



Components & JSX Cheat Sheet

Sección 3, Clase 33

Components are the **core building block of React apps**. Actually, React really is just a library for creating components in its core.

A typical React app therefore could be depicted as a **component tree** - having one root component ("App") and then an potentially infinite amount of nested child components.

Each component needs to return/ render some **JSX** code - it defines which HTML code React should render to the real DOM in the end.

JSX is NOT HTML but it looks a lot like it. Differences can be seen when looking closely though (for example `className` in JSX vs `class` in "normal HTML"). JSX is just syntactic sugar for JavaScript, allowing you to write HTMLish code instead of nested `React.createElement(...)` calls.

When creating components, you have the choice between **two different ways**:

1. **Functional components** (also referred to as "presentational", "dumb" or "stateless" components - more about this later in the course) =>

```
const cmp = () => { return <div>some JSX</div> }
```

 (using ES6 arrow functions as shown here is recommended but optional)
2. **class-based components** (also referred to as "containers", "smart" or "stateful" components) =>

```
class Cmp extends Component { render () { return <div>some JSX</div> } }
```

We'll of course dive into the difference throughout this course, you can already note that you should use 1) as often as possible though. It's the best-practice.

Recursos de esta clase

components.pdf

