

REACT JS

Components

Sreekanth M. E.

Freelance Trainer & Consultant

<http://www.SreekanthME.com>

Components let you split the UI into independent, reusable pieces, and think about each piece in isolation.

The simplest way to define a component is to write a JavaScript function.

This function can have an optional argument of an object named "props".

Conceptually, components are like JavaScript functions.

They accept input arguments (called "props") and return React elements describing what should appear on the screen.

Components written with a JS function are termed
Functional Components.

Functional components don't have an internal state.

They are also called **stateless functional components**
or **dumb components** or **presentational**
components.

Functional components don't support state or lifecycle methods.

They perform better due to their stateless nature.

Stateful Components or Smart Components
can be written using ES6 class syntax.

State provides components with behavior.

Note: Smart components will be covered in detail in future chapters.

Presentational components focus on the UI rather than behavior, so it's important to avoid using state in presentational components.

Instead, state should be managed by higher-level “container” components, or via state management specialists like Flux/Redux/etc.

React Virtual DOM is created from React Elements.

React elements can be HTML DOM tags or (user-defined / custom) components.

Components can themselves be made up of other HTML DOM Tags or Components.

Components must return a single root React element.

Component names must start with a capital letter. This is a convention.

This distinguishes them from HTML DOM tags which are always in lowercase

END OF CHAPTER

Several thin, white, parallel lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

APPENDIX