

this ⇒ keyword.

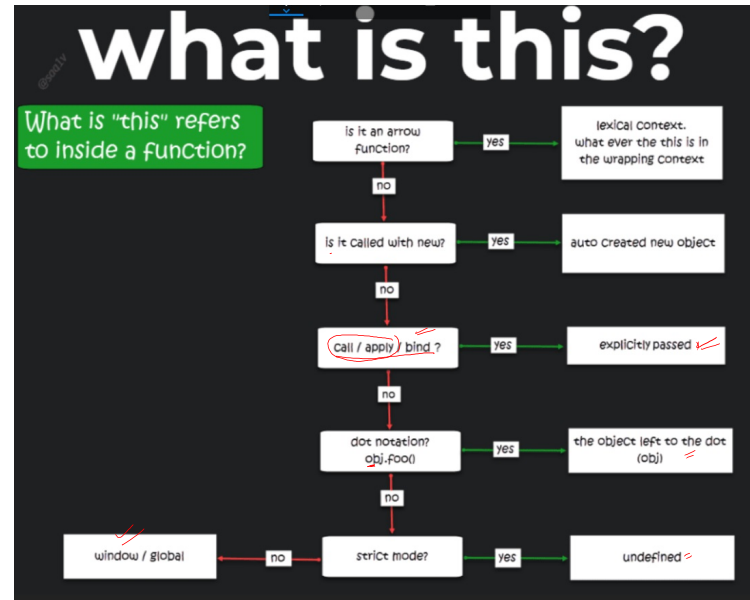
↳ global ⇒ this → window obj.

↳ function calls.

X [→ fn ⇒ how that fn is called.
↳ normally → window obj.
↳ obj → obj.
↳ defn ⇒ call → window.

↳ Bind

↳ Jis b fn bind call hoga, uske defn copy krega,
jo naya dega uska this jo argument pass hia h
uske equal kr deta h.



React
 ↳ It is a JS library for building UI. Developed in Facebook.

Why React?

↳ event listeners

↳ easy manage (shortcut)

↳ page & snapshot

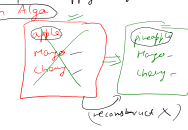
↳ Virtual DOM & Reconciliation Algo

items: [apple, mango, cherry]

↳ loop

[apple, mango, cherry]
 ↳ loop

↳ change very fast detect
 ↳ apply only change



↳ React → library → API very small → easier to learn

↳ UI ⇒ state manage

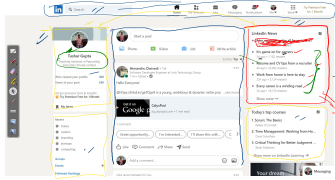
↳ UI ko manage

Components ⇒ components are independent, isolated & reusable piece of code just like our functions

Components are of 2 types

↳ class components

↳ Function components

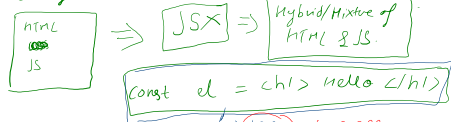


isolated
 independent
 reusable
 function(bh, tl, readers)
 <div>
 </div>

babel

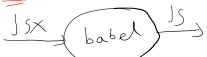


↳ change



↳ JSX ⇒ easy

browser understand JSX
 browser only knows JS



babel is basically transpiler/compiler ⇒ JSX → JS

Static UI → Functional Component (X) (Hooks)

↳ user interaction?

Class Components

↳ State

↳ The state is a built in React object that is used to contain data & information about the component.

↳ Render ⇒ Screen (UI display) (first time ⇒ code ko UI p)

↳ Whenever our state, component re-render

↳ re-render ⇒ hmare comp. changes ko UI p reflect.

Counter Application:-

