REACT BASICS

* + *Data Binding*React uses one-way data binding and an application architecture called Flux controls the flow of data to components through one control point – the dispatcher. It's easier to debug self-contained components of large ReactJS apps.
  + *Performance*React does not offer any concept of a built-in container for dependency. You can use Browserify, Require JS, EcmaScript 6 modules which we can use via Babel, ReactJS-di to inject dependencies automatically.

* + *Simplicity*

ReactJS is just simpler to grasp right away. The component-based approach, well-defined lifecycle, and use of just plain JavaScript make React very simple to learn, build a professional web (and mobile applications), and support it. React uses a special syntax called JSX which allows you to mix HTML with JavaScript. This is not a requirement; Developer can still write in plain JavaScript but JSX is much easier to use.

* + *Easy to learn*

Anyone with a basic previous knowledge in programming can easily understand React while Angular and Ember are referred to as ‘Domain specific Language’, implying that it is difficult to learn them. For react you just need basic knowledge of CSS and HTML.

* + *Virtual Document Object Model*

React creates an in-memory data structure cache which computes the changes made and then updates the browser. This allows a special feature which enableprogrammer to code as if whole page is render on each change where as react library only render components which actually change.

* + *Single-Way data flow*

In React, a set of immutable values are passed to the components renderer as properties in its HTML tags. Component cannot directly modify any properties but can pass a call back function with help of which we can do modifications. This complete process is known as “properties flow down; actions flow up”.

