

🚩 Current Skill Create Components

Creating our first component:

After all that talk about components, it's time to get our hands dirty.

Let's create our first react component!

Creating a component is much easier than you think. In practice, a component is simply a function that returns JSX, that's all we need to create a component.

Using JavaScript's power, we can manipulate these functions as we wish. We are going to create the component Greeting that we've seen before. It's very simple. Just follow the instructions:

1. Create the project in the src folder,
2. Open the App.js,
3. Then, copy-paste this code.

```
import React from "react";

const App = () => {

  return (

    <>

      <h2>Hello from my first component !!</h2>

    </>

  );

};

export default App;
```

Output



Hello from my first component !!

Congratulation ! You have created your first component. And YES! The **App.js** is also a component. We will get to know it more in the next slides.

What should we remember?

There are a few things that you should notice about that first example, the first one is the **import React from "react"**; well it's mandatory since we work with JSX and the **export default App**; this is also a mandatory action that must be performed. We'll explain its necessity. Remember the course on ES6, and how it made our life much easier as a developer. Time to rewind and take advantage of it.

📄 In order to separate the components from each other, we have to use the JavaScript modularity.

React also give us the possibility to use the named exports and the default exports

The last thing that you should notice is that the component is starting with an uppercase letter. That's necessary so we do not confuse it with simple HTML tags.

Default export

Similar to JavaScript file, we can export our component, as the example below shows, using the keyword **export default**.



```
const MyFirstComponent = () => {  
  return (  
    <>  
    <h2>Hello from my first component !!</h2>  
    </>  
  )  
}
```



```
);  
};  
  
export default MyFirstComponent;
```

By doing that we've made the file **MyComponent.js** export automatically the component `MyFirstComponent`.

Named exports

The other way to export a component is by the named export. Let's take a look at the example.

This is how we can export more than one component from a single file.

```
import React from "react";  
  
export const MyFirstComponent = () => {  
  return (  
    <>  
      <h2>Hello from my first component !!</h2>  
    </>  
  );  
};
```



```
export const MySecondComponent = () => {  
  return (  
    <>  
      <h2>Hello from my second component!! </h2>  
    </>  
  );  
};
```



Best Practices for Creating a Component:

After creating our first component, here is a list of some common best practices that we should know:

- Keep components small and function-specific:



- Function-specific components can be standalone, which makes testing and maintenance easier.
- Each small component can be reused across multiple projects.
- With smaller components, it's easier to implement performance optimizations.
- It's easier to update smaller components.
- Bigger components have to perform harder and may be difficult to maintain.

- Name the component after the function:

It's a good idea to name a component after the function that it executes so that it's easily recognizable.

- Keep them all in a single folder:

Keep all files relating to a component in a single folder, including styling files.

- Reusability is important.

[< Previous](#)

[next >](#)