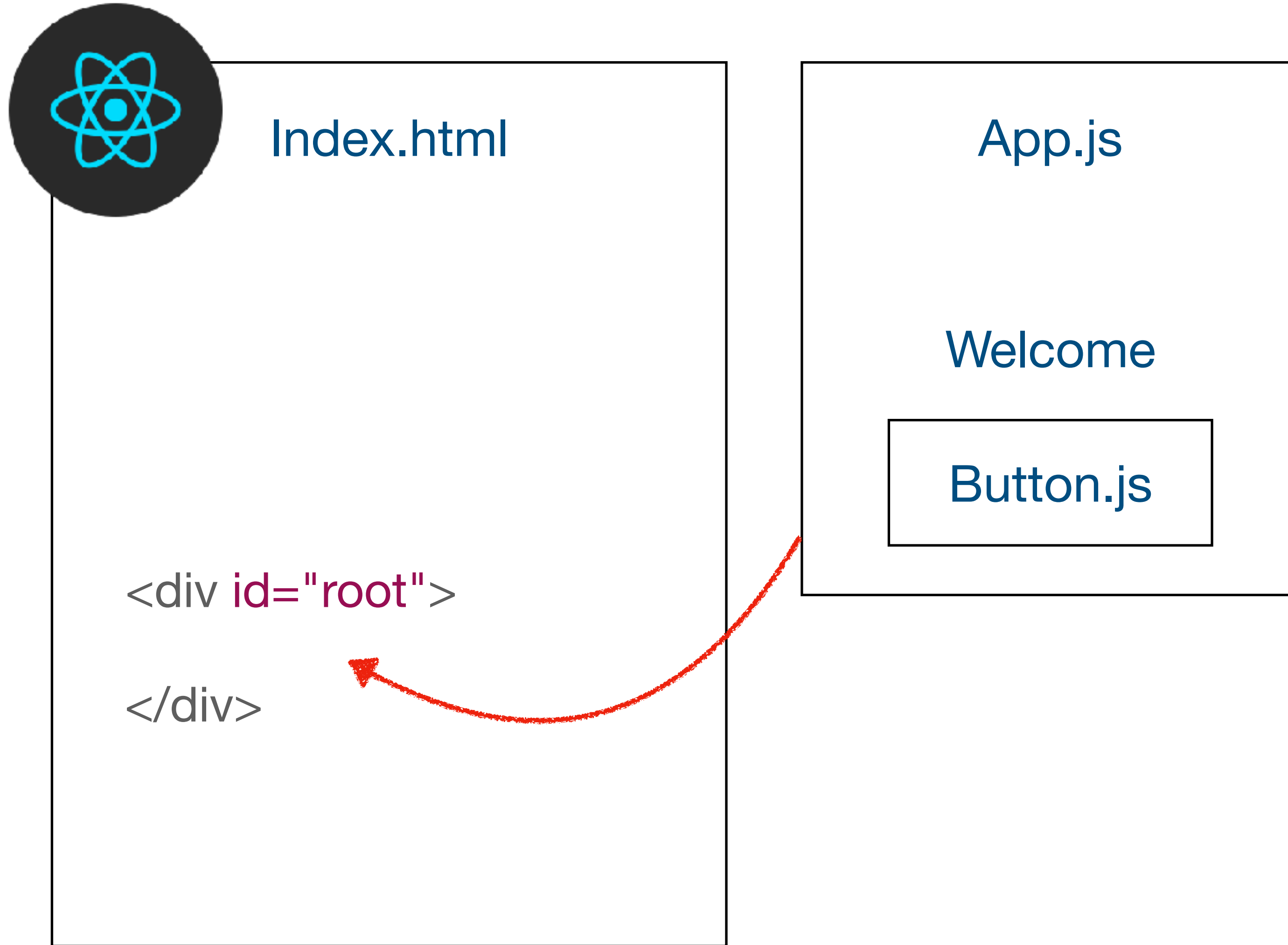


Single-page Application

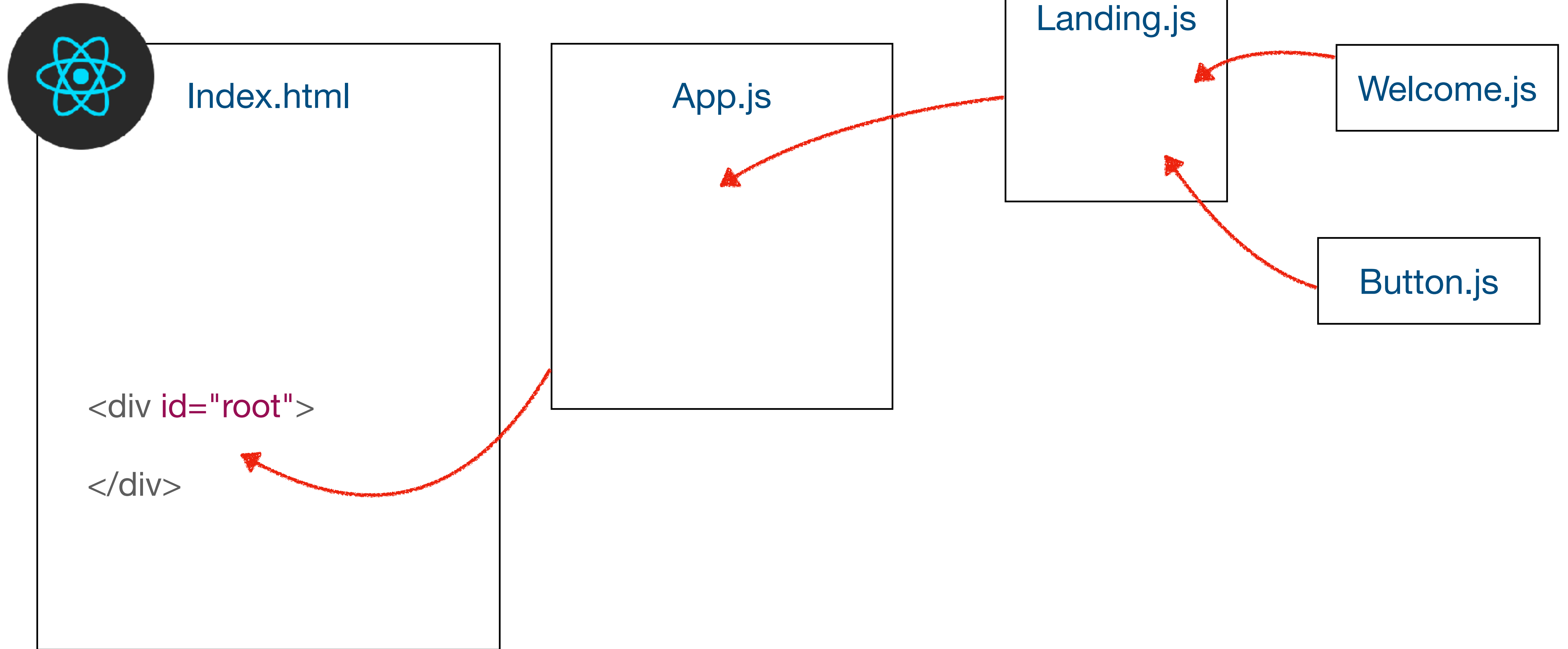
Welcome

Button.js

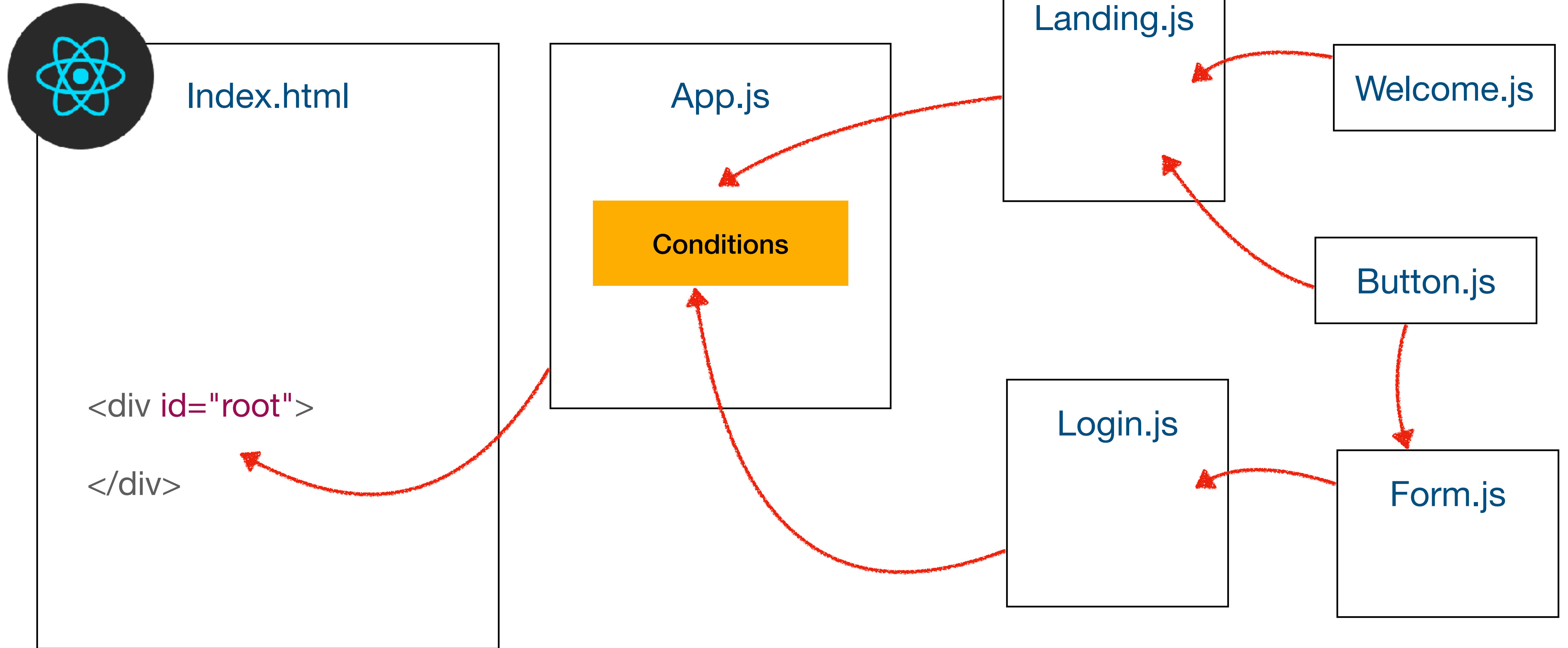
Single-page Application



Single-page Application



Single-page Application



App Cars



Index.html

```
<div id="root">
```

```
</div>
```

App.js

Mon Catalogue Voiture

Marque: Ford

Couleur: red

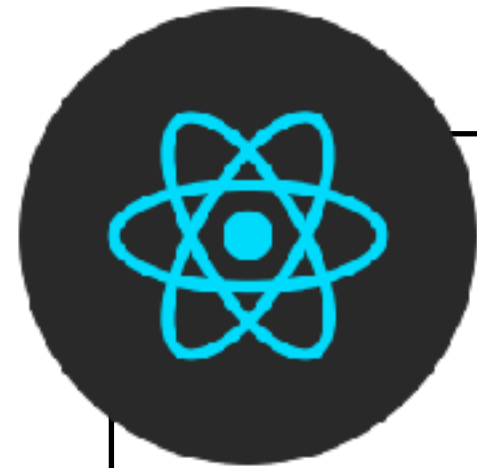
Marque: Mercedes

Couleur: black

Marque: Peugeot

Couleur: green

App Cars



Index.html

```
<div id="root">
```

```
</div>
```

App.js

```
state = {  
  titre: 'Mon Catalogue Voitures'  
}
```

MyCars.js

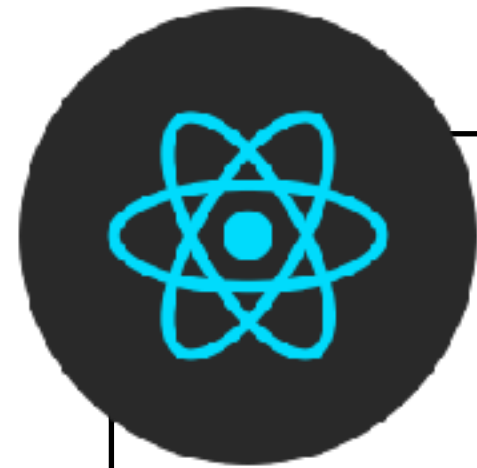
Mon Catalogue Voiture

Cars.js

Marque: Ford

Couleur: red

Exercice sur les props



Index.html

```
<div id="root">
```

```
</div>
```

App.js

```
state = {  
  titre: 'Mon Catalogue Voitures'  
  color: 'green'  
}
```

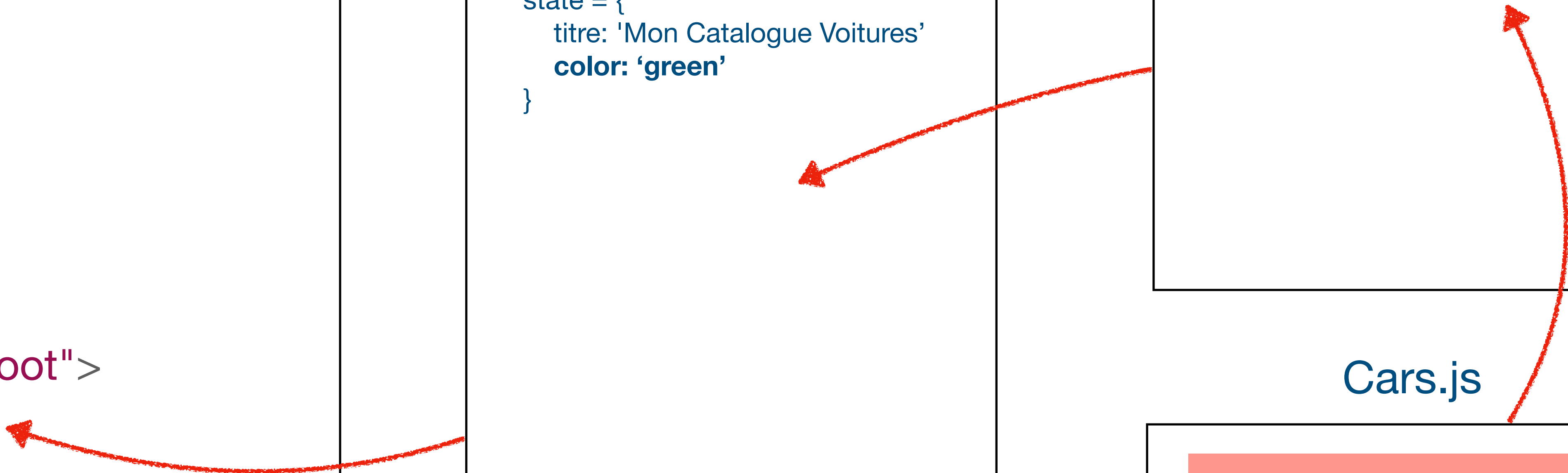
MyCars.js

Mon Catalogue Voiture

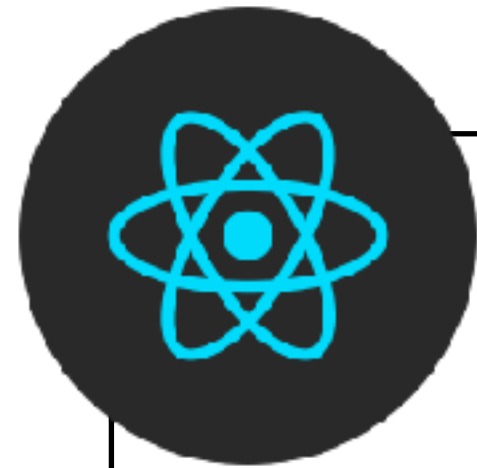
Cars.js

Marque: Ford

Couleur: red



Exercice sur les states



Index.html

```
<div id="root">
```

```
</div>
```

App.js

```
state = {  
  titre: 'Mon Catalogue Voitures'  
  color: 'green'  
}
```

MyCars.js

Mon Catalogue Voiture

Créer un state avec une propriété **cars** dans laquelle on aura les trois noms de voitures dans un array et utiliser ce state pour appliquer les noms sur les trois composants Car ["Ford", "Mercedes", "Peugeot"]

Cars.js

Marque: Ford

Couleur: red

Exporter et importer un composant

"Named Export" vs "Default Export" en ES6

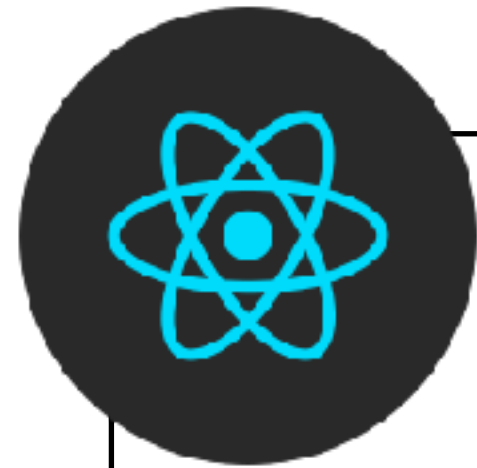
Option 1: Export Default

Dans la vidéo précédente, nous avons créé des composants que nous avons ensuite exporté et importé dans d'autres composants. Pour ce faire, nous avons opté pour le "Export Default" en ES6. Cela permet d'exporter un élément et de l'importer dans un autre composant avec le nom de notre choix et pas obligatoirement avec le vrai nom de la classe ou de la fonction.

Option 2: Named Export

En plus du "Export default", vous pouvez exporter et importer vos composants via les "Named Exports" en ES6. Pour ce faire, vous exportez le composant via un simple "export" et vous l'importez dans des accolades avec le vrai nom du composant.

Exercice sur les states



Index.html

```
<div id="root">
```

```
</div>
```

App.js

```
state = {  
  titre: 'Mon Catalogue Voitures'  
  color: 'green'  
}
```

MyCars.js

Mon Catalogue Voiture

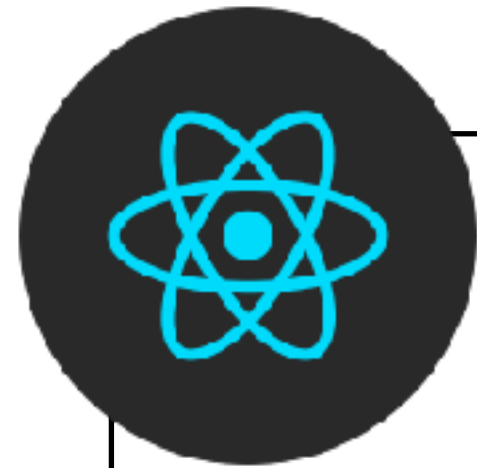
Créer un state avec une propriété **cars** dans laquelle on aura les trois noms de voitures dans un array et utiliser ce state pour appliquer les noms sur les trois composants Car **["Ford", "Mercedes", "Peugeot"]**

Cars.js

Marque: Ford

Couleur: red

Exercice sur les states



Index.html

```
<div id="root">
```

```
</div>
```

App.js

```
state = {  
  titre: 'Mon Catalogue Voitures'  
  color: 'green'  
}
```

Wrapper.js



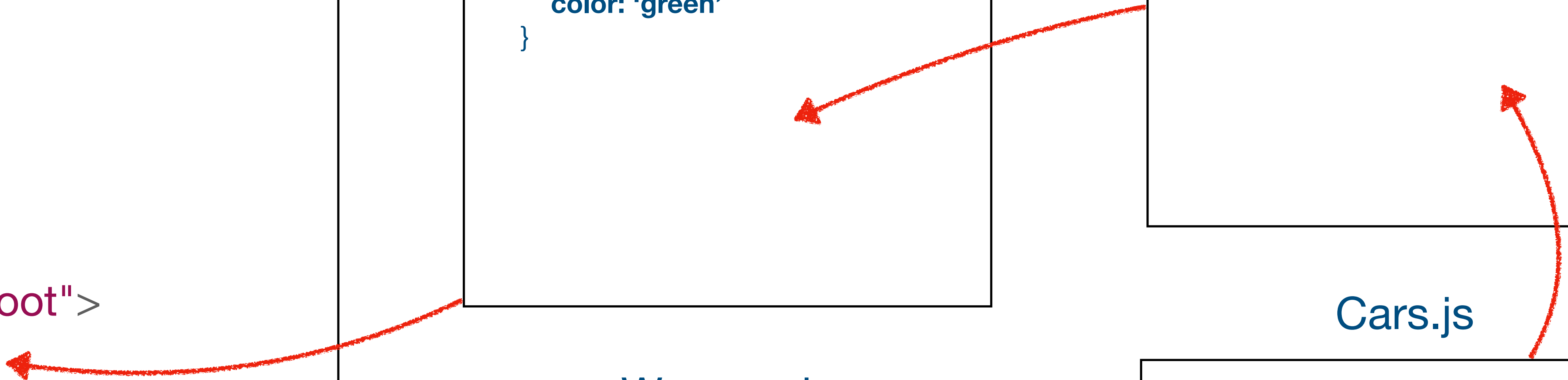
MyCars.js

Mon Catalogue Voiture

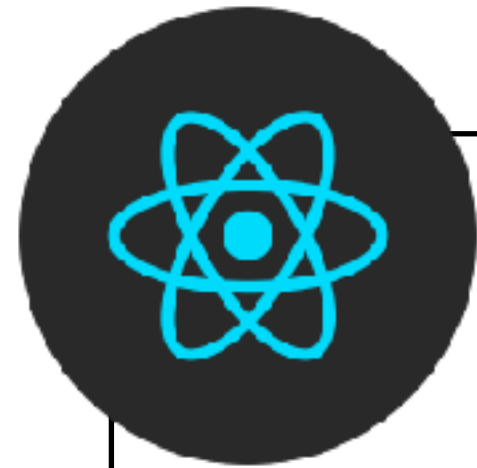
Cars.js

Marque: Ford

Couleur: red



Exercice sur les props children



Index.html

```
<div id="root">
```

```
</div>
```

App.js

```
state = {  
  titre: 'Mon Catalogue Voitures'  
  color: 'green'  
}
```

Wrapper.js

MyCars.js

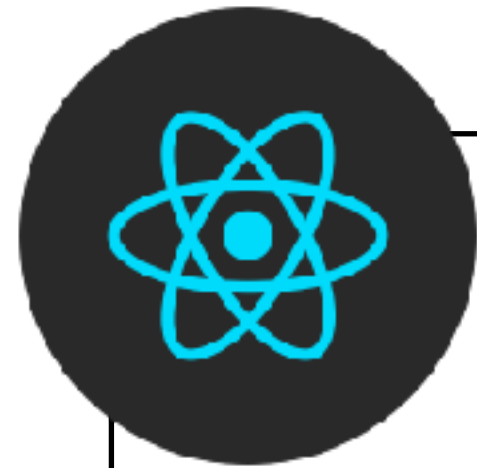
Mon Catalogue Voiture

Cars.js

Marque: Ford

Couleur: red

Exercice sur les props children



Index.html

```
<div id="root">
```

```
</div>
```

App.js

```
state = {  
  titre: 'Mon Catalogue Voitures'  
  color: 'green'  
}
```

Wrapper.js

MyHeader.js

<h1>

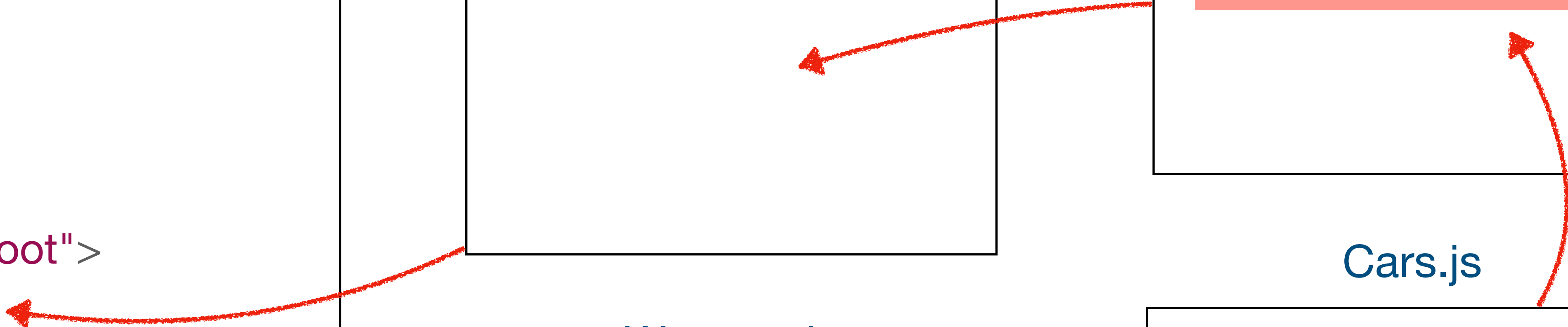
MyCars.js

Mon Catalogue Voiture

Cars.js

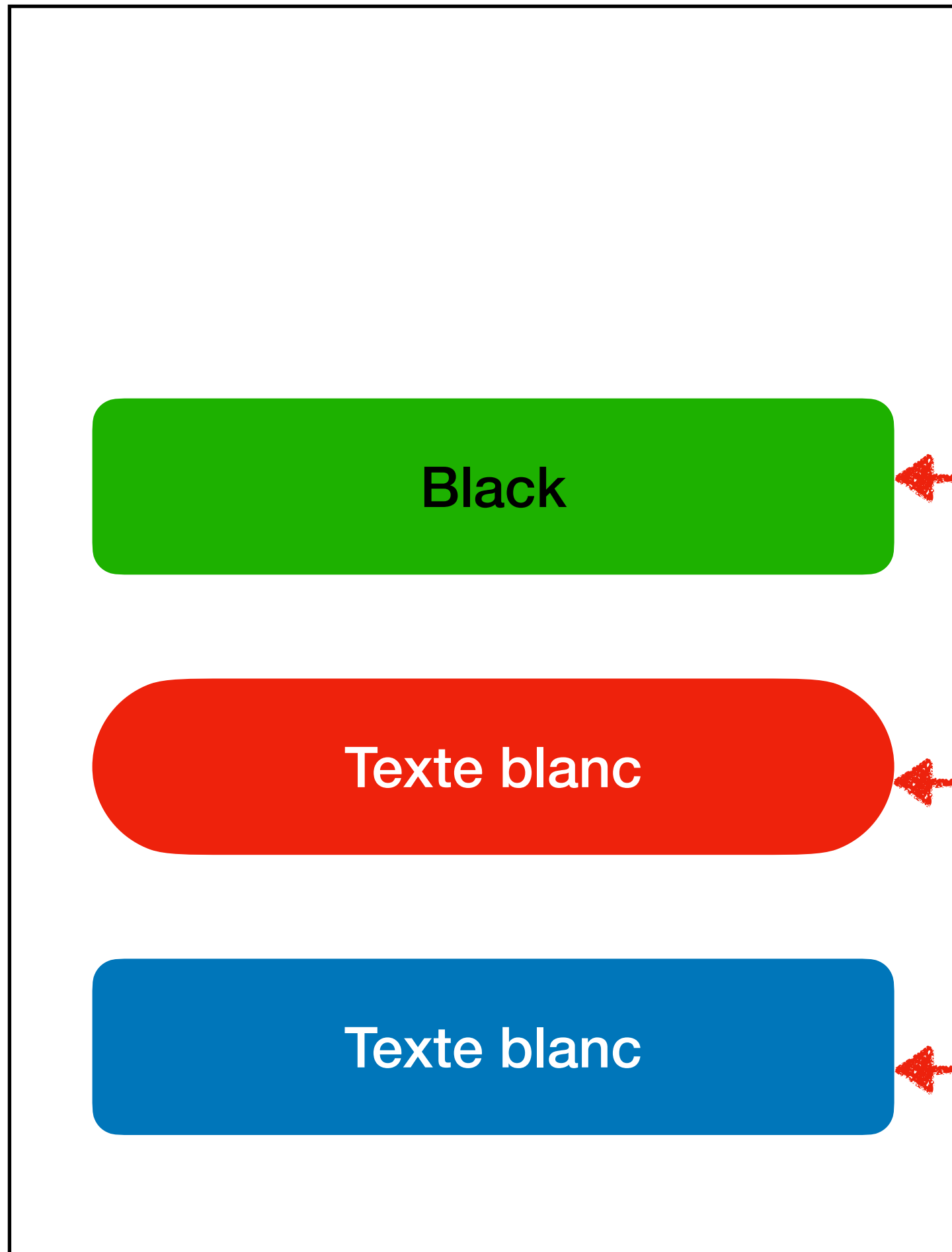
Marque: Ford

Couleur: red



Exercice CSS - Custom button

Game.js



CustomBtn.js

```
const customBtn = {  
  backgroundColor: "grey",  
  border: 'none',  
  color: '#fff',  
  fontSize: '19px',  
  padding: '15px 30px',  
  textAlign: 'center',  
  textDecoration: 'none',  
  borderRadius: '7px',  
  display: 'block',  
  margin: '5px auto'  
}
```

