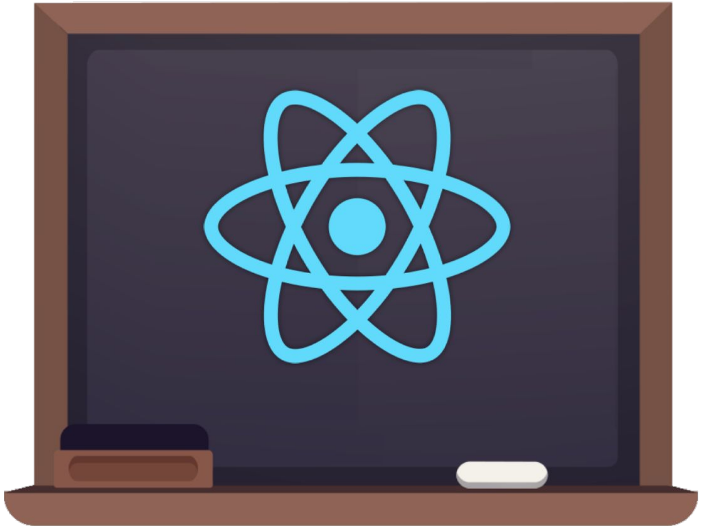


3. Render

Introduction to React

Index

- 🦠 Components
- 🏗 JSX
- 🎀 Styling
- 🕹 Dynamic render

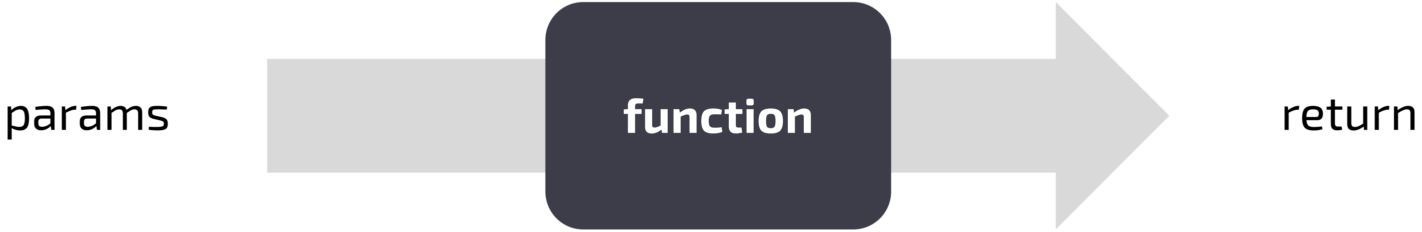
Index

- 🦠 Components
- 🏗 JSX
- 🎀 Styling
- 🕹 Dynamic render

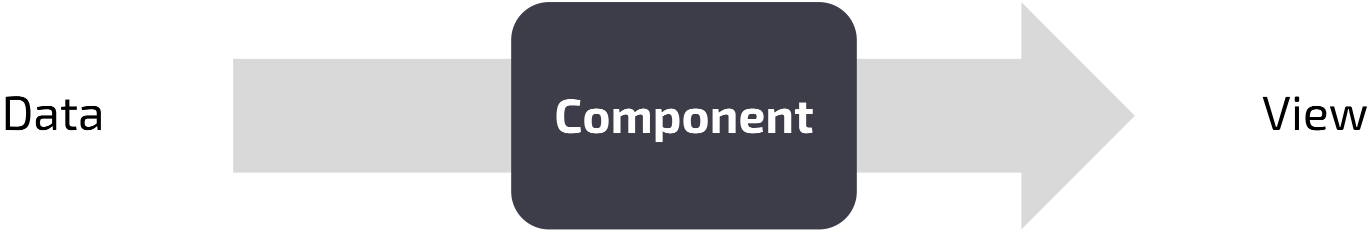
Index

- 🦠 **Components**
- 🏗️ JSX
- 🎀 Styling
- 🕹️ Dynamic render

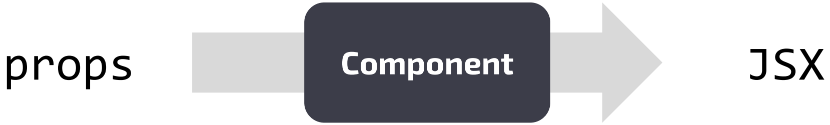
🦠 Components



Components



Components


props

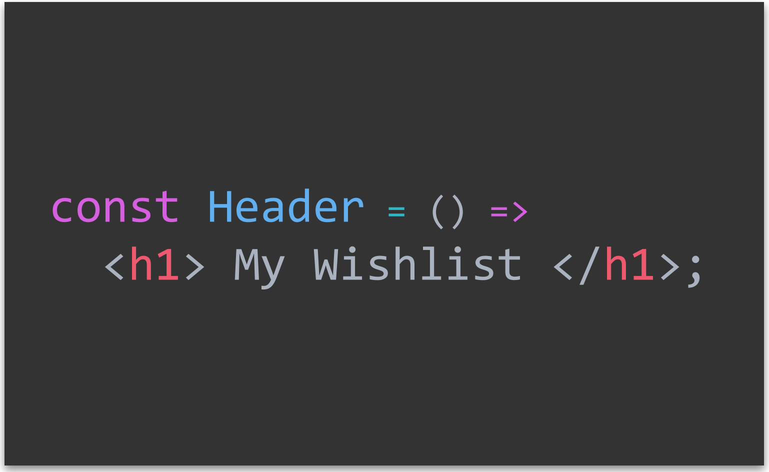
Component

JSX

```
const Header = () => <div> My Wishlist </div>;
```

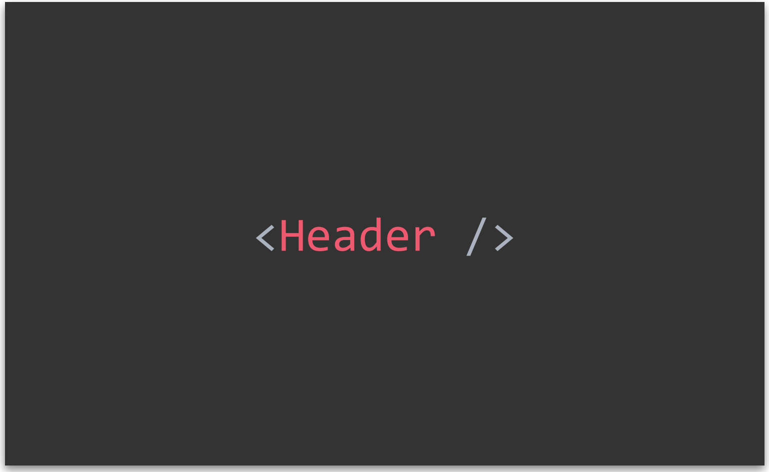
Components

Declaration



```
const Header = () =>
  <h1> My Wishlist </h1>;
```

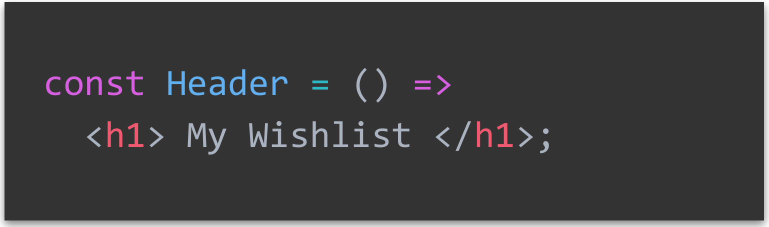
Usage



```
<Header />
```

Components

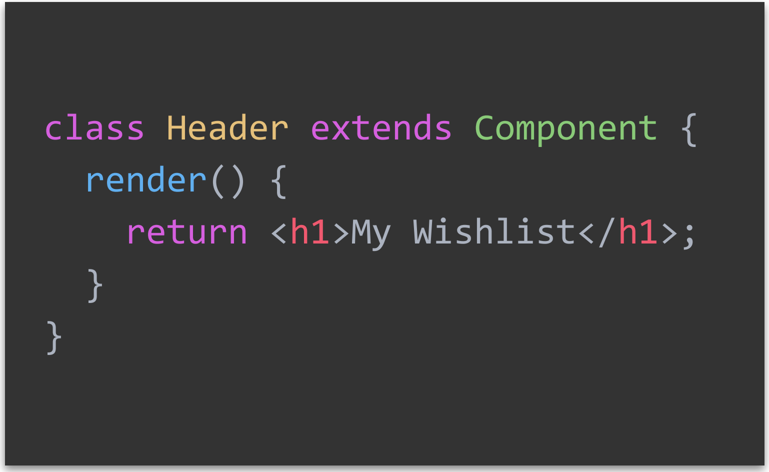
Functional component



```
const Header = () =>
  <h1> My Wishlist </h1>;
```

~

Class component



```
class Header extends Component {
  render() {
    return <h1>My Wishlist</h1>;
  }
}
```

Index

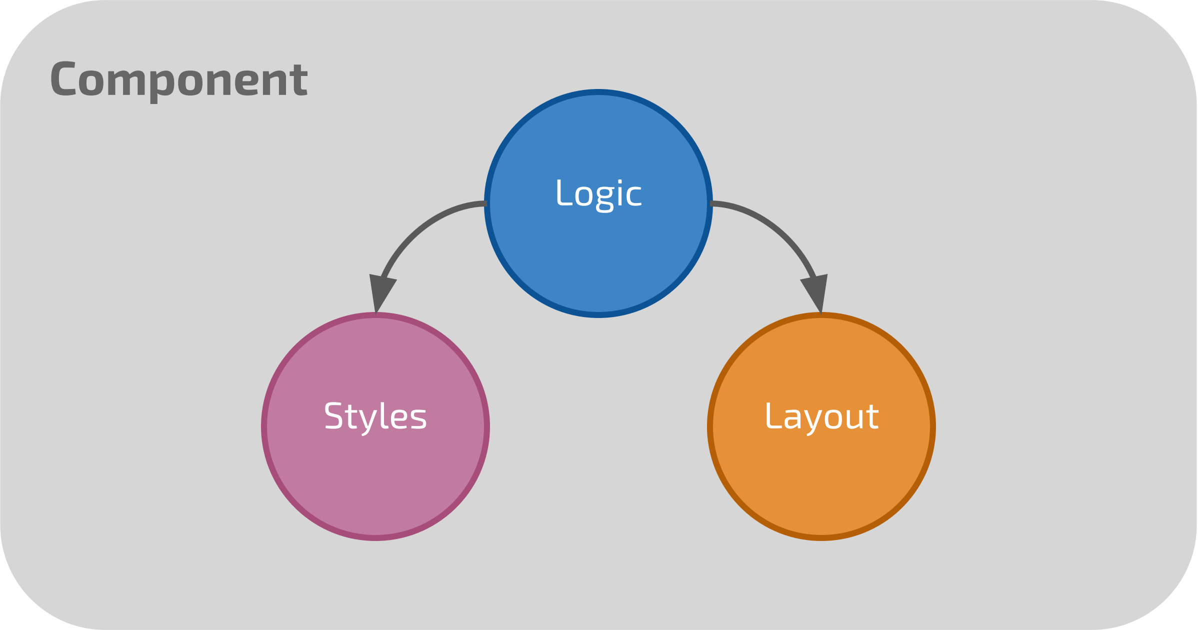
✅ Components

🏗 **JSX**

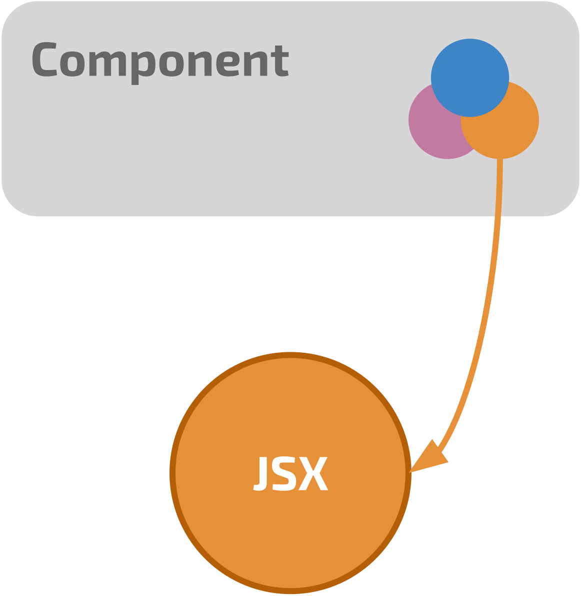
🎀 Styling

🕹 Dynamic render

🏗 JSX

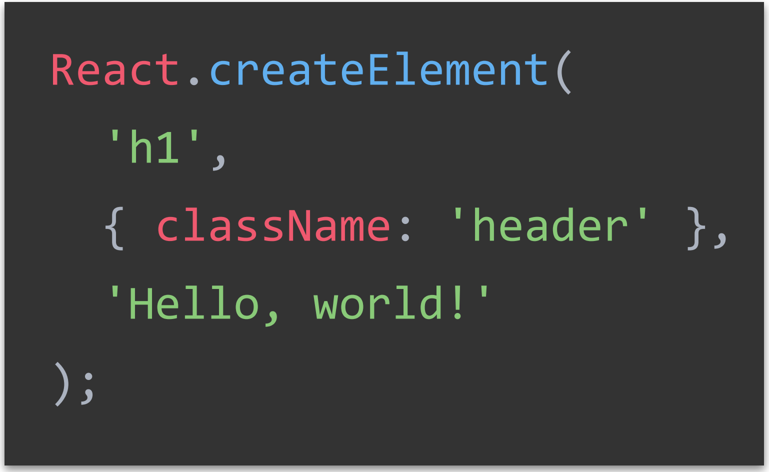


🏗 JSX



JSX – Compared to pure JS

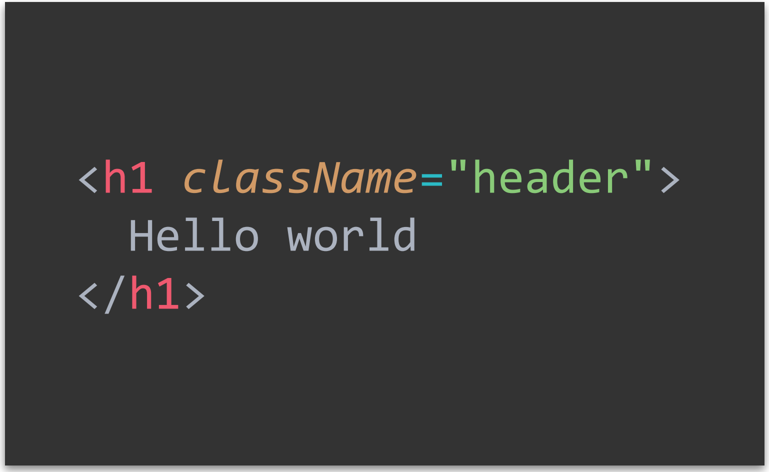
Pure JS



```
React.createElement(
  'h1',
  { className: 'header' },
  'Hello, world!'
);
```

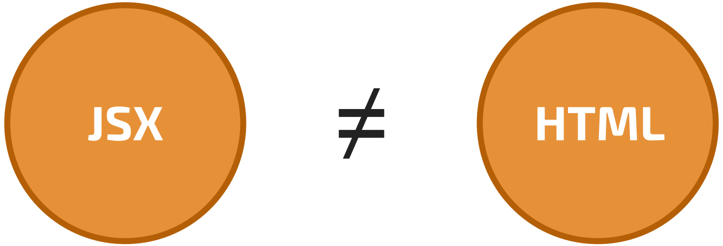
~

JSX



```
<h1 className="header">
  Hello world
</h1>
```

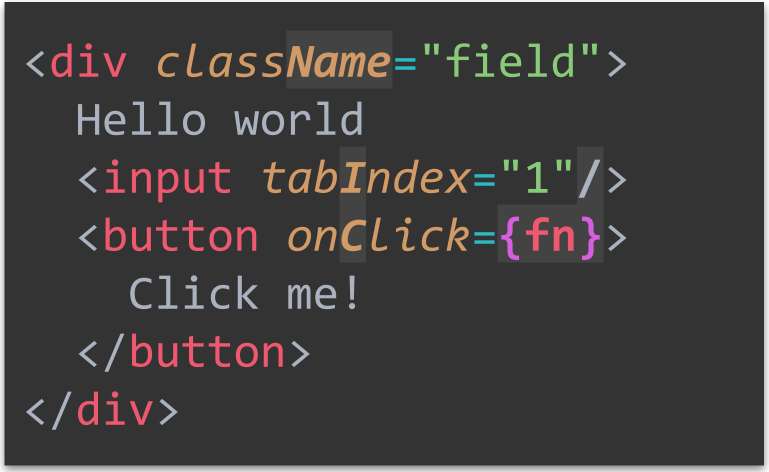
JSX – Compared to HTML



JSX ≠ HTML

JSX – Compared to HTML

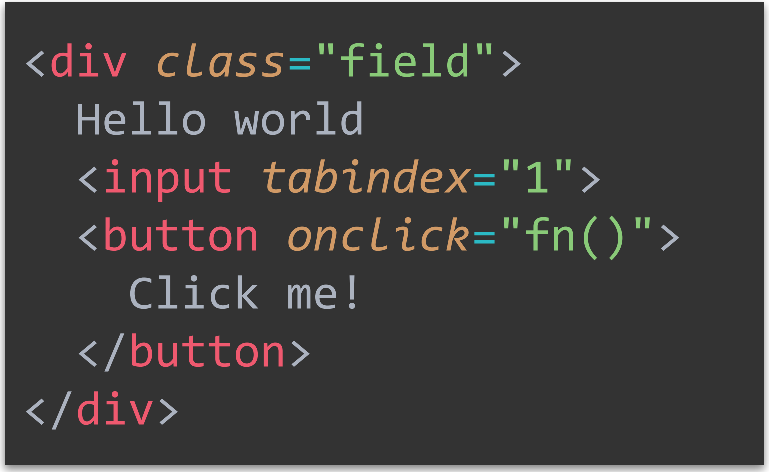
JSX



```
<div className="field">
  Hello world
  <input tabIndex="1"/>
  <button onClick={fn}>
    Click me!
  </button>
</div>
```

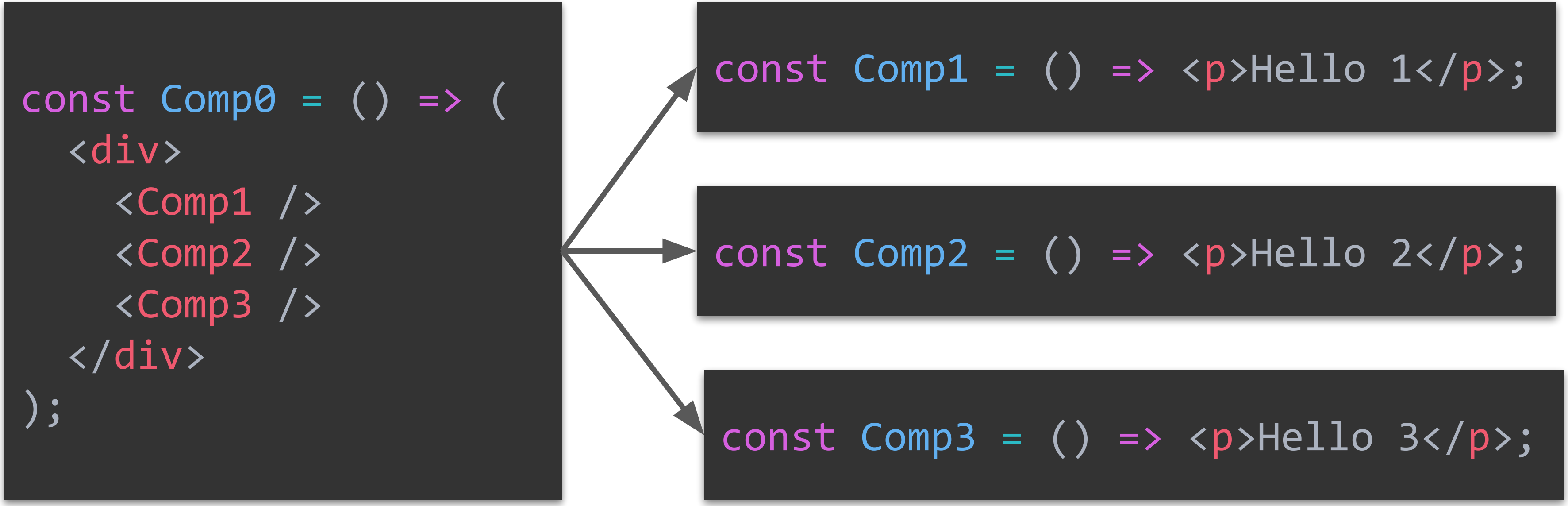
≠

HTML



```
<div class="field">
  Hello world
  <input tabindex="1">
  <button onclick="fn()">
    Click me!
  </button>
</div>
```

JSX - Component relations



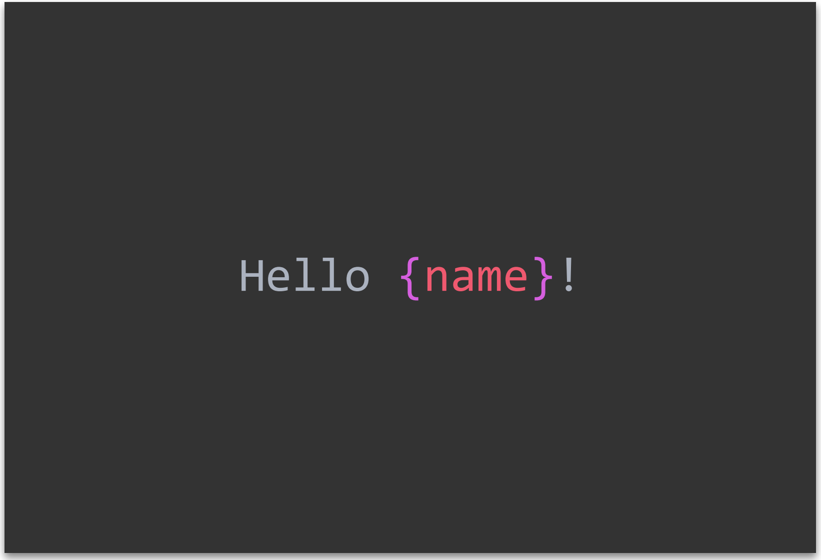
```
const Comp0 = () => (
  <div>
    <Comp1 />
    <Comp2 />
    <Comp3 />
  </div>
);
```

```
const Comp1 = () => <p>Hello 1</p>;
```

```
const Comp2 = () => <p>Hello 2</p>;
```

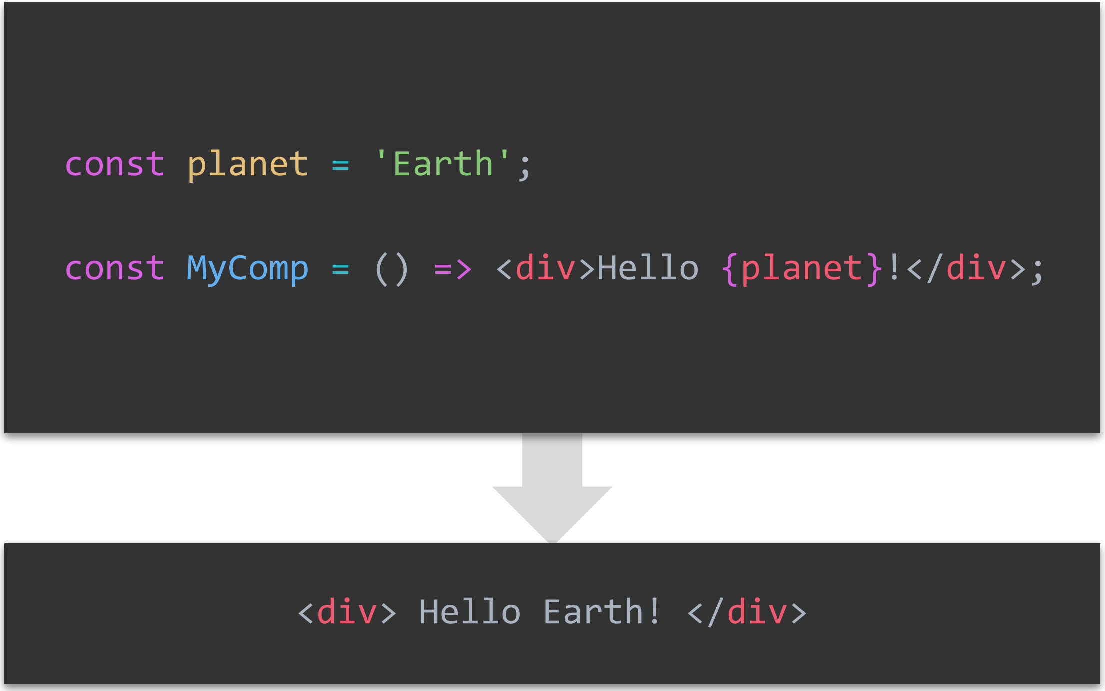
```
const Comp3 = () => <p>Hello 3</p>;
```

JSX - Expressions



```
Hello {name}!
```

JSX – Expressions. Variables.

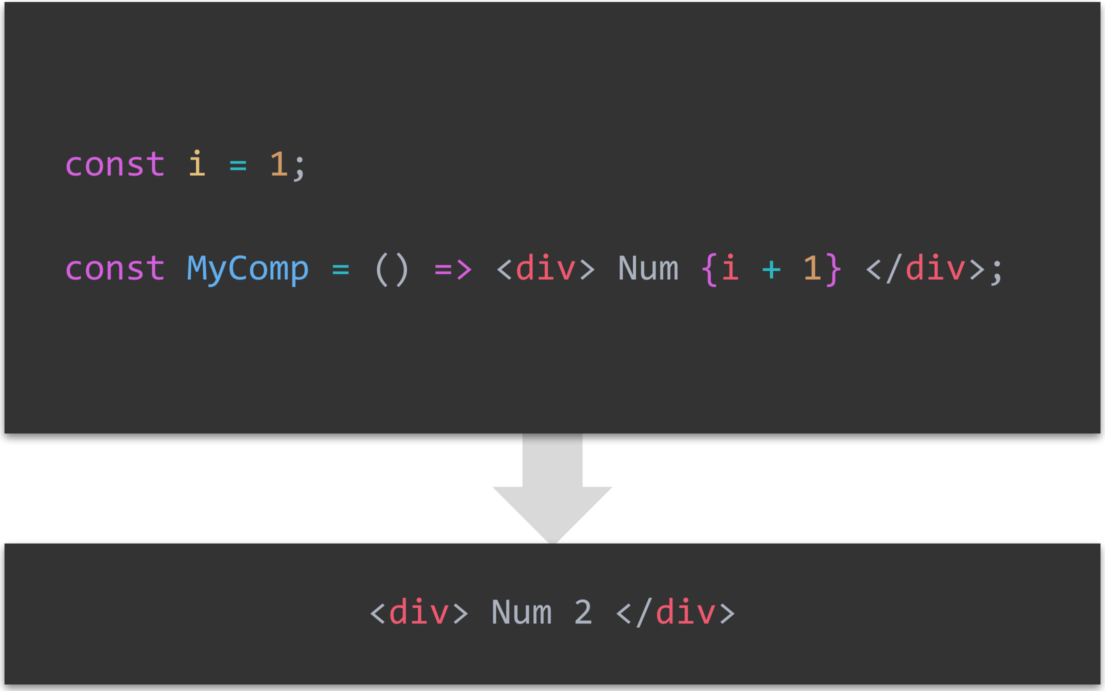


```
const planet = 'Earth';

const MyComp = () => <div>Hello {planet}!</div>;
```

```
<div> Hello Earth! </div>
```

JSX – Expressions. Operations.

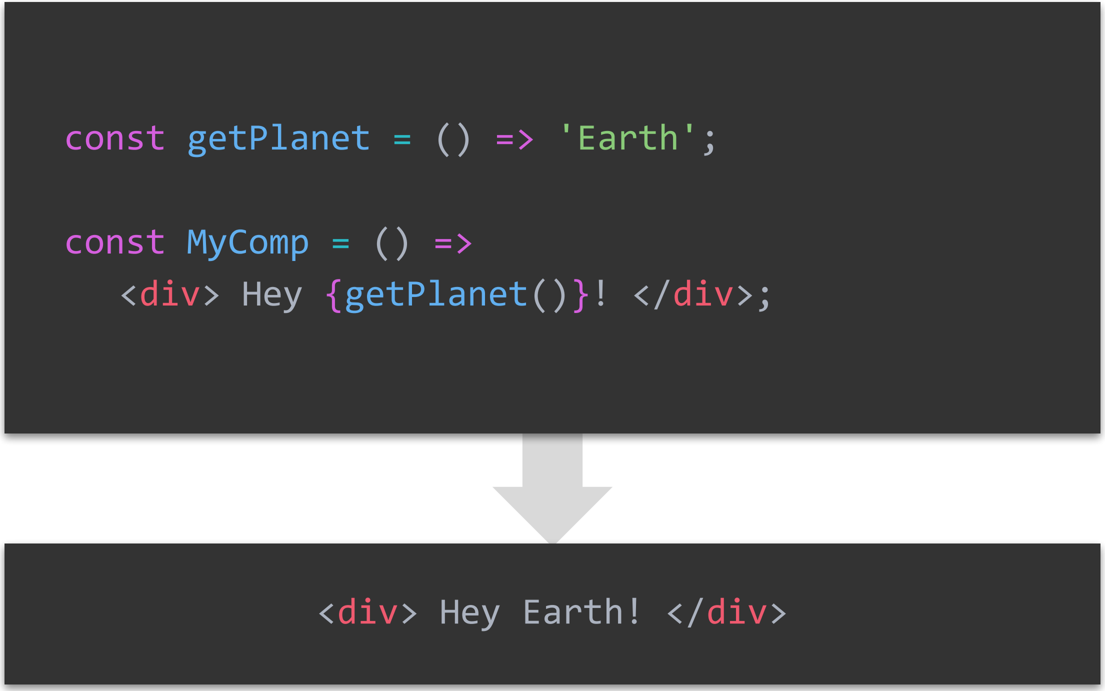


```
const i = 1;

const MyComp = () => <div> Num {i + 1} </div>;
```

```
<div> Num 2 </div>
```

JSX – Expressions. Functions.

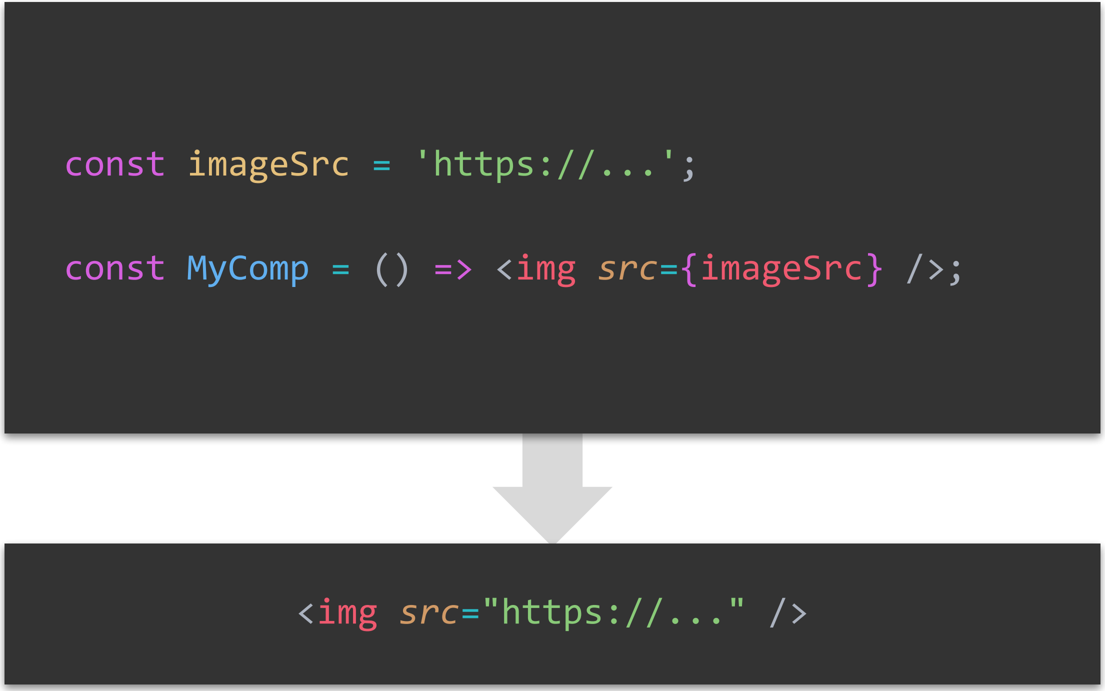


```
const getPlanet = () => 'Earth';

const MyComp = () =>
   <div> Hey {getPlanet()}! </div>;
```

```
<div> Hey Earth! </div>
```

JSX – Expressions. Attributes.

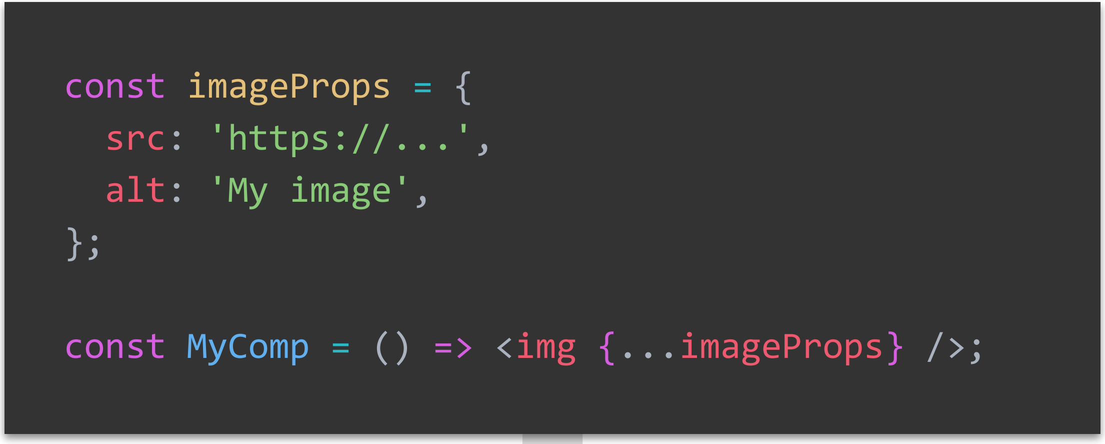


```jsx
const imageSrc = 'https://...';

const MyComp = () => <img src={imageSrc} />;
```

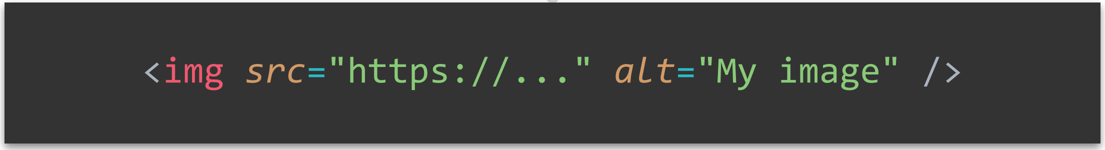
```html
<img src="https://..." />
```

JSX - Expressions. Spread object.



```
const imageProps = {
  src: 'https://...',
  alt: 'My image',
};

const MyComp = () => <img {...imageProps} />;
```



```
<img src="https://..." alt="My image" />
```

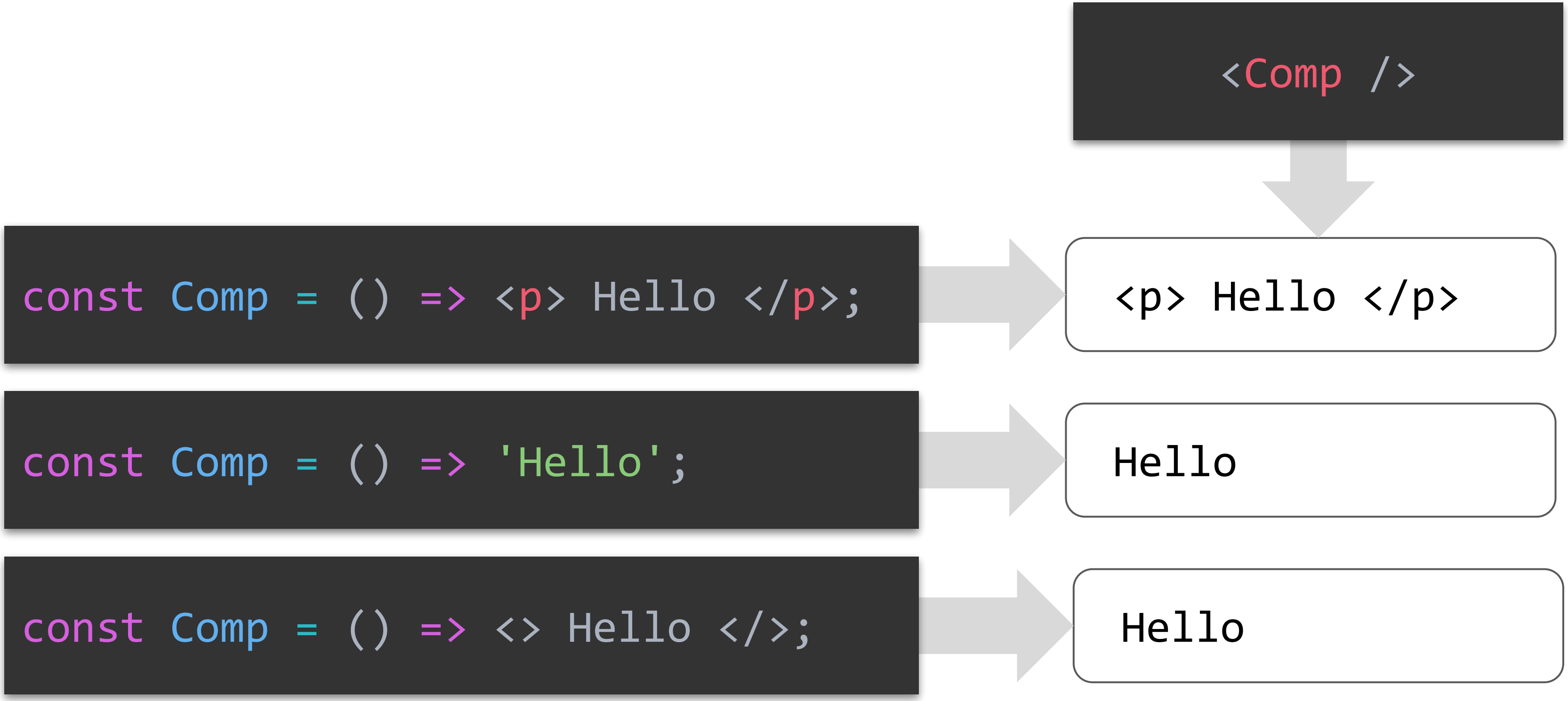
JSX – Fragments

```
<React.Fragment>
  Hello world
</React.Fragment>
```

~

```
<>
  Hello world
</>
```

JSX - Fragments



```
<Comp />
```

```
const Comp = () => <p> Hello </p>;
```

```
<p> Hello </p>
```

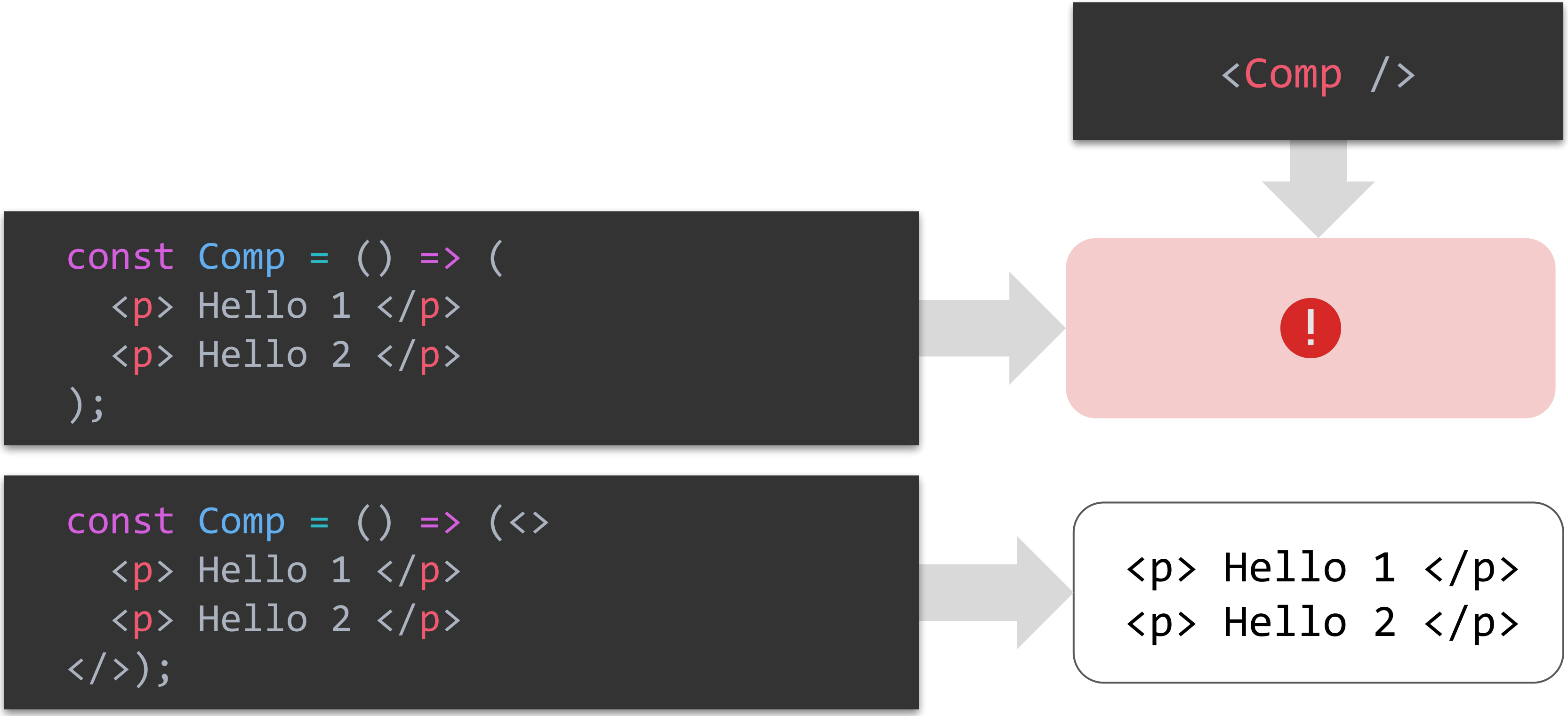
```
const Comp = () => 'Hello';
```

```
Hello
```

```
const Comp = () => <> Hello </>;
```

```
Hello
```

JSX – Fragments



```
<Comp />
```

```
const Comp = () => (
  <p> Hello 1 </p>
  <p> Hello 2 </p>
);
```

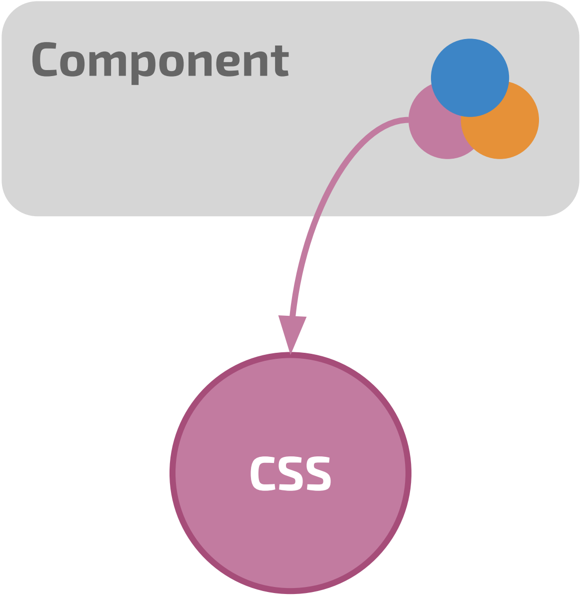
```
const Comp = () => (<>
  <p> Hello 1 </p>
  <p> Hello 2 </p>
</>);
```

```
<p> Hello 1 </p>
<p> Hello 2 </p>
```

Index

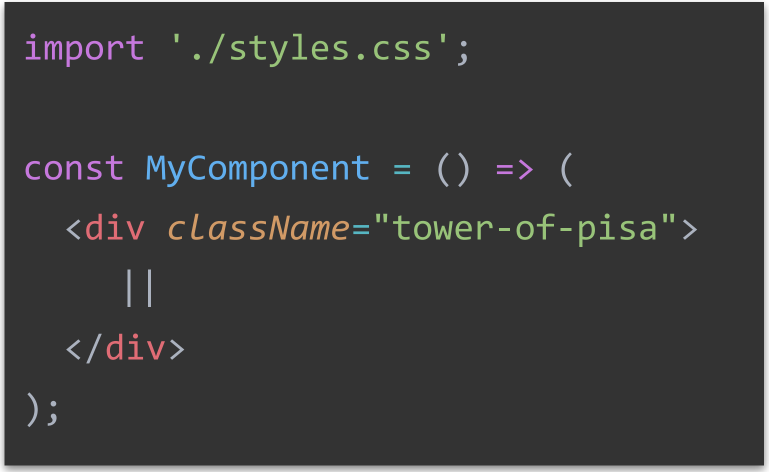
- ✅ Components
- ✅ JSX
- 🎀 **Styling**
- 🕹️ Dynamic render

🎀 Styling



🎀 Styling

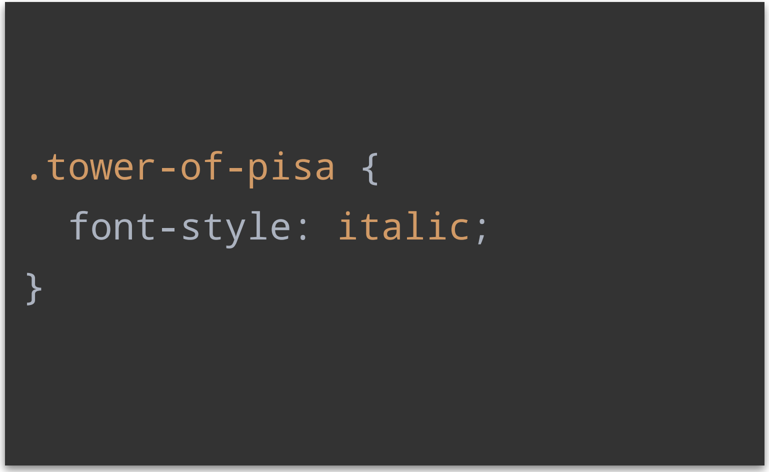
Component



```
import './styles.css';

const MyComponent = () => (
  <div className="tower-of-pisa">
    ||
  </div>
);
```

styles.css



```
.tower-of-pisa {
  font-style: italic;
}
```

🎀 Styling

Component

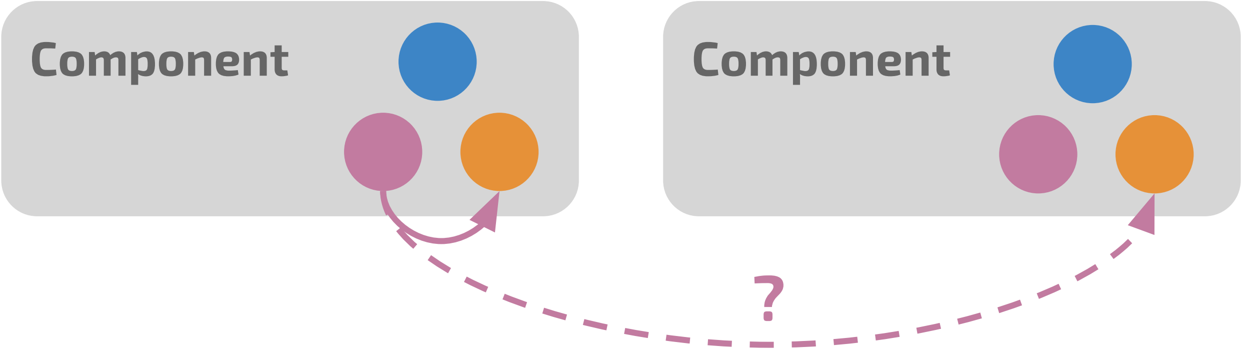
```
import './styles.css';
const MyComponent = () => (
  <div>
    <div className="ikea"></div>
    <div className="lego"></div>
  </div>
);
```

styles.css

```
.ikea {
  display: table;
}
.lego {
  display: block;
}
```

🎀 Styling - Encapsulation

¿Real encapsulation?



🎀 Styling - BEM

BEM

Block

🎀 Styling - BEM

B**E**M

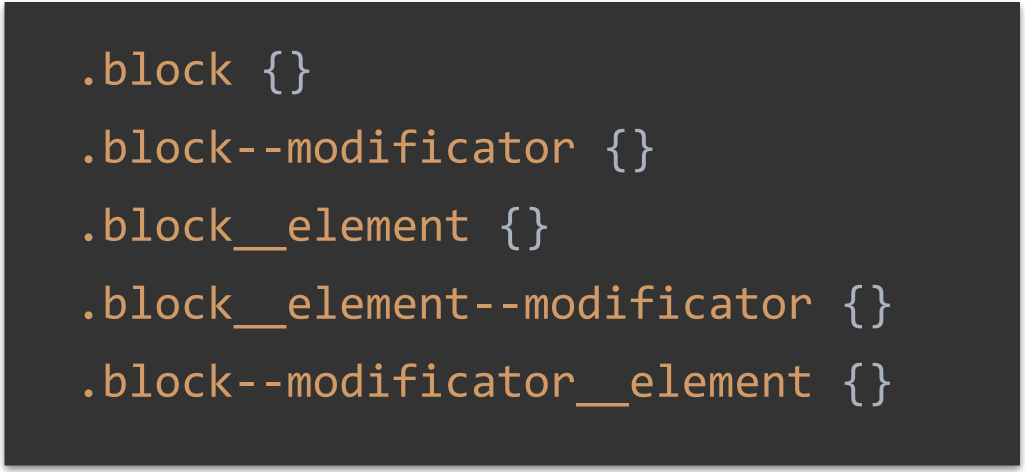
Element

🎀 Styling - BEM

BE**M**

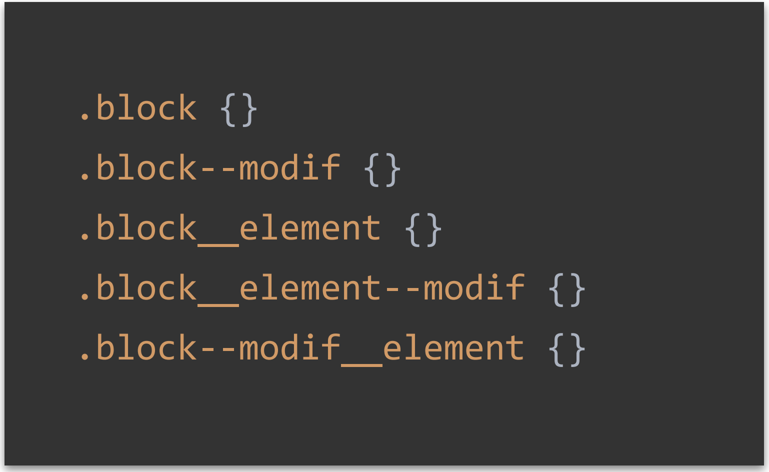
Modificator

Styling - BEM

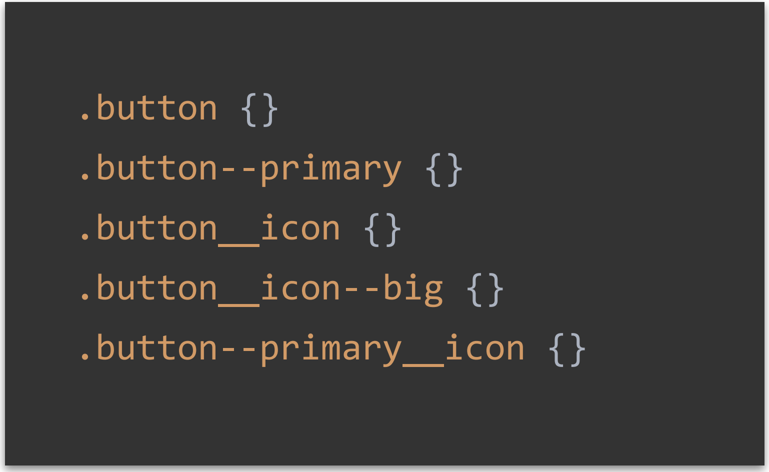


```
.block {}
.block--modificator {}
.block__element {}
.block__element--modificator {}
.block--modificator__element {}
```

🎀 Styling - BEM



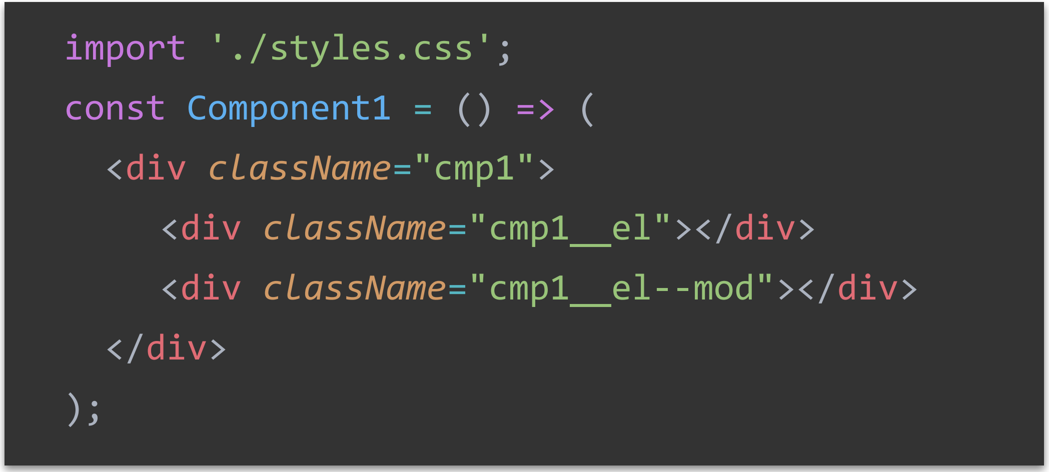
```
.block {}
.block--modif {}
.block__element {}
.block__element--modif {}
.block--modif__element {}
```



```
.button {}
.button--primary {}
.button__icon {}
.button__icon--big {}
.button--primary__icon {}
```

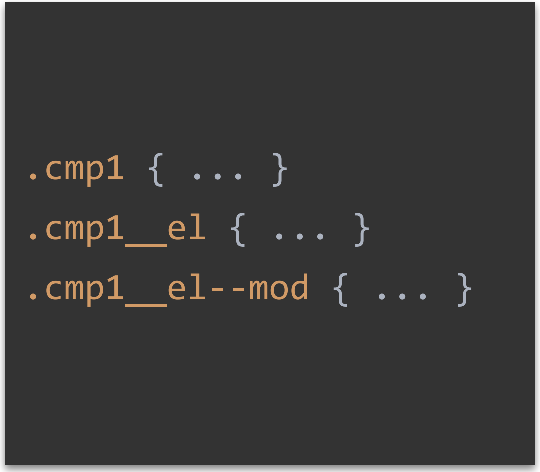
Styling - BEM

Component



```jsx
import './styles.css';
const Component1 = () => (
  <div className="cmp1">
    <div className="cmp1__el"></div>
    <div className="cmp1__el--mod"></div>
  </div>
);
```

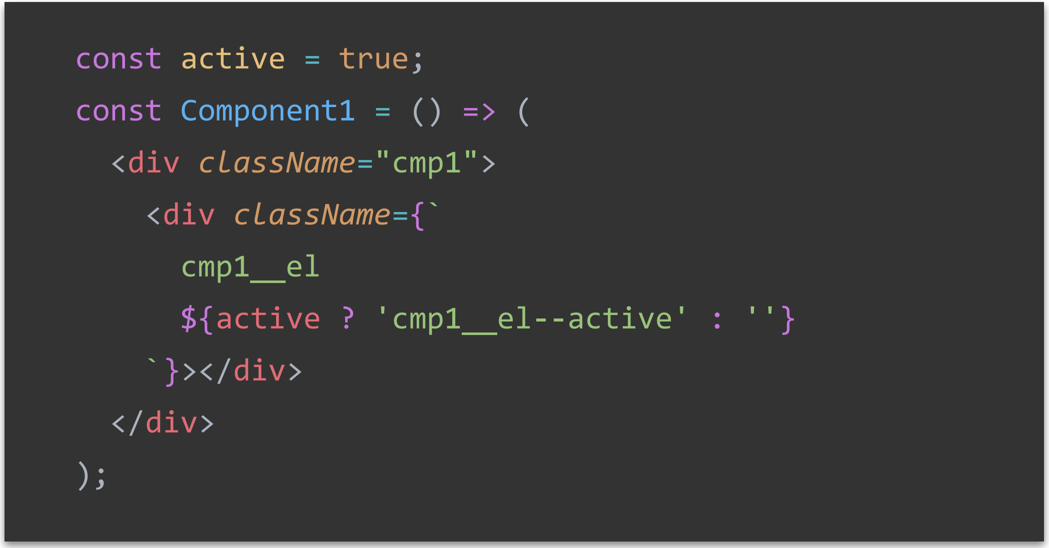
styles.css



```css
.cmp1 { ... }
.cmp1__el { ... }
.cmp1__el--mod { ... }
```

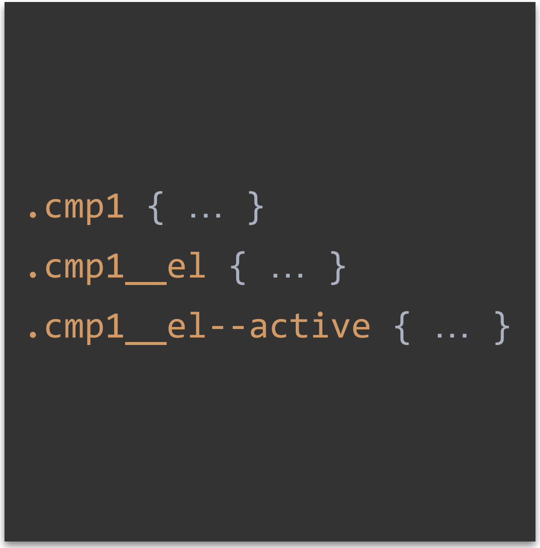
Styling - BEM

Component



```
const active = true;
const Component1 = () => (
  <div className="cmp1">
    <div className={`
      cmp1__el
      ${active ? 'cmp1__el--active' : ''}
    `}></div>
  </div>
);
```

styles.css



```
.cmp1 { … }
.cmp1__el { … }
.cmp1__el--active { … }
```

Styling - BEM

Component

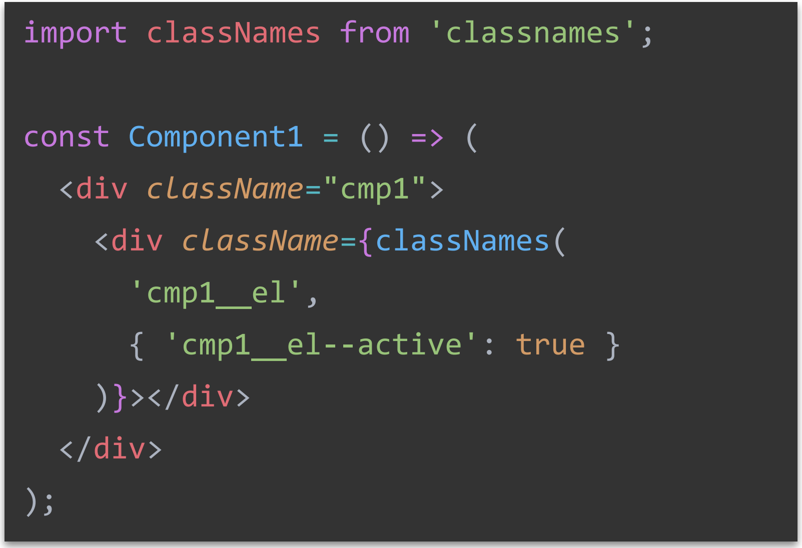
```
const active = true;
const Component1 = () => (
  <div className="cmp1">
    <div className={`
      cmp1__el
      ${active ? 'cmp1__el--active' : ''}
    `}></div>
  </div>
);
```

HTML rendered

```
<div class="cmp1">
  <div class="cmp1__el cmp1__el--active">
  </div>
</div>
```

Styling - BEM

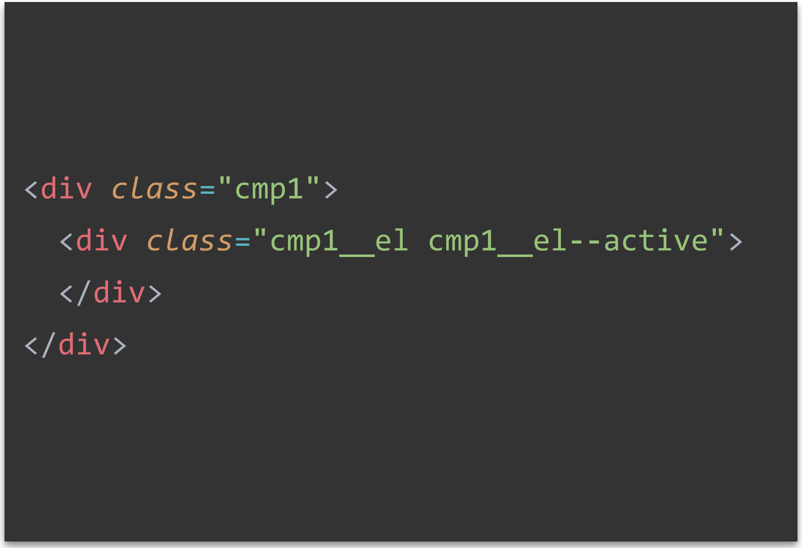
Component



```
import classNames from 'classnames';

const Component1 = () => (
  <div className="cmp1">
    <div className={classNames(
      'cmp1__el',
      { 'cmp1__el--active': true }
    )}></div>
  </div>
);
```

HTML rendered



```
<div class="cmp1">
  <div class="cmp1__el cmp1__el--active">
  </div>
</div>
```

Index

- ✅ Components
- ✅ JSX
- ✅ Styling
- 🕹️ **Dynamic render**

🕹️ Dynamic render

🚥 Conditionals

🛣️ Switch

🎡 Loops

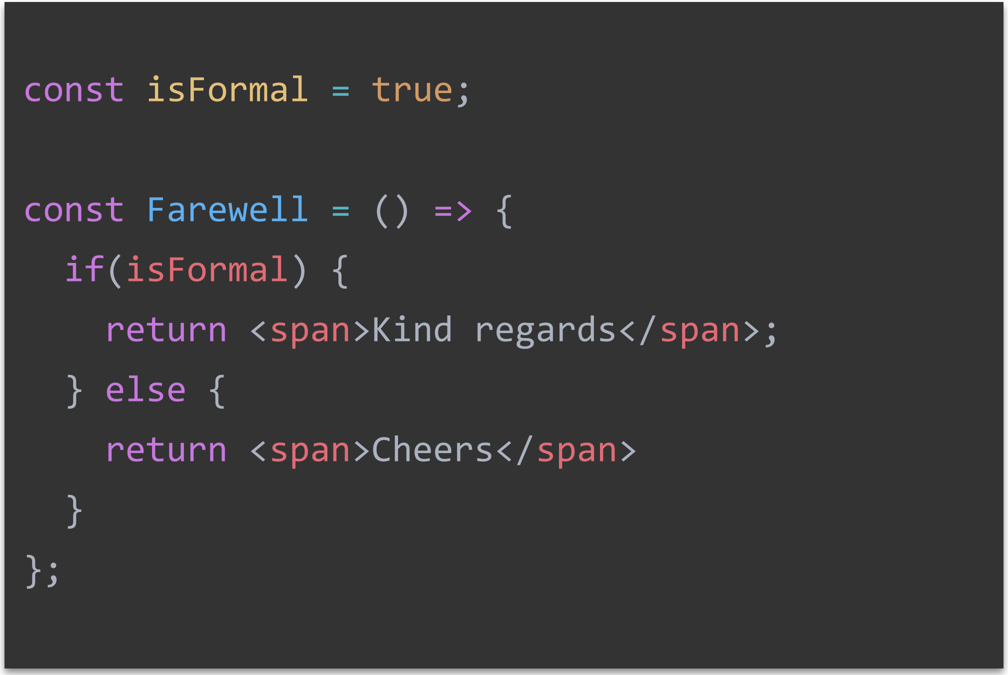
Dynamic render

Conditionals

Switch

Loops

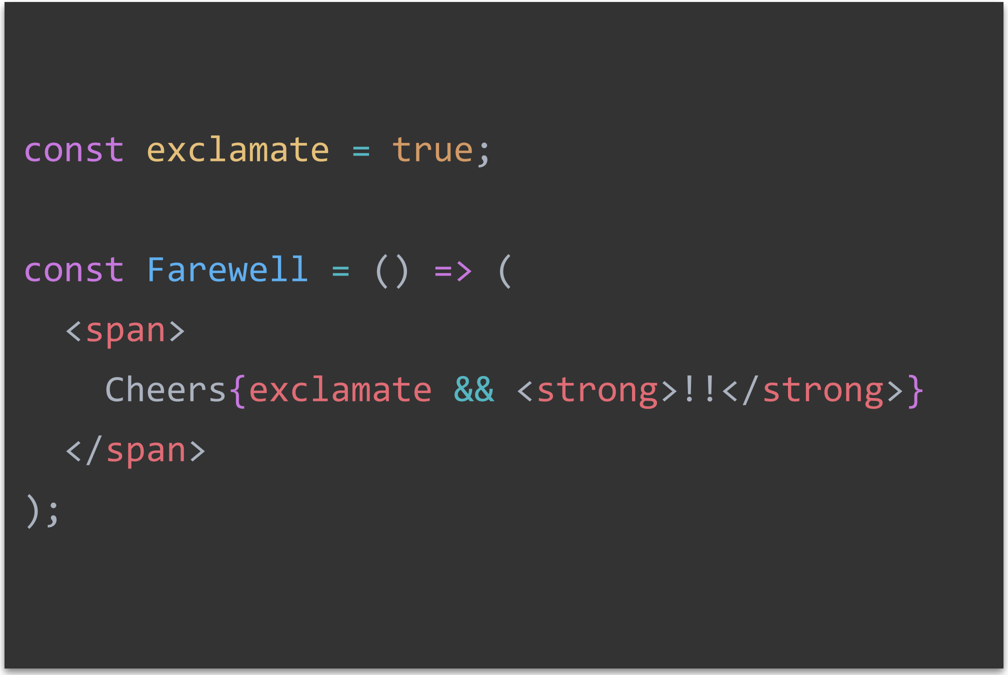
Dynamic render – Default conditional



```
const isFormal = true;

const Farewell = () => {
  if(isFormal) {
    return <span>Kind regards</span>;
  } else {
    return <span>Cheers</span>
  }
};
```

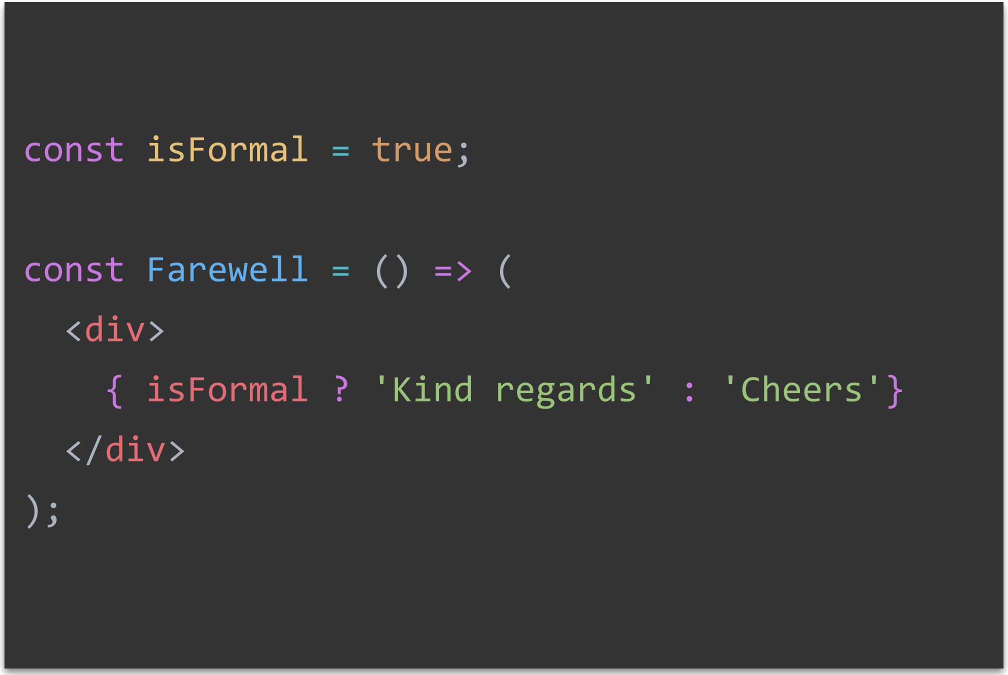
Dynamic render – Logical operator



```
const exclamate = true;

const Farewell = () => (
  <span>
    Cheers{exclamate && <strong>!!</strong>}
  </span>
);
```

Dynamic render - Ternary operator



```
const isFormal = true;

const Farewell = () => (
  <div>
    { isFormal ? 'Kind regards' : 'Cheers'}
  </div>
);
```

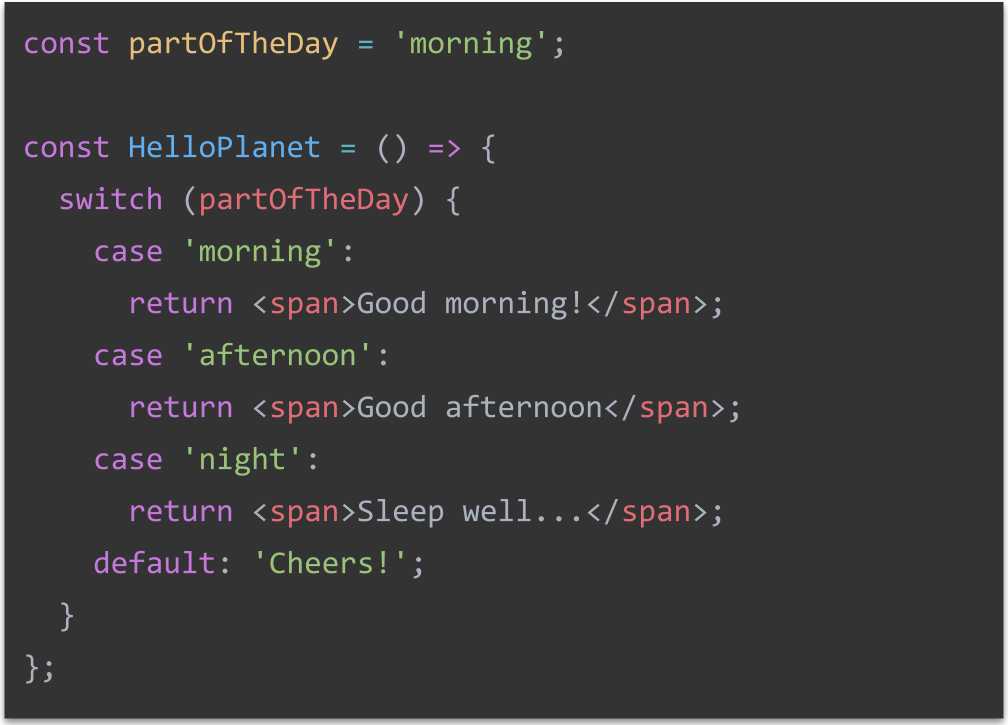
🕹️ Dynamic render

Conditionals

Switch

Loops

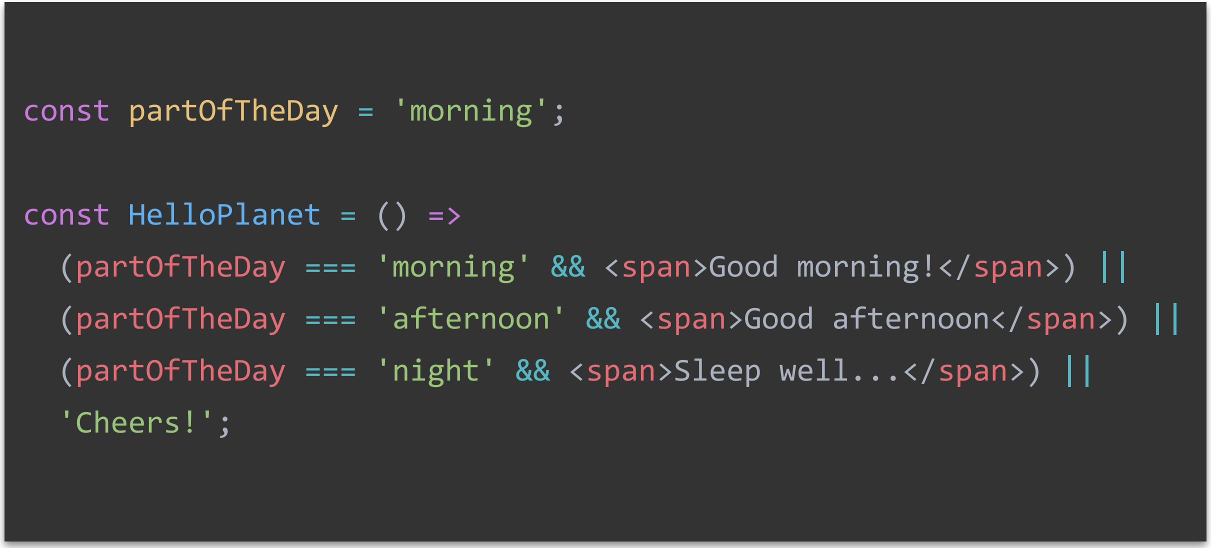
Dynamic render - Switch



```
const partOfTheDay = 'morning';

const HelloPlanet = () => {
  switch (partOfTheDay) {
    case 'morning':
      return <span>Good morning!</span>;
    case 'afternoon':
      return <span>Good afternoon</span>;
    case 'night':
      return <span>Sleep well...</span>;
    default: 'Cheers!';
  }
};
```

Dynamic render - Switch



```
const partOfTheDay = 'morning';

const HelloPlanet = () =>
  (partOfTheDay === 'morning' && <span>Good morning!</span>) ||
  (partOfTheDay === 'afternoon' && <span>Good afternoon</span>) ||
  (partOfTheDay === 'night' && <span>Sleep well...</span>) ||
  'Cheers!';
```

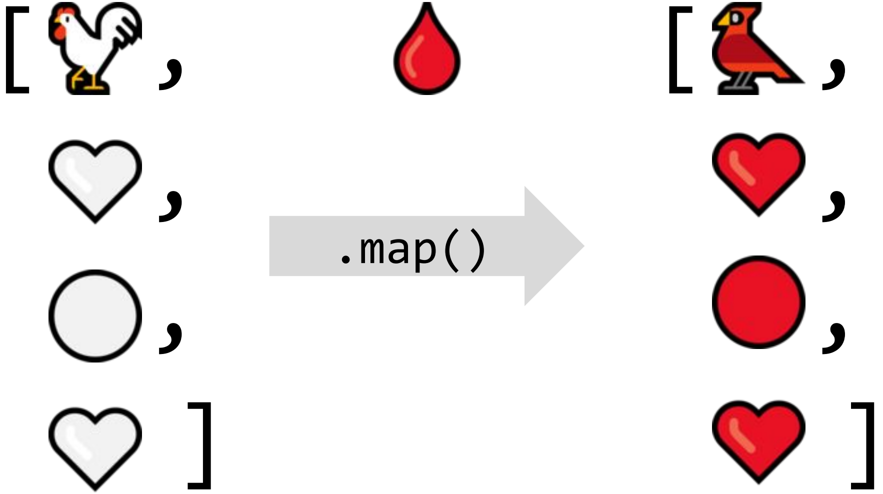
Dynamic render

Conditionals

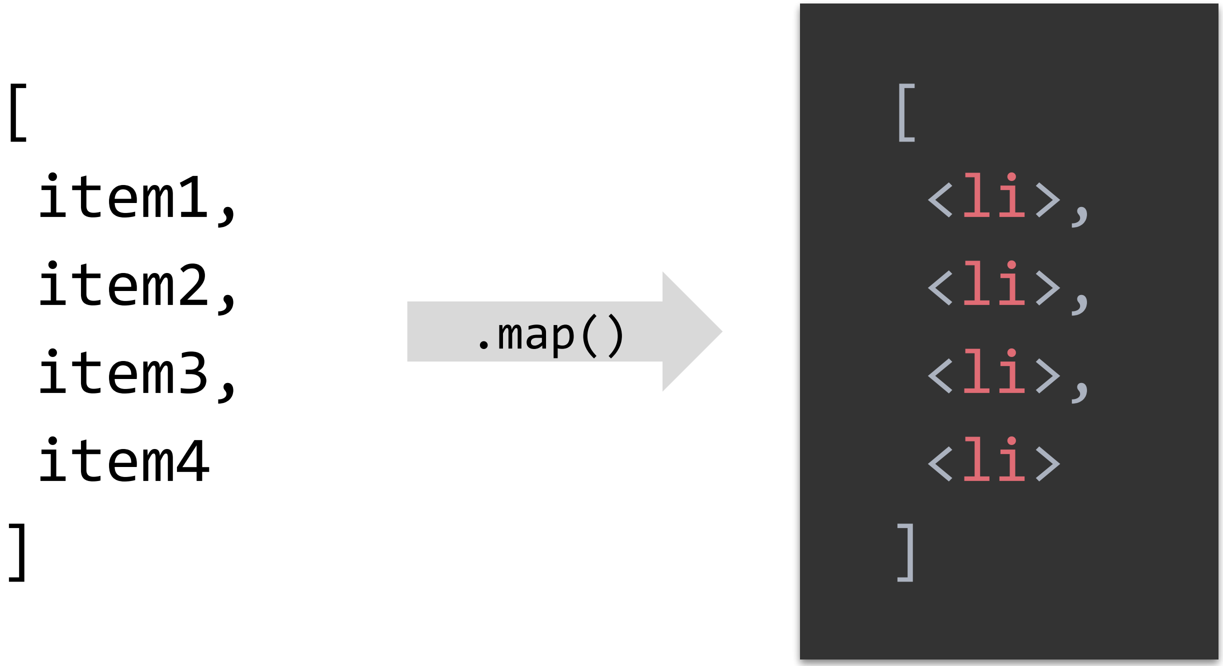
Switch

Loops

Dynamic render – Array map



Dynamic render – Array map

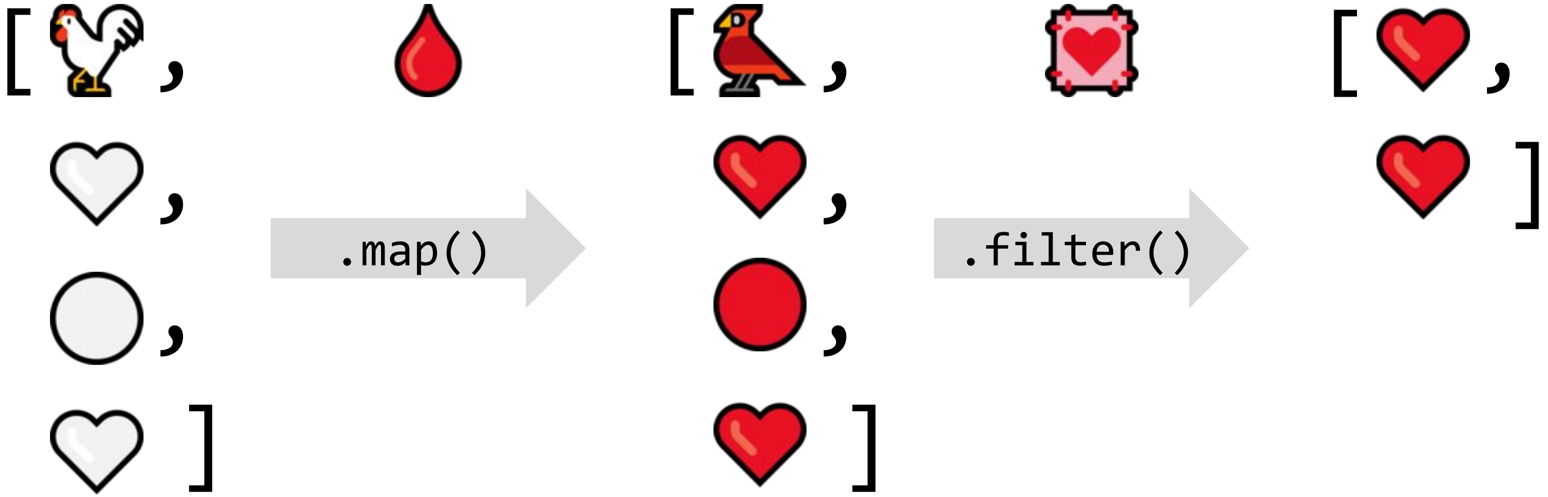


```
[
  item1,
  item2,
  item3,
  item4
]
```

.map()

```
[
  <li>,
  <li>,
  <li>,
  <li>
]
```

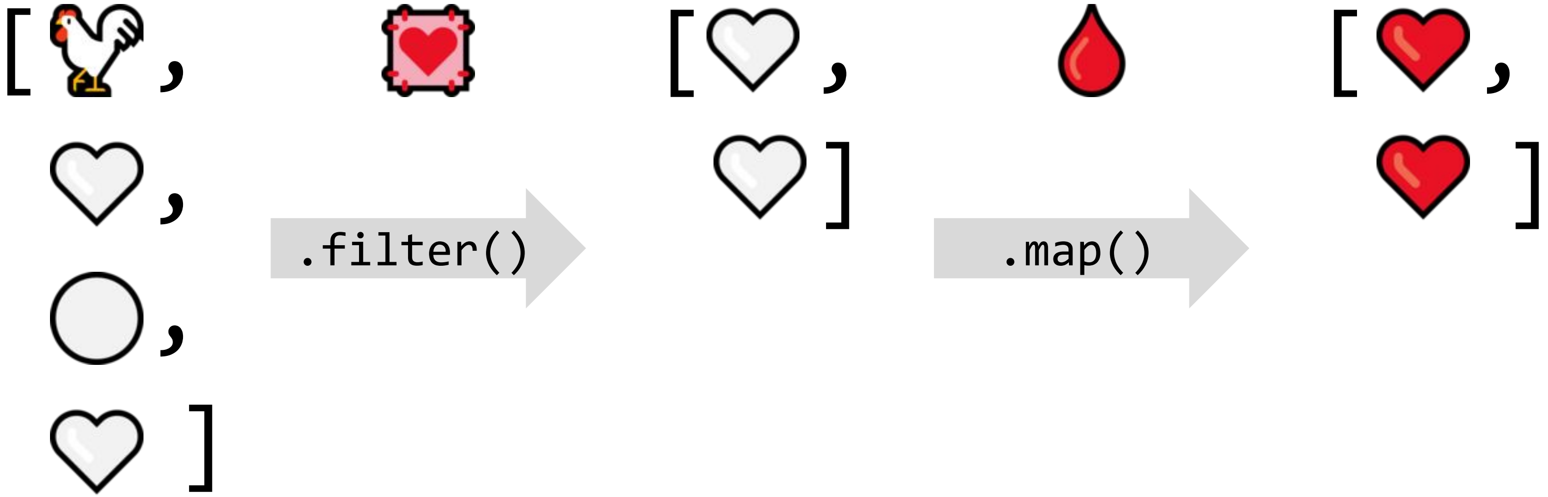
🕹️ Dynamic render – Array filter



.map()

.filter()

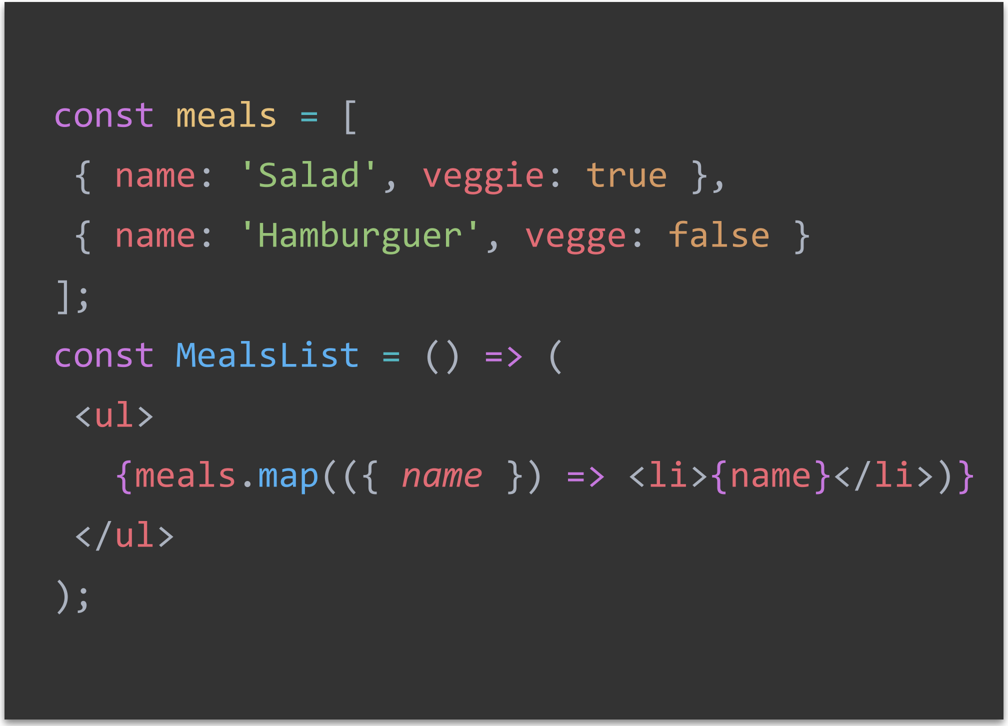
Dynamic render - Array filter



.filter()

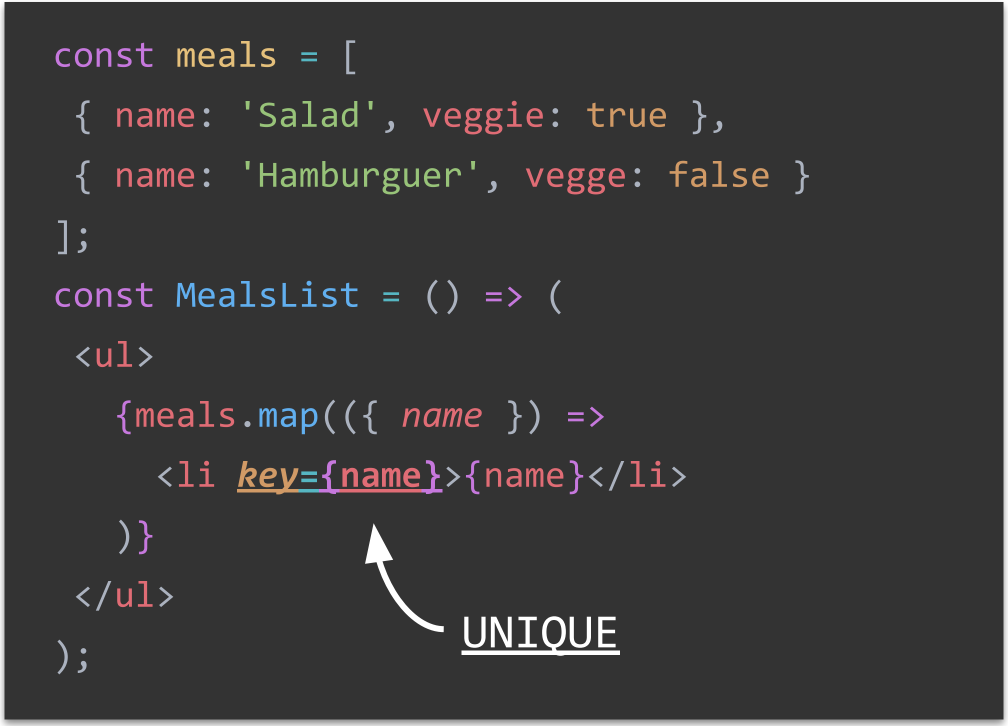
.map()

Dynamic render – Render loop



```
const meals = [
 { name: 'Salad', veggie: true },
 { name: 'Hamburguer', vegge: false }
];
const MealsList = () => (
 <ul>
   {meals.map(({ name }) => <li>{name}</li>)}
 </ul>
);
```

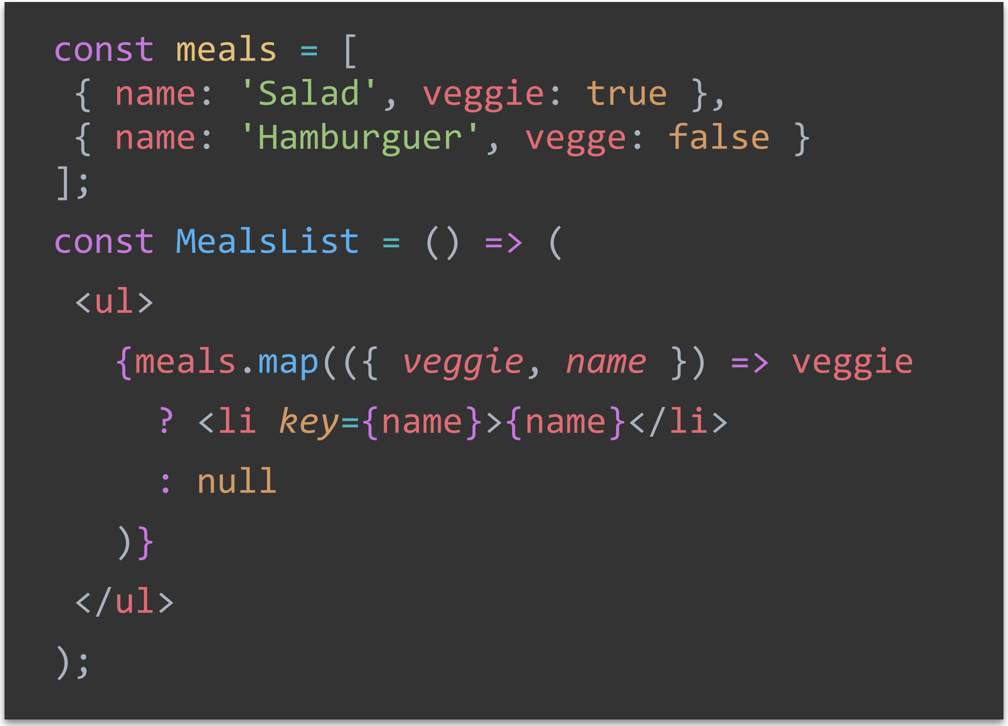
Dynamic render - Render loop



```
const meals = [
 { name: 'Salad', veggie: true },
 { name: 'Hamburguer', vegge: false }
];
const MealsList = () => (
 <ul>
   {meals.map(({ name }) =>
     <li key={name}>{name}</li>
   )}
 </ul>
);
```

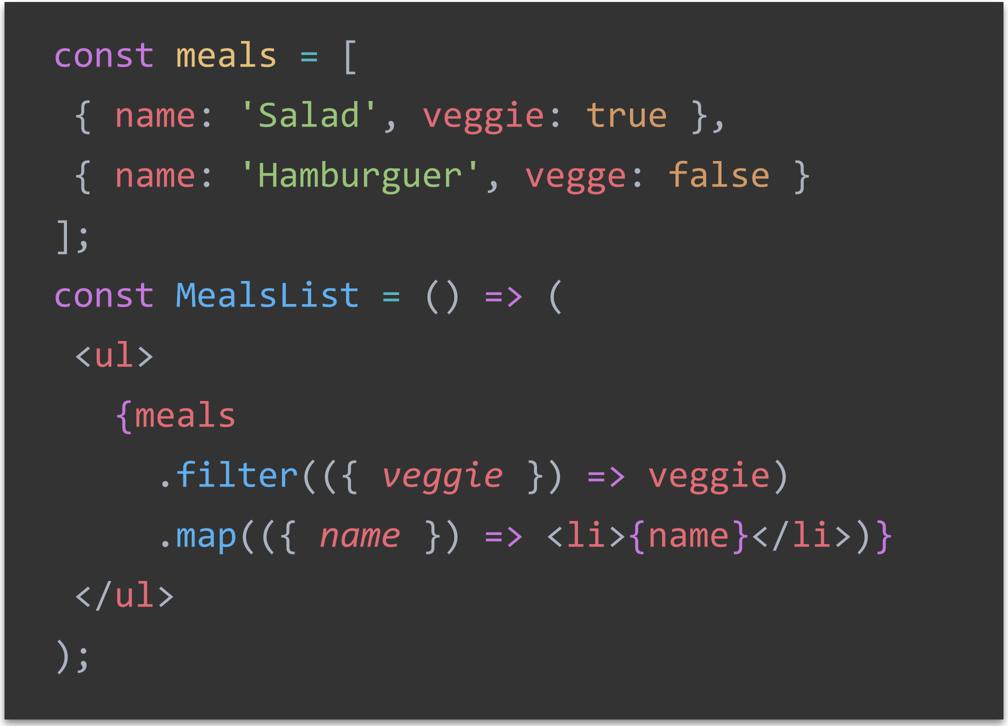
UNIQUE

Dynamic render - Render loop



```
const meals = [
 { name: 'Salad', veggie: true },
 { name: 'Hamburguer', vegge: false }
];
const MealsList = () => (
 <ul>
   {meals.map(({ veggie, name }) => veggie
     ? <li key={name}>{name}</li>
     : null
   )}
 </ul>
);
```

Dynamic render - Render loop



```
const meals = [
 { name: 'Salad', veggie: true },
 { name: 'Hamburguer', vegge: false }
];
const MealsList = () => (
 <ul>
   {meals
     .filter(({ veggie }) => veggie)
     .map(({ name }) => <li>{name}</li>)}
 </ul>
);
```

Index

- ✅ Components
- ✅ JSX
- ✅ Styling
- ✅ Dynamic render

👨‍🏫 Exercises!

1. Create an interface for a Wishlist application. Follow the JSX best practices commented.
   a. An input to create new wishes
   b. A list of the wishes created
   c. Checkboxes to mark a wish came true.
   d. Buttons to archive completed wishes

The wishes must be stored in an array.

2. Create styles for the main elements of the UI, using the BEM naming strategy.

3. Render

Introduction to React