Chapter 01 Inception

recap by @jx

1 React Element(s)

An element is a lightweight description of a piece of the user interface. This is the basic building block in React.

Syntax: createElement(type, props, ...children)

```
const heading = React.createElement(
    "h1",
    {
      id: "title",
    },
    "Heading 1"
  );
```

2 Rendering Element(s)

2.1 createRoot

createRoot lets you create a root to display React components inside a browser DOM node. Commonly, we use a 'div' with it's 'id' property set to 'root'.

This tells React to take over managing the DOM inside it. Applications built with just React usually have a single root DOM node.

Syntax: createRoot(domNode, options?)

```
const root = ReactDOM.createRoot(document.getElementById("root"));
```

Note 1. An HTML file must exist with a 'div' html tag in it and id set to 'root'.

2.2 render

After you've created a **root**, you need to call root.render() to display a React element inside of it,

Syntax: root.render(reactNode)

```
root.render(heading);
```

Note 2. A 'reactNode' can be a piece of JSX or a React element constructed with createElement(), a string, a number, null, or undefined.

Note 3. If the React element was previously rendered into container, this will perform an update on it and only mutate the DOM as necessary to reflect the latest React element.

3 Library vs. Framework

A library/framework is a bunch of re-usable code that other developers have written for you.

We imported React library using CDN links(for simplicity) to play with it's APIs(createElement, createRoot etc.).

When using a library, you are in charge of the flow of the application. You are choosing when and where to call React in your code. This is different from using a framework as it provides some places for you to plug in your code, but it calls the code you plugged in as needed.

4 References

More on 'createElement' here.
More on 'render' here.
More on 'createRoot' here.
The Difference Between a Framework and a Library by Brandon Wozniewicz