

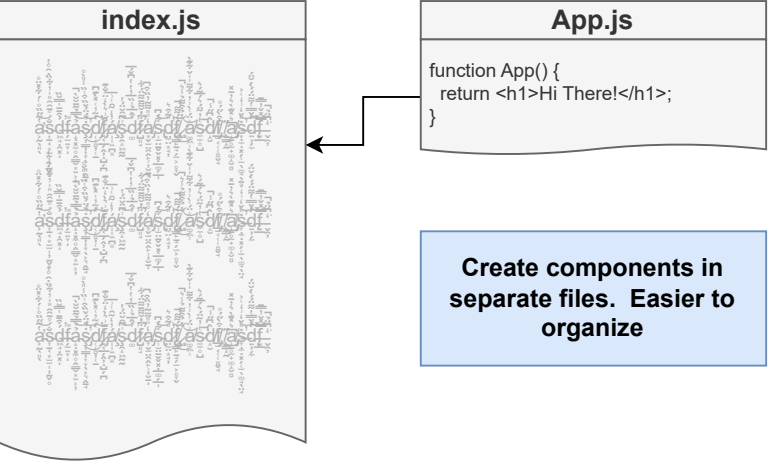
index.js

```
const express = require('express');
const app = express();
const port = 3000;

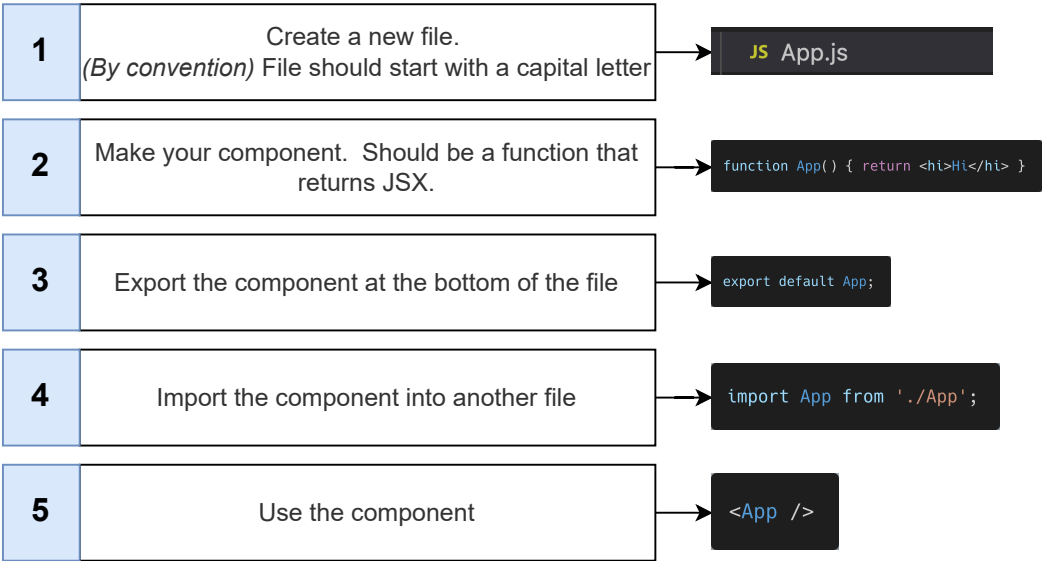
app.get('/', (req, res) => {
  res.send('Hello World!');
});

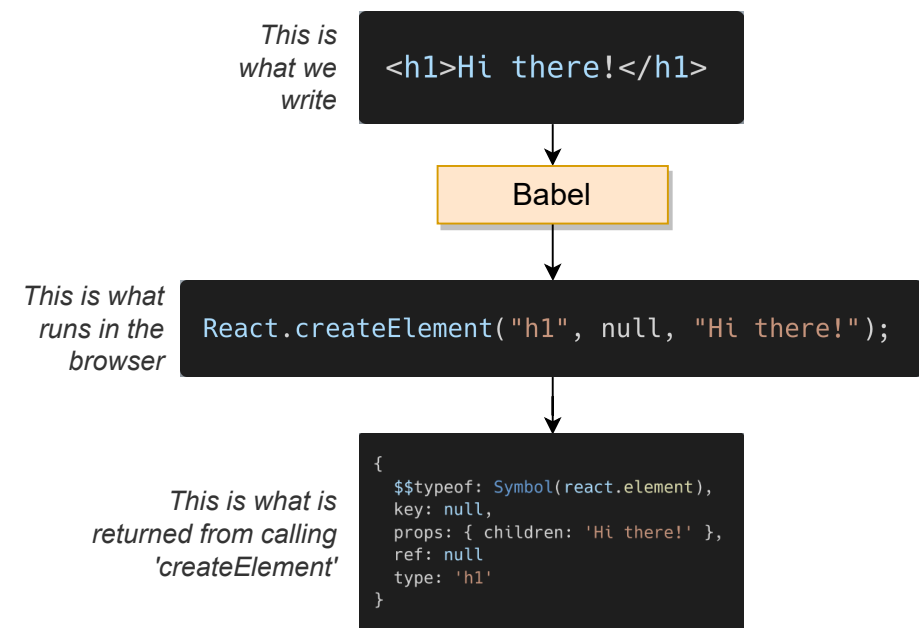
app.listen(port, () => {
  console.log(`Example app listening on port ${port}`);
});
```

The index.js file
can get messy
really quickly



Creating a Component





babeljs.io/repl

*Tool to show you what your JSX
is turned into*

```
<h1>Hi there!</h1>
```

Writing this doesn't make anything show up in the browser automatically

This creates an **instruction** for React, telling it to make an element

We have to **return** it from a component for React to use it

Good!
JSX on same line as return

```
function App() {  
  return <h1>Hi!</h1>  
}
```

Good!
Paren on same line as return

```
function App() {  
  return (  
    <h1>Hi!</h1>  
  );  
}
```

Bad!
JSX on next line

```
function App() {  
  return;  
  <h1>Hi!</h1>;  
}
```

Bad!
Paren on next line

```
function App() {  
  return  
    (<h1>Hi!</h1>);  
}
```

Standard HTML

Some tags can be self closing

`` `<link />`
`<area />` `<hr />`
`<col />` `<meta />`

JSX

Any tag can be self closing

`<div />` ``
`` ``
`<button />` `<label />`

Use self-closing tags unless you want to nest an element

Very long elements should be multi-lined

```
<input id="number" type="number" min={5} max={10} value={7} placeholder={5} />
```

↓

```
<input  
  id="number"  
  type="number"  
  min={5}  
  max={10}  
  value={7}  
  placeholder={5}  
/>
```

Curly braces mean we are about to reference a JS variable or expression

```
function App() {  
  const message = 'Hi there!';  
  
  return <h1>{message}</h1>  
}
```

Expressions are any snippets of code that result in a value.

Good

```
function App() {  
  return <h1>{1 + 1}</h1>  
}
```

Bad

```
function App() {  
  return <h1>{const message = 'hi there'}</h1>  
}
```

Good

```
function App() {  
  return <h1>{'a,b,c'.split(',')}</h1>  
}
```

Bad

```
function App() {  
  const colors = {};  
  
  return <h1>{colors.favorite = 'red'}</h1>  
}
```

Some values might not display quite as you expect

```
function App() {  
  const value = true;  
  
  return <h1>{value}</h1>  
}
```

*What will be
in the h1?*

Basics of JSX

Returning JSX
Handling JSX tags
Referencing JS values in JSX
Element Props
Inline Styles

```
function App() {  
  return <input type="number" min={5} max={10} />  
}
```

"Props"

*Customizes
an element*

