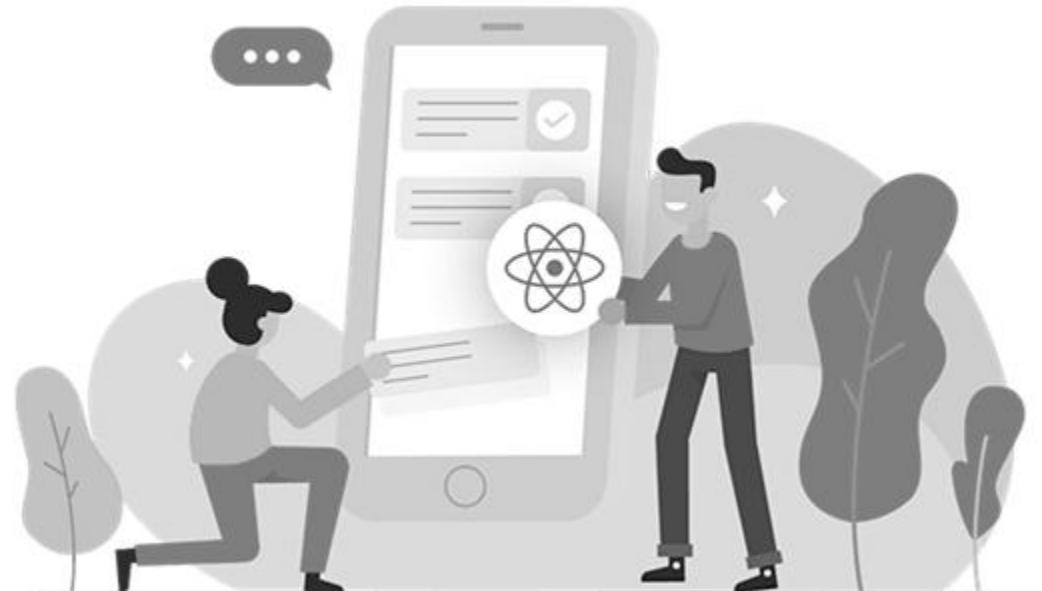


FRONT END DEVELOPMENT

WEEK 6 – Props & Lists





Watch the video and answer questions

Use **props**

<https://www.youtube.com/watch?v=hpZmKR9FmQw>

https://www.w3schools.com/react/react_props.asp

Q1 - What are the benefits of using props ?

Q2 – How many props can be passed to a component?

Render a **list**

<https://www.youtube.com/watch?v=f640Z6QZawc>

https://www.w3schools.com/react/react_lists.asp

*Q3 - Why React requires the **key attribute to be set** when displaying a list of objects?*

Q4 - What are the 2 ways to loop on array the author proposes in the video ?



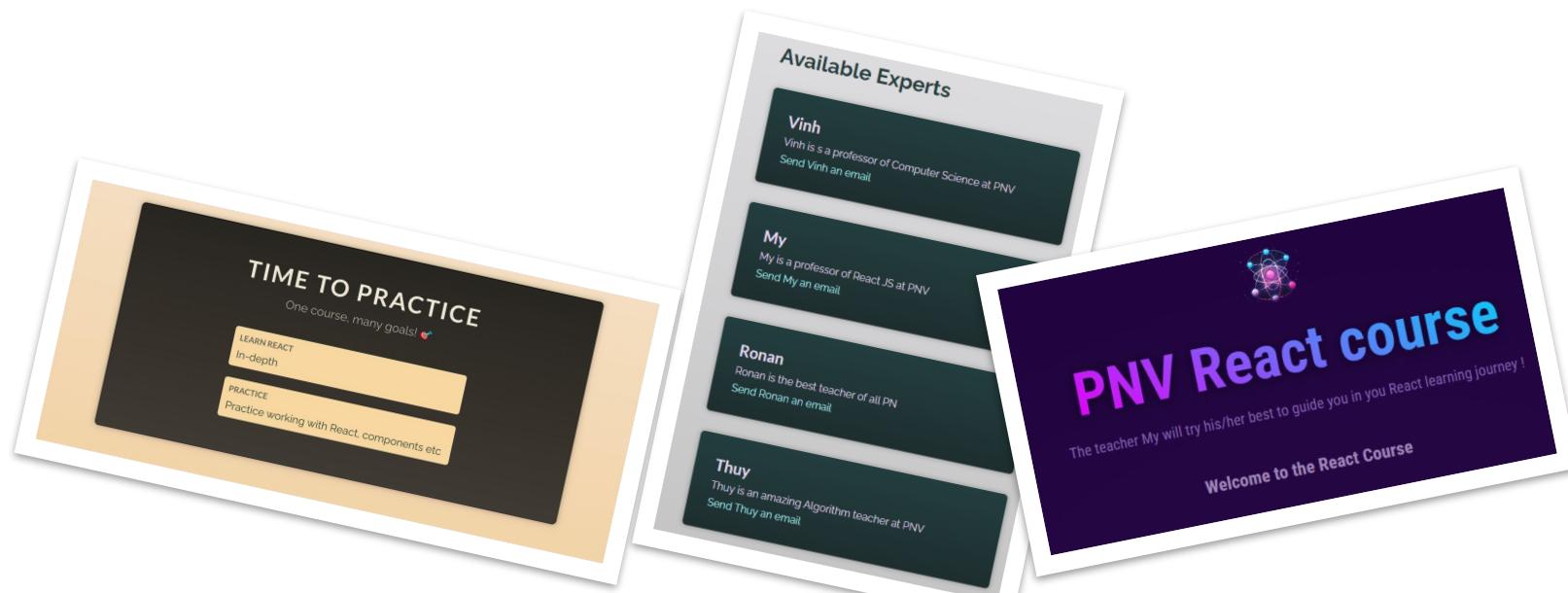


What will you learn today?



- ✓ Use **props** to **re-use** your components
 - ✓ Destructure **props**

- ✓ Render a list
 - ✓ Understand the **key** attribute on loops





10 MIN

Group discussion about the homework

Share your answer, discuss in teams

Use **props**

Q1 - What are the benefits of using props ?

Q2 – How many props can be passed to a component?

Render a **list**

Q3 - Why React requires the **key attribute to be set when displaying a list of object?**

Q4 - What are the 2 way to loop on array the author proposes in the video ?



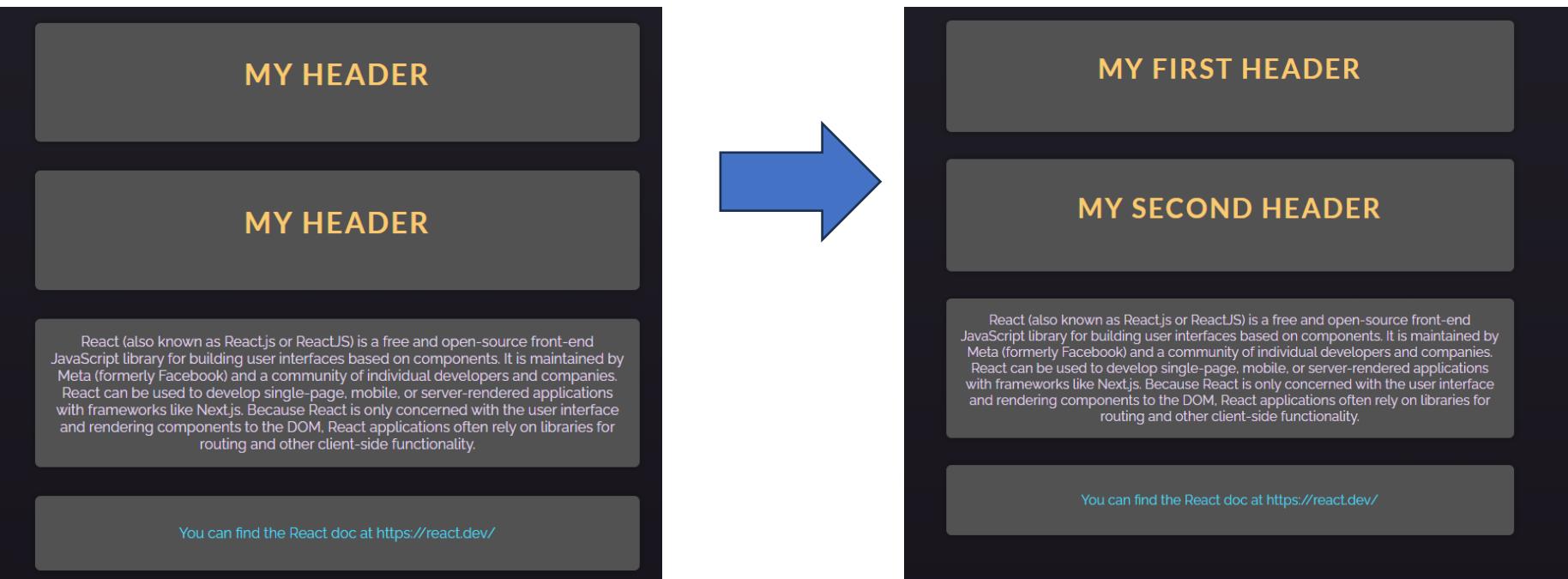


10 MIN

Activity 1

Working with props

- ✓ Connect to the sandbox
- ✓ How can we **re-use** the **same component Header**, but with a **different title** ?



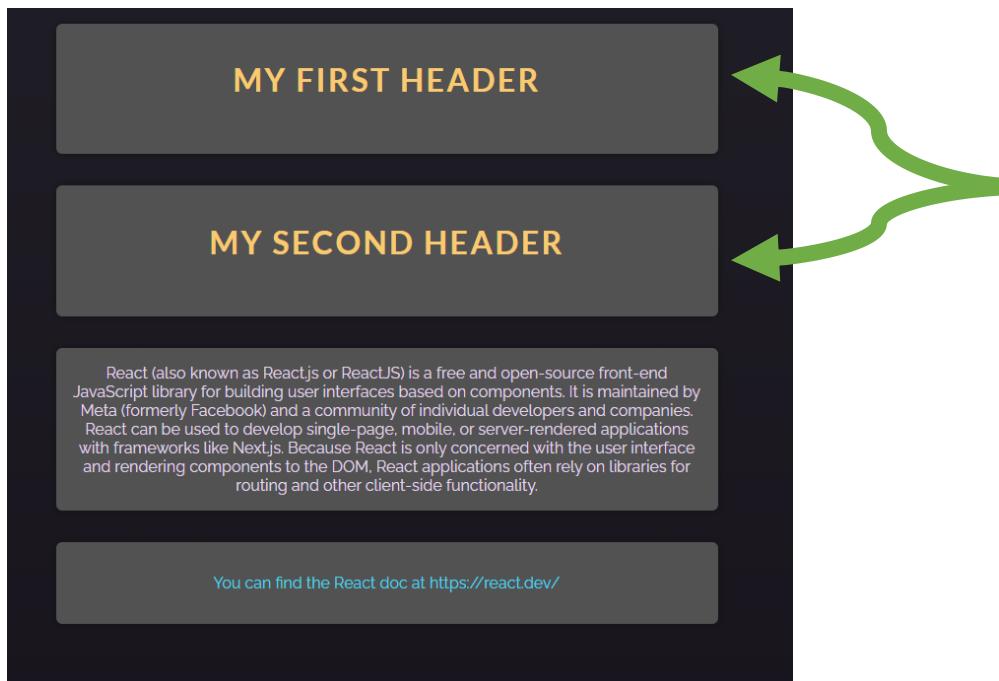


You Can **Reuse** React Components

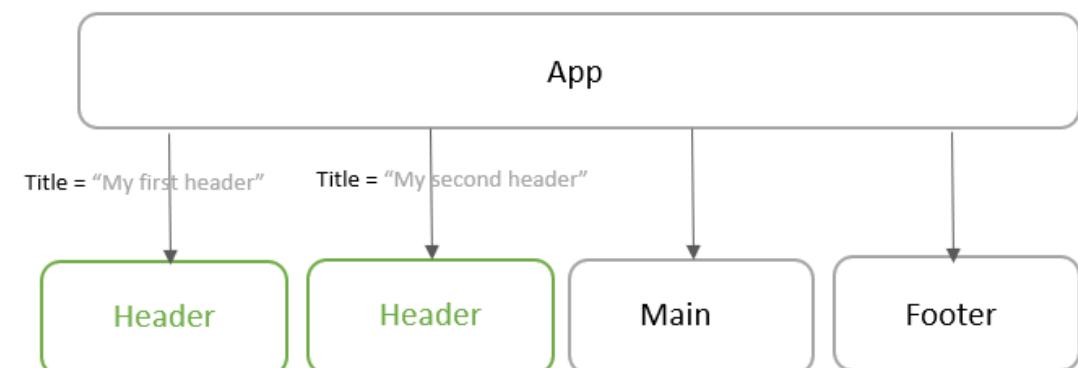


10 MIN

*A same component can be used **multiple times***



*Header is used 2 times
With different data*



Configuring Components with *props*

*Props is used to **pass data** between **parent** and **child** components*



```
<Header title="my first header"/>
```

```
export default function Header(props) {  
  return (  
    <header className="block">  
      <h1>{props.title}</h1>  
    </header>  
  );  
}
```



10 MIN

Activity 2

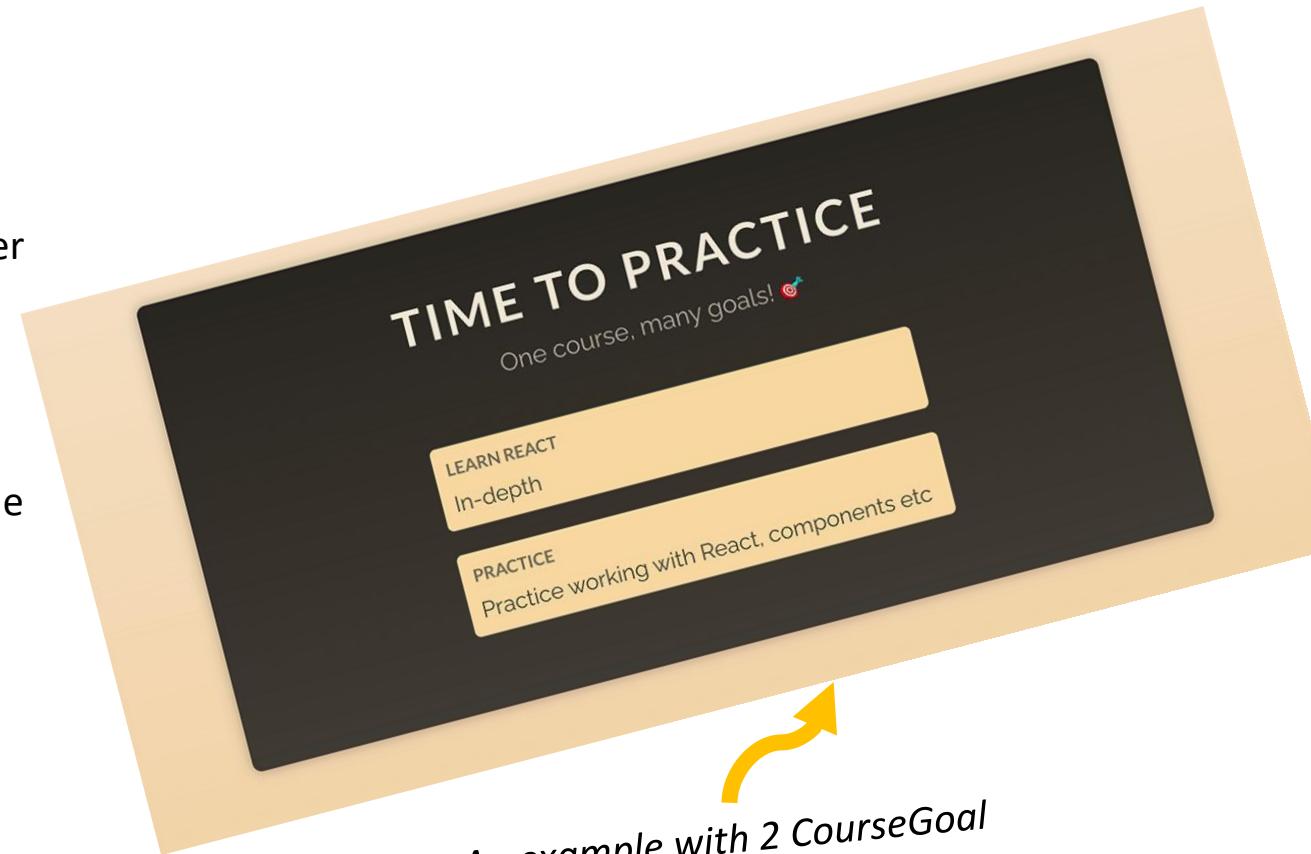
Working with props

1 – Connect to the sandbox

2 - Make the **CourseGoal** component **reusable**.

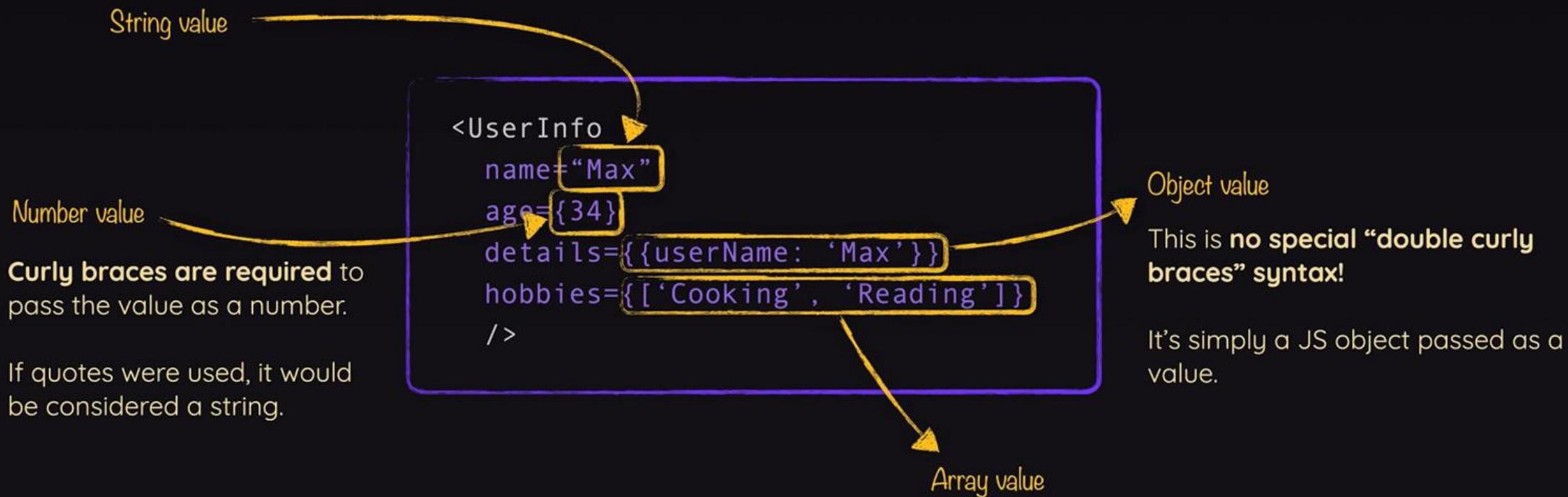
- ✓ It should accept a "*title*" and a "*description*" as parameter
- ✓ Display those values on
 - <h2> (*title*)
 - <p> (*description*)

3 - The **App** component should render at least two instances of the CourseGoal component.



An example with 2 CourseGoal

You are not limited to text values !

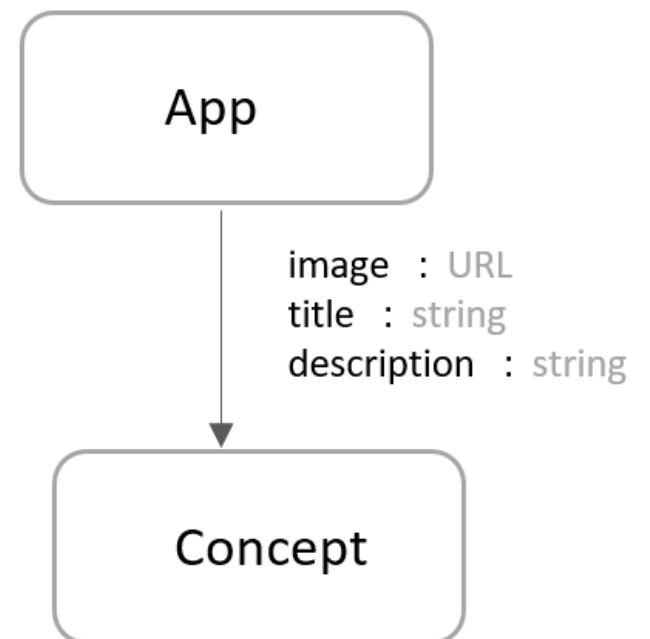


Props with many parameters

1 – Connect to the sandbox

The component Concept takes 3 parameters: the image, title and description

2 – We are going to complete the code to *instantiate* 3 component Concepts

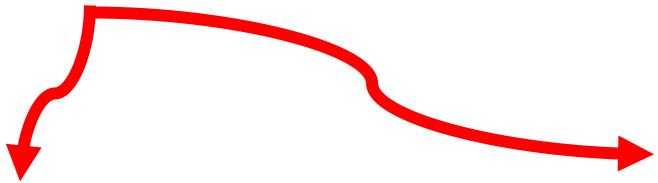


Activity 3

Destructuring props

Let's simplify how we write props using the ES6 **Destructuring** syntax

```
function Concept(props) {  
  return (  
    <li>  
      <img src={props.image} />  
      <h3>{props.title}</h3>  
      <p>{props.description}</p>  
    </li>  
  );  
}
```



```
function Concept( { image, title, description } ) {  
  return (  
    <li>  
      <img src={image} />  
      <h3>{title}</h3>  
      <p>{description}</p>  
    </li>  
  );  
}
```



10 MIN

Activity 4

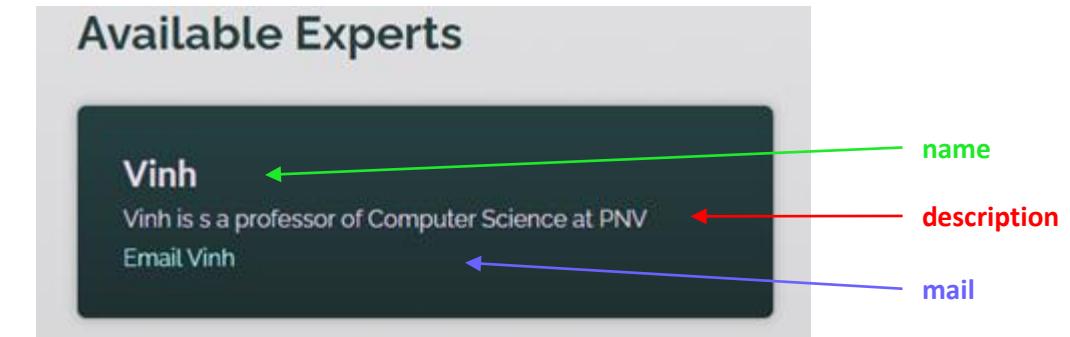
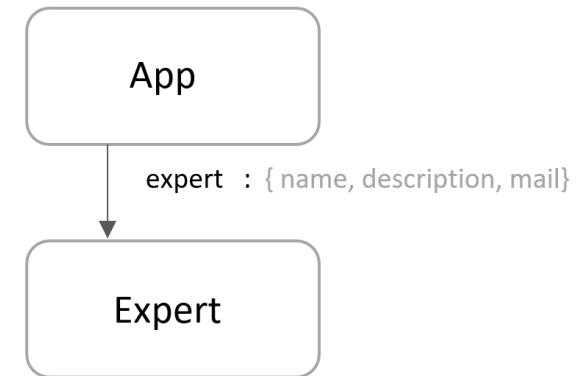
Props with many parameters

1 – Connect to the sandbox

2 - Make the **Expert** component **reusable**.

- ✓ It should accept a " expert " as parameter
- ✓ It should display the expert name, description and mail
- ✓ Use the **Destructuring** to get props parameter

3 - The **App** component should render two instances of the **Expert** component (Vinh and My)



```

<div className="card">
  <h2>Vinh</h2>
  <p>Vinh is a professor of Computer Science at PNV</p>
  <a href="mailto:vinh.hoang@passerellesnumeriques.org">Email Vinh</a>
</div>
  
```



4 people

10 MIN

Rendering lists

- ✓ Connect to the sandbox
- ✓ How can we **loop** on JSX to create a component for all **array items** ?

The image shows a split-screen interface. On the left, a code editor displays a snippet of React code. A blue curved arrow labeled "Array items" points from the text "Fundamental React concepts you will need for almost any app you are going to bu" towards the code. On the right, a browser window displays a slide titled "React Essentials" with the subtitle "Fundamental React concepts you will need for almost any app you are going to build". A blue curved arrow labeled "Component For all array items" points from the title back towards the code editor.

```
import propsImg from "./assets/props.png";
import jsxImg from "./assets/jsx-ui.png";

const ALL_CONCEPTS = [
  {
    image: componentsImg,
    title: "Components",
    description:
      "The core UI building block - compose the user interface by combining multiple components."
  },
  {
    image: propsImg,
    title: "Props",
    description:
      "Make components configurable (and therefore reusable) by passing input data to them."
  },
  {
    image: jsxImg,
    title: "JSX",
    description:
      "Return (potentially dynamic) HTML(ish) code to define the actual markup that will be rendered"
  }
];

function App() {
  return (
    <div>
      <h1>React Essentials</h1>
      <p>Fundamental React concepts you will need for almost any app you are going to bu</p>
    </div>
}
```



10 MIN

Rendering lists

LOOP INSIDE THE JSX

```
function App() {  
  return (  
    <div>  
      <Header />  
      <main>  
        <section id="core-concepts">  
          <ul>  
            {myArray.map((myItem) => (  
              <Concept  
                image={myItem.image}  
                title={myItem.title}  
              />  
            ))}  
          </ul>  
        </section>  
      </main>  
    </div>  
  )  
}
```

LOOP OUTSIDE THE JSX

```
const myConcepts = myArray.map((myItem) => (  
  <Concept  
    image={myItem.image}  
    title={myItem.title}  
  />));  
  
function App() {  
  return (  
    <div>  
      <Header />  
      <main>  
        <section id="core-concepts">  
          <ul>  
            { myConcepts }  
          </ul>  
        </section>  
      </main>  
    </div>  
  )  
}
```

Rendering lists: the **Key attribute**

Open the console: you should will see a warning that there is no "key" provided for the list items.

```
✖ ► Warning: Each child in a list should have a unique "key" prop.  
Check the render method of `App`. See https://reactjs.org/link/warning-keys for more information.  
  at Concept (http://127.0.0.1:5173/src/components/Concept.jsx:18:3)  
  at App
```

Keys allow React to keep track of elements.

Keys need to be unique to each sibling. But they can be duplicated globally.

```
myArray.map((myItem, index) => (  
  <Concept  
    key = index  
    image={myItem.image}  
    title={myItem.title}  
  />))}
```



10 MIN

Activity 5

Rendering lists

1 – Connect to the sandbox

Note : Nothing to change on **Expert** component (done already).

2 - The **App** component should render the list of expert using the **Expert** component

Warning : Don't forget to set the key of the component **Expert** when looping:

```
<Expert key={index} >
```

Available Experts

Vinh

Vinh is a professor of Computer Science at PNV
[Send Vinh an email](#)

My

My is a professor of React JS at PNV
[Send My an email](#)

Ronan

Ronan is the best teacher of all PN
[Send Ronan an email](#)

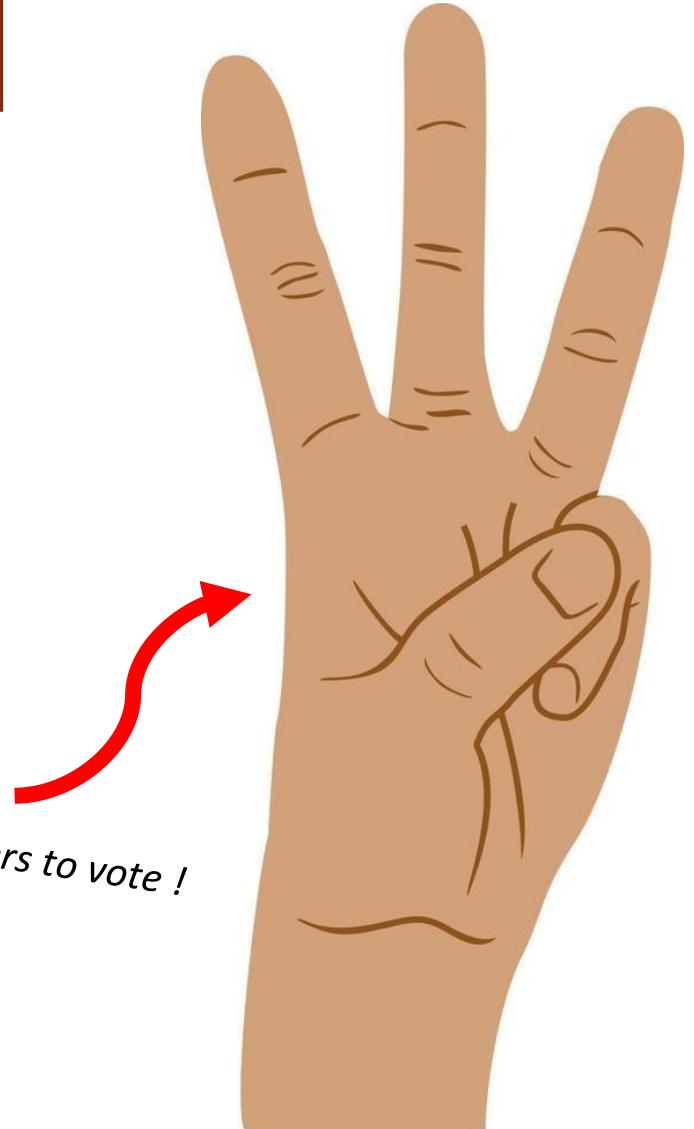
Thuy

Thuy is an amazing Algorithm teacher at PNV
[Send Thuy an email](#)

Time For

QUIZ !

Use your fingers to vote !



Q1

Which syntax can be used to output dynamic values / JavaScript expressions in JSX?

1

[] (Square brackets)

2

() (Parentheses)

3

{ } (Curly braces)

4

{{ }} (Double curly braces)

Q1

Which syntax can be used to output dynamic values / JavaScript expressions in JSX?

1

[] (Square brackets)

2

() (Parentheses)

3

{ } (Curly braces)

4

{{ }} (Double curly braces)

Q2

Which values can be output as dynamic values in JSX (i.e., between curly braces)?

1

- Any valid JavaScript statement

2

- Any valid React template instruction

3

- Any valid JavaScript expression

4

- Any valid React template expression

Q2

Which values can be output as dynamic values in JSX (i.e., between curly braces)?

1

- Any valid JavaScript statement

2

- Any valid React template instruction

3

- Any valid JavaScript expression

4

- Any valid React template expression

Q3

How should you typically load / use images in React projects?

1

- Set the ` src` equal to a relative string path to the image

2

- Set the ` src` equal to the path generated via an import statement (that points at the image)

3

- Set the ` src` equal to an absolute string path to the image

4

- Set the ` src` equal to the image itself

Q3

How should you typically load / use images in React projects?

1

- Set the ` src` equal to a relative string path to the image

2

- Set the ` src` equal to the path generated via an import statement (that points at the image)

3

- Set the ` src` equal to an absolute string path to the image

4

- Set the ` src` equal to the image itself

Q4

How can you assign a dynamic value to an HTML element attribute?

1

- Via the same syntax you use for dynamic values in JSX in general: Curly braces

2

- Via the special React attribute directive that can be added to all built-in and custom components

3

- Via the same syntax you use for attributes in general: You set the value as a hard-coded string

4

- Via the special React template attribute syntax that allows you to store dynamic values for attributes

Q4

How can you assign a dynamic value to an HTML element attribute?

1

- Via the same syntax you use for dynamic values in JSX in general: Curly braces

2

- Via the special React attribute directive that can be added to all built-in and custom components

3

- Via the same syntax you use for attributes in general: You set the value as a hard-coded string

4

- Via the special React template attribute syntax that allows you to store dynamic values for attributes

Q5

Which core React concept can increase the reusability of React components?

1



2



3



4



Q5

Which core React concept can increase the reusability of React components?

1

JSX

2

Classes

3

Functions

4

Props

Q6

How do you pass props to a component in ReactJS?

- 1** By using the `setState()` method
- 2** By passing them directly as attributes in the component's JSX tag
- 3** By using the `this.props` object inside the component
- 4** By importing them from a separate props file

Q6

How do you pass props to a component in ReactJS?

- 1 By using the `setState()` method
- 2 By passing them directly as attributes in the component's JSX tag
- 3 By using the `this.props` object inside the component
- 4 By importing them from a separate props file

Q7

Which of the following is an example of passing a prop to a component in ReactJS?

- 1** <MyComponent myProp= "myValue" />

- 2** <MyComponent myProp: "myValue" />

- 3** <MyComponent {myProp="myValue"} />

- 4** <MyComponent (myProp="myValue") />

Q7

Which of the following is an example of passing a prop to a component in ReactJS?

- 1** `<MyComponent myProp= "myValue" />`
- 2** `<MyComponent myProp: "myValue" />`
- 3** `<MyComponent {myProp="myValue"} />`
- 4** `<MyComponent (myProp="myValue") />`

Q8

What does this component do?

```
jsx
function Greeting(props) {
  return <h1>Hello, {props.name}!</h1>;
}
```

- 1 It **renders** a greeting message using the value of `props.name`
- 2 It **sets the value** of `props.name` to "Hello"
- 3 It throws an error because components **cannot accept props**
- 4 It returns a **function** instead of JSX.

Q8

What does this component do?

```
jsx
function Greeting(props) {
  return <h1>Hello, {props.name}!</h1>;
}
```

- 1 It **renders** a greeting message using the value of `props.name`
- 2 It **sets the value** of `props.name` to "Hello"
- 3 It throws an error because components **cannot accept props**
- 4 It returns a **function** instead of JSX.

Q9

What does this component do?

```
jsx Copy code
<MyComponent name="John" age={25} />
```

- 1** Renders a component MyComponent with props name set to "John" and age set to 25
- 2** Renders a component MyComponent with props name set to {25} and age set to "John"
- 3** Throws an error because props should be passed as strings (and 25 is an integer....)
- 4** Renders a component MyComponent without any props

Q9

What does this component do?

jsx

`<MyComponent name="John" age={25} />`

 Copy code

- 1 Renders a component MyComponent with props name set to "John" and age set to 25
- 2 Renders a component MyComponent with props name set to {25} and age set to "John"
- 3 Throws an error because props should be passed as strings (and 25 is an integer....)
- 4 Renders a component MyComponent without any props