

## COMPONENT VS. INSTANCE VS. ELEMENT

### Component

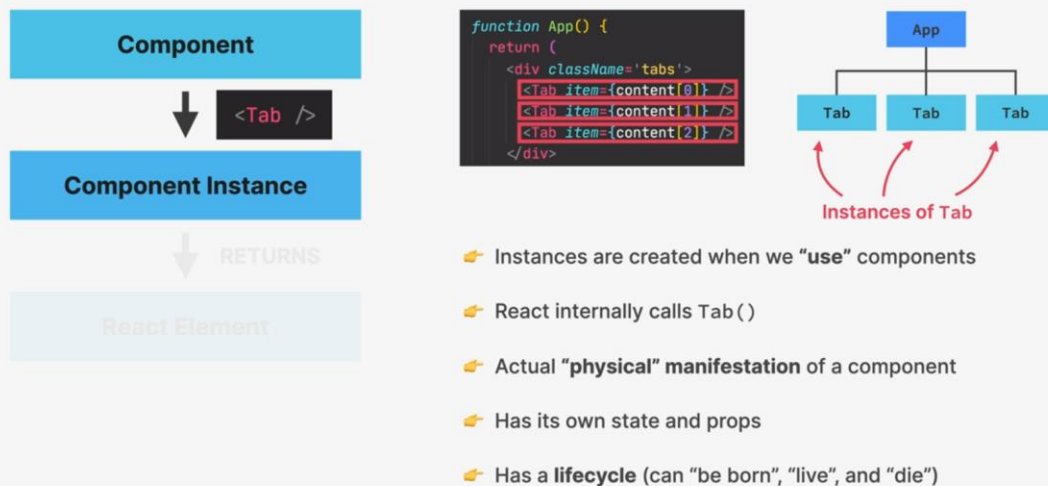
```
function Tab({ item }) {  
  return (  
    <div className='tab-content'>  
      <h4>All contacts</h4>  
      <p>Your post will be visible</p>  
    </div>  
  );  
}
```

- 👉 Description of a piece of UI
- 👉 A component is a function that **returns React elements** (element tree), usually written as JSX
- 👉 "Blueprint" or "Template"

Scrimba

1. Components are what we write in order to describe a piece of the user interface.
2. The component is just a regular JavaScript function but it's a function that returns React elements. So, it returns an element tree. We usually write these elements using the JSX syntax.
3. A component is a generic description of the UI. We can essentially think of a component as a blueprint or a template, and it's out of this one blueprint or template that React then creates one or multiple component instances.

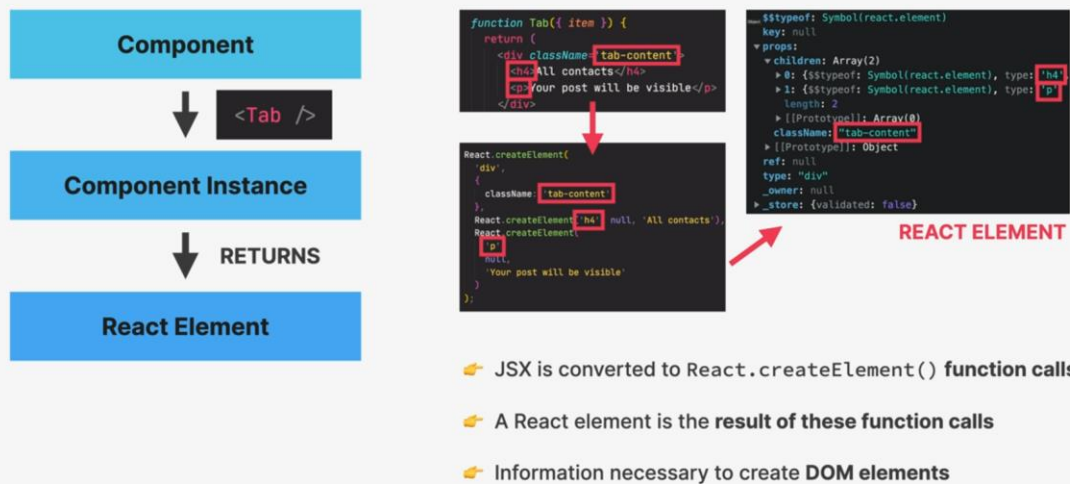
## COMPONENT VS. INSTANCE VS. ELEMENT



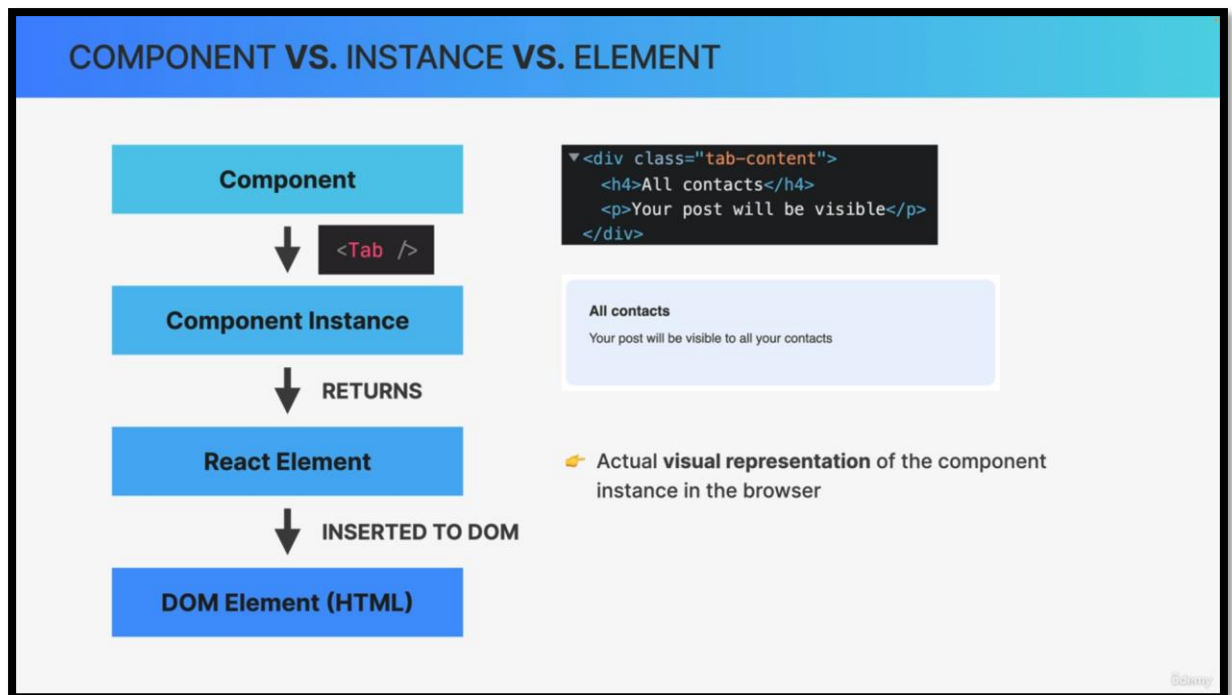
1. React does this each time that we use the component somewhere in our code. For example, the `tab` component that we saw in the last slide is used, so it is included three times in this `App` component. Three instances of `tab` are placed in a component tree.
2. In our actual application. Behind the scenes, this happens because React will call the `tab` function three times. So, one time for each instance.
3. An instance is like the actual physical manifestation of a component living in our component tree.
4. It's each instance that holds its own state and props and that also has its own life cycle.
5. A component instance can be born, it can live for some time until it will eventually die. So, it's a bit like a living organism really.

In practice, we many times just use the terms component and component instance interchangeably. For example, we just say component life cycle and not component instance life cycle. We also say that a UI is made up of components, not of component instances, even though instances would technically be more accurate. So, just keep that in mind in the future when you read documentation or some stack overflow post or something like that.

## COMPONENT VS. INSTANCE VS. ELEMENT



1. As React executes the code in each of these instances, each of them will return one or more React elements.
2. JSX will actually get converted to multiple `React.createElement` function calls. React calls these create element functions the result will be a React element. So, a React element is basically the result of using a component in our code.
3. It's simply a big immutable JavaScript object that React keeps in memory.
4. A React element basically contains all the information that is necessary in order to create DOM elements for the current component instance.



React element that will eventually be converted to actual DOM elements and then painted onto the screen by the browser.

The DOM elements are the final and visual representation of the components instance in the browser.

It's not React elements that are rendered to the DOM. React elements just live inside the React app and have nothing to do with the DOM. They are simply converted to DOM elements when they are painted on the screen in this final step.