jab wo js wala bundle browser pe jaye toh wo minimized ho. Agar koi library import ki aur usme 20+ helper function hain but humne 3 hi use kiye toh wo unused walo ko ignore kar dega.
For example unused functions wtc, sab hata dete hain.

● - Hot Reloading?

Jab file mei kuch changes hote hain toh parcel use dobara se build karke, browser pe update kar deti hai.

● - Hot Module Reloading / Hot Module Replacement?

Jab bahot hi minor changes hote hain toh pura page refresh nahi hota aur app ki state lose nahi hoti, bas wo module update hojata hai.aur immediately changes reflect hojate hain.
Ye js bundlers ka hi feature hai. Running app mei hi update kardeta hai.

3.LAYING THE FOUNDATION

Polyfill: a code which is a replacement for a newer version of code. (eg: if browser doesn't know array.map() so it will be replaced with another older code that it will understand.
ES6 concepts are new. If we work on a older version of browser, it won't understand new concepts.
Const, let, .map, promises …. are new concepts.

Babel: it is a node package that converts code which converts the code ununderstandable by bowser to a alternative code which will be understood by browser. So, its a piece of code that reads our code, and converts into another code.

New code —------> babel—-------------> polyfill

Browserlist used.

Npm parcel index.html is long to write everytime so we write
"scripts": {
   "start":"parcel index.html",
} in package.json.
So now we can do npm run start or npm start everytime.
npx == npm run

#(if we want parcel to remove console logs also, we need to configure it using a babel plugin which is called : babel-plugin-trandform-remove-console)
Make .babelrc for configuration.

React uses keys to uniquely identify elements we need to give key in the props.
Create react element gives an object which is converted into HTML and inserted into dom.
Using create react element for all html, complex nested is very difficult because we will need so many createREactElement ans it will be a mess.

createReactElement —-->object—------>html dom

```
1) const heading2 = React.createElement(
    "h2",
    {
      id: "title",
      className:'heading2',
      key:"h2",
    },
    "Heading 2"
  );
```

So, **JSX was introduced.** JSX was built by FB developers who developed React. Its not a package. Its a syntax.

```
2) const heading2jsx=<h2 id="title" key="h2">Heading 2 using jsx</h2>
      //or
  const headingjsx=(               //these are called jsx expression
  <h2 id="title" key="h2">
      Heading 2 using jsx
  </h2>
  );
```

If single line no need to () else put jsx code in ( ).
JSX is a html like syntax but its not html inside js.

id is in html, key in React. React only recongnises key and not id.
In jsx, names are in camelCase only.
In React, we give className for class.

Superpowers of JSX:
1)Readability
2)Embedding js
3)Reusable components
4)Inline styling ....

BABEL:

Its a library, javascript compiler.
Js file ke andar wala html code browser nahi smjhta but babel smjhta hai.
Babel line by line code padhta hai aur check karta hai html hai kya hai, aur us gisab se abstract syntax tree banata hai.
Babel come as a dependency when we do npm i parcel. We can see it in node_modules.

How does JSX executes?

Jsx uses React.createElement behind the scenes which then createReactElement —>object—>html dom
Babel understand our jsx and converts code number 2 to code number 1. (Check out babel.io)
We don't code in React, we code in jsx.
Babel is the most imp. It reads our code superfast and converts.it into js.

Difference between html and jsx.

Read about babel

Abstract syntax tree

How to create img tag in jsx

Components

Everything is a component in React.

1) Class Based Components   // old way
2) **Functional Components**   //new way

Functional component are Js functions that return (jsx)a function or component.
React element bas ek js object hota hai jo dom mei add hota hai but component ek function hota hai

Header is a component.

```
const Header =()=>{
    return (
      <div>
        {heading}// element
        <Heading/> or {Heading()} //component
        <h1>Hello World!</h1>
        <h2>Hello Kriti!</h2>
      </div>
    )
    };
```

If Heading is an element. We use element inside component like **{Heading}**
Component composition/Composing components: using component inside a component(nested component)
If Heading is an component. We use element inside component like **<Heading/> or {Heading()}**

We can write any js code inside **{ }.**
All the code which is inside { } is sanitized by jsx so that it can be protected from XSS(Cross-Site Scripting) attack.