


Ex 6 - Conditional Rendering

Challenge 1 :

Challenge 1 of 3: Show an icon for incomplete items with ? :

Use the conditional operator (`cond ? a : b`) to render a  if `isPacked` isn't `true`.

[App.js](#)Download Reset Fork

```
1 function Item({ name, isPacked }) {
2   return (
3     <li className="item">
4       {name} {isPacked ? '✓' : '✗'}
5     </li>
6   );
7 }
8
9 export default function PackingList() {
10  return (
11    <section>
12      <h1>Sally Ride's Packing List</h1>
13      <ul>
14        <Item
15          isPacked={true}
16          name="Space suit"
17        />
18      </ul>
19    </section>
20  );
21 }
```

Sally Ride's Packing List

- Space suit ✓
- Helmet with a golden leaf ✓
- Photo of Tam ✗

▼ Show more

Challenge 2:

Challenge 2 of 3: Show the item importance with &&

In this example, each `Item` receives a numerical `importance` prop. Use the `&&` operator to render “(*Importance: X*)” in italics, but only for items that have non-zero importance. Your item list should end up looking like this:

- Space suit (*Importance: 9*)
- Helmet with a golden leaf
- Photo of Tam (*Importance: 6*)

Don't forget to add a space between the two labels!

[App.js](#)Download Reset Fork

```
1 function Item({ name, importance }) {
2   return (
3     <li className="item">
4       {name} {importance > 0 &&
5         <i>(Importance: {importance})</i>}
6     </li>
7   );
8 }
9
10 export default function PackingList() {
11  return (
12    <section>
13      <h1>Sally Ride's Packing List</h1>
14      <ul>
15        <Item
16          importance={9}
17        />
18      </ul>
19    </section>
20  );
21 }
```

Sally Ride's Packing List

- Space suit (*Importance: 9*)
- Helmet with a golden leaf
- Photo of Tam (*Importance: 6*)

▼ Show more

Challenge 3 :

Challenge 3 of 3: Refactor a series of `? :` to `if` and variables

This `Drink` component uses a series of `? :` conditions to show different information depending on whether the `name` prop is `"tea"` or `"coffee"`. The problem is that the information about each drink is spread across multiple conditions. Refactor this code to use a single `if` statement instead of three `? :` conditions.

App.js

[Download](#) [Reset](#) [Fork](#)

```
1 function Drink({ name }) {
2   let part, content, age;
3   if (name === "tea") {
4     part = 'leaf';
5     content = '15-70 mg/cup';
6     age = '4,000+ years';
7   } else if (name === "coffee") {
8     part = 'bean';
9     content = '80-185 mg/cup';
10    age = '1,000+ years';
11  }
12  return (
13    <section>
14      <h1>{name}</h1>
15      <dl>
16        <dt>Part of plant</dt>
17        <dd>{part}</dd>
18        <dt>Caffeine content</dt>
19        <dd>{content}</dd>
20        <dt>Age</dt>
21        <dd>{age}</dd>
22      </dl>
23    </section>
24  );
25 }
```

tea

Part of plant
leaf
Caffeine content
15-70 mg/cup
Age
4,000+ years

coffee

Part of plant
bean
Caffeine content
80-185 mg/cup
Age
1,000+ years