Rendering Lists

Using the map() Function to Render Lists

LEARNING OBJECTIVES

- Render lists of components using the map () method
- Understand the importance of "key" attribute
- Use the fragments feature for rendering multiple top level components

Using the map () function to render lists

COMMON FEATURE

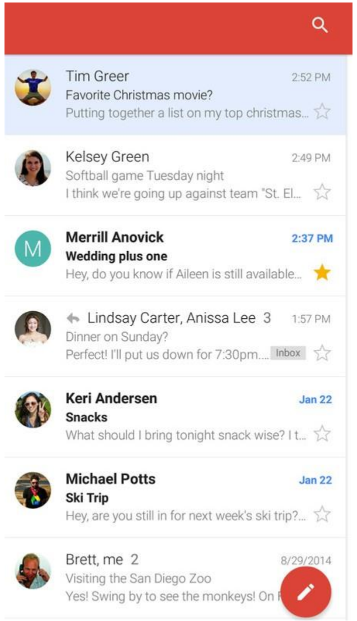
What is the most common type of feature that we see these days in web applications and mobile apps?

COMMON FEATURE

Lists

LISTS ARE EVERYWHERE

Gmail



Tim Greer 2:52 PM
Favorite Christmas movie?
Putting together a list on my top christmas...

Kelsey Green 2:49 PM
Softball game Tuesday night
I think we're going up against team "St. El...

Merrill Anovick 2:37 PM
Wedding plus one
Hey, do you know if Aileen is still available...

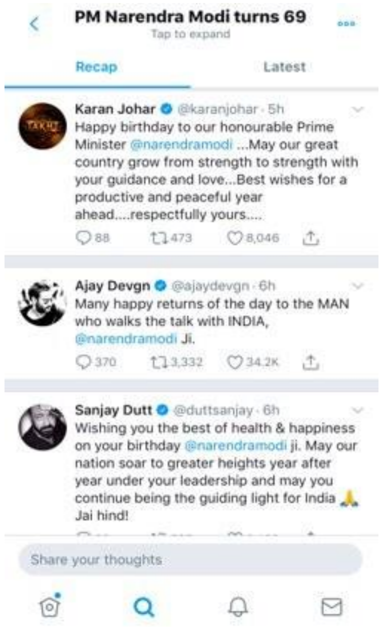
Lindsay Carter, Anissa Lee 3 1:57 PM
Dinner on Sunday?
Perfect! I'll put us down for 7:30pm... Inbox

Keri Andersen Jan 22
Snacks
What should I bring tonight snack wise? I t...

Michael Potts Jan 22
Ski Trip
Hey, are you still in for next week's ski trip?...

Brett, me 2 8/29/2014
Visiting the San Diego Zoo
Yes! Swing by to see the monkeys! On F

Twitter



PM Narendra Modi turns 69
Tap to expand

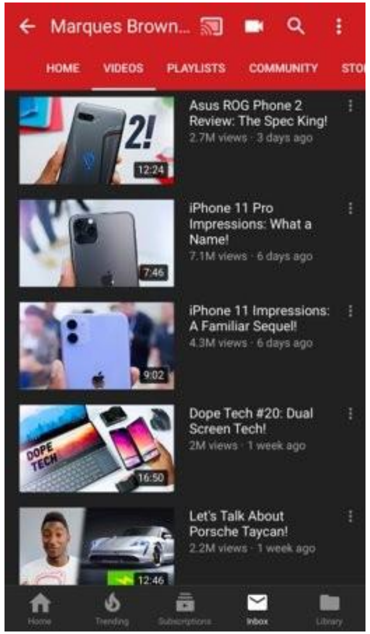
Karan Johar @karanjohar · 5h
Happy birthday to our honourable Prime Minister @narendramodi ...May our great country grow from strength to strength with your guidance and love...Best wishes for a productive and peaceful year ahead....respectfully yours....
88 473 8,046

Ajay Devgn @ajaydevgn · 6h
Many happy returns of the day to the MAN who walks the talk with INDIA, @narendramodi Ji.
370 3,332 34.2K

Sanjay Dutt @duttsanjay · 6h
Wishing you the best of health & happiness on your birthday @narendramodi ji. May our nation soar to greater heights year after year under your leadership and may you continue being the guiding light for India 🙏 Jai hind!

Share your thoughts

YouTube



Marques Brown...

Asus ROG Phone 2 Review: The Spec King!
2.7M views · 3 days ago
12:24

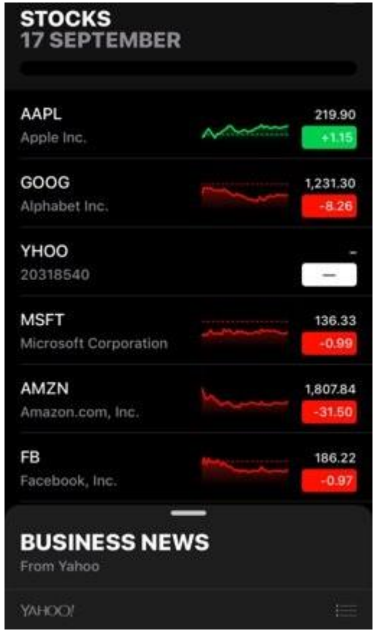
iPhone 11 Pro Impressions: What a Name!
7.1M views · 6 days ago
7:46

iPhone 11 Impressions: A Familiar Sequel!
4.3M views · 6 days ago
9:02

Dope Tech #20: Dual Screen Tech!
2M views · 1 week ago
16:50

Let's Talk About Porsche Taycan!
2.2M views · 1 week ago
12:46

Stock Trading



STOCKS
17 SEPTEMBER

AAPL Apple Inc.	219.90 +1.15
GOOG Alphabet Inc.	1,231.30 -8.26
YHOO 20318540	- —
MSFT Microsoft Corporation	136.33 -0.99
AMZN Amazon.com, Inc.	1,807.84 -31.50
FB Facebook, Inc.	186.22 -0.97

BUSINESS NEWS
From Yahoo

YAHOO!

WHAT IS A LIST?

A list is a collection of items which are mapped to instances of **UI elements** such as React components.

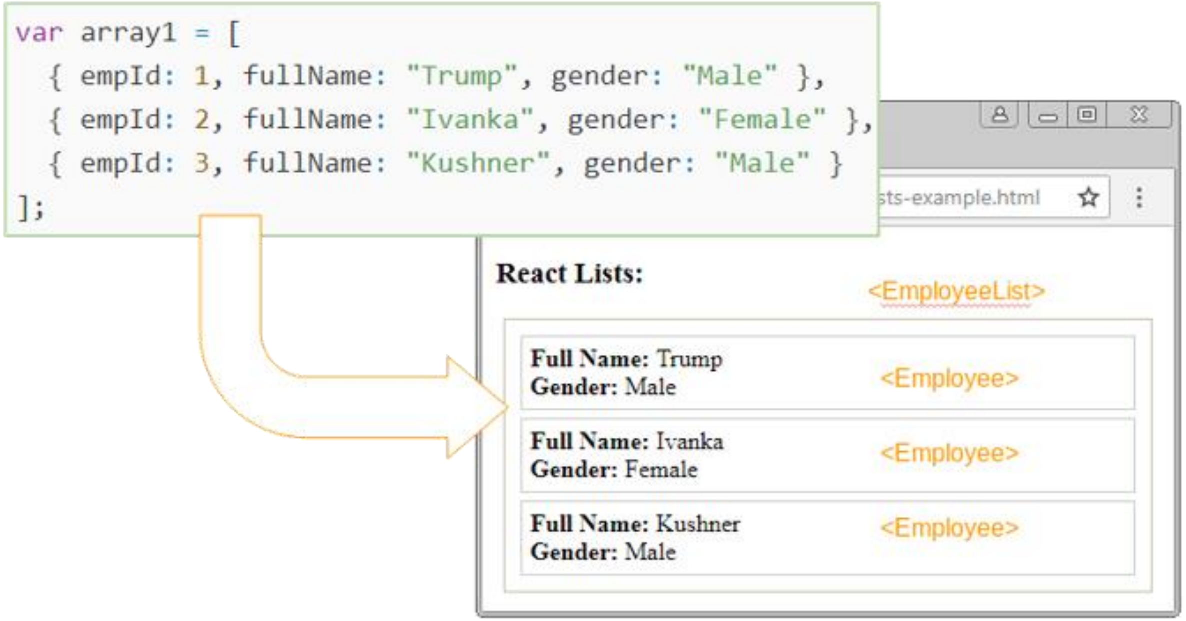
JavaScript **map ()** function is used to translate a collection of items to renderable elements.

Hands-On

Why use map () function?

WHY USE THE map () FUNCTION?

A collection of items in an Array can be remodeled using map () function



```
var array1 = [
  { empId: 1, fullName: "Trump", gender: "Male" },
  { empId: 2, fullName: "Ivanka", gender: "Female" },
  { empId: 3, fullName: "Kushner", gender: "Male" }
];
```

React Lists:

Full Name: Trump
Gender: Male

Full Name: Ivanka
Gender: Female

Full Name: Kushner
Gender: Male

Hands-On

SUMMARY

- Map function produces an Array of React elements, based on a data collection provided
- Declarative rendering is when the data comes from the state and is updated, the list on the UI will automatically update and reflect the change
- Rendering component instances uses key attribute set to unique ID for every item in list

Rendering Lists

The 'Key' Attribute

COMMON LIST OPERATIONS

Adding elements

Editing elements

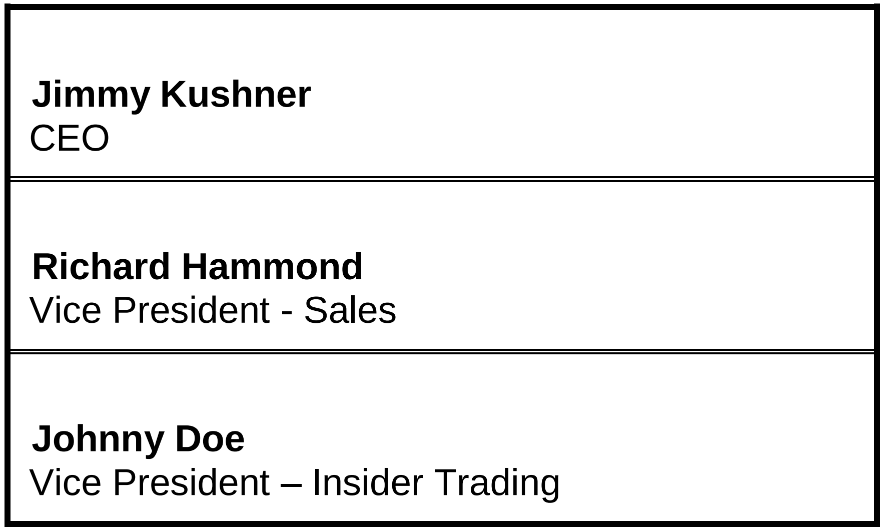
Removing elements

Sorting elements

DECLARATIVE PROGRAMMING

```
const employees = [
  { empId: 1, fullName: "Jimmy Kushner", jobTitle: "CEO" },
  { empId: 2, fullName: "Richard Hammond", jobTitle: "Vice President - Sales" },
  { empId: 3, fullName: "Johnny Doe", jobTitle: "Vice President - Insider Trading" }]
```

```
employees.map(emp =>
 <ListComponent data={emp} />)
```

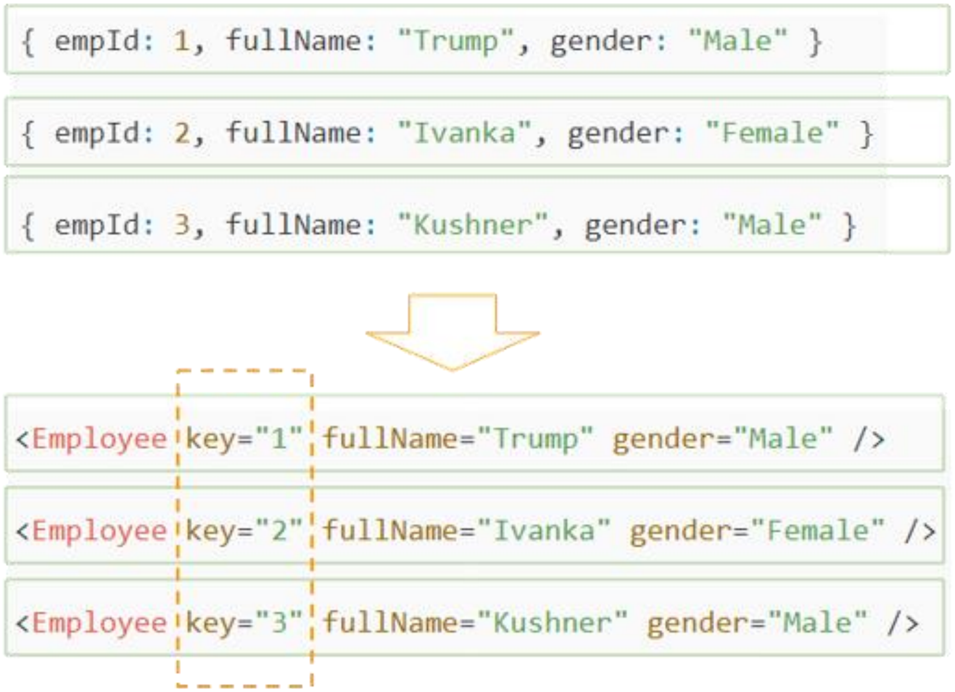


Jimmy Kushner
CEO

Richard Hammond
Vice President - Sales

Johnny Doe
Vice President – Insider Trading

THE KEY ATTRIBUTE

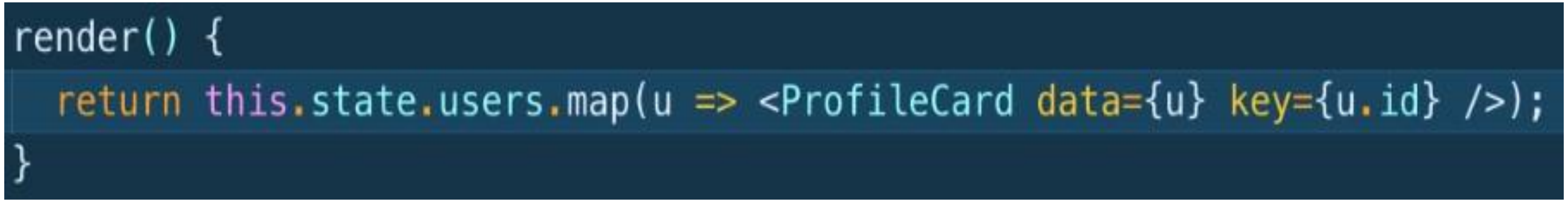


```
{ empId: 1, fullName: "Trump", gender: "Male" }
{ empId: 2, fullName: "Ivanka", gender: "Female" }
{ empId: 3, fullName: "Kushner", gender: "Male" }
```

```
<Employee key="1" fullName="Trump" gender="Male" />
<Employee key="2" fullName="Ivanka" gender="Female" />
<Employee key="3" fullName="Kushner" gender="Male" />
```

WHAT DOES KEY ATTRIBUTE DO?

What does this do?



```
render() {
  return this.state.users.map(u => <ProfileCard data={u} key={u.id} />);
}
```

```
const employees
= [
{ empId: 1, fullName:
"Jimmy Kushner", jobTitle: "CEO" },
{ empId: 2, fullName:
 { empId: 3, fullName:
"Johnny Doe", jobTitle: "Vice
 President - Insider Trading" }
```

Jimmy Kushner

CEO

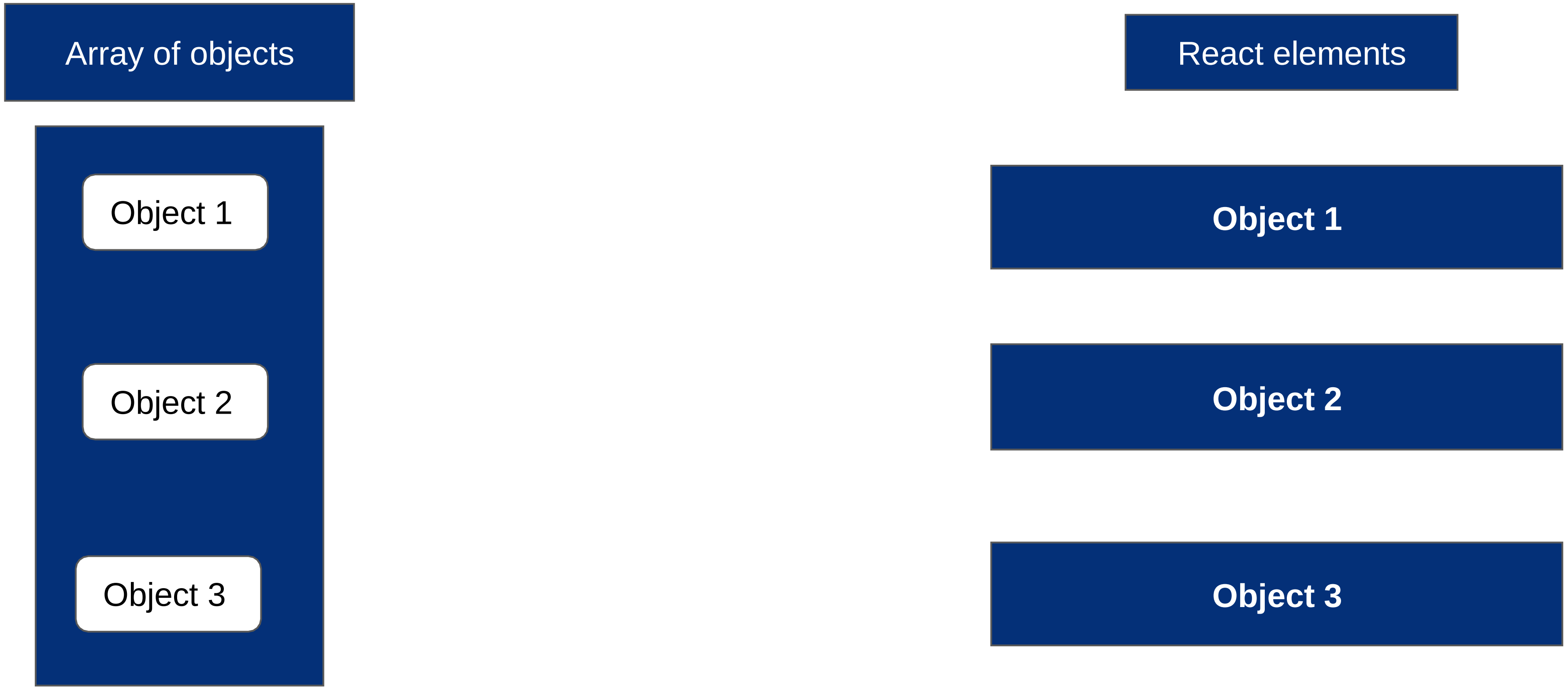
Richard Hammond

Vice President - Sales

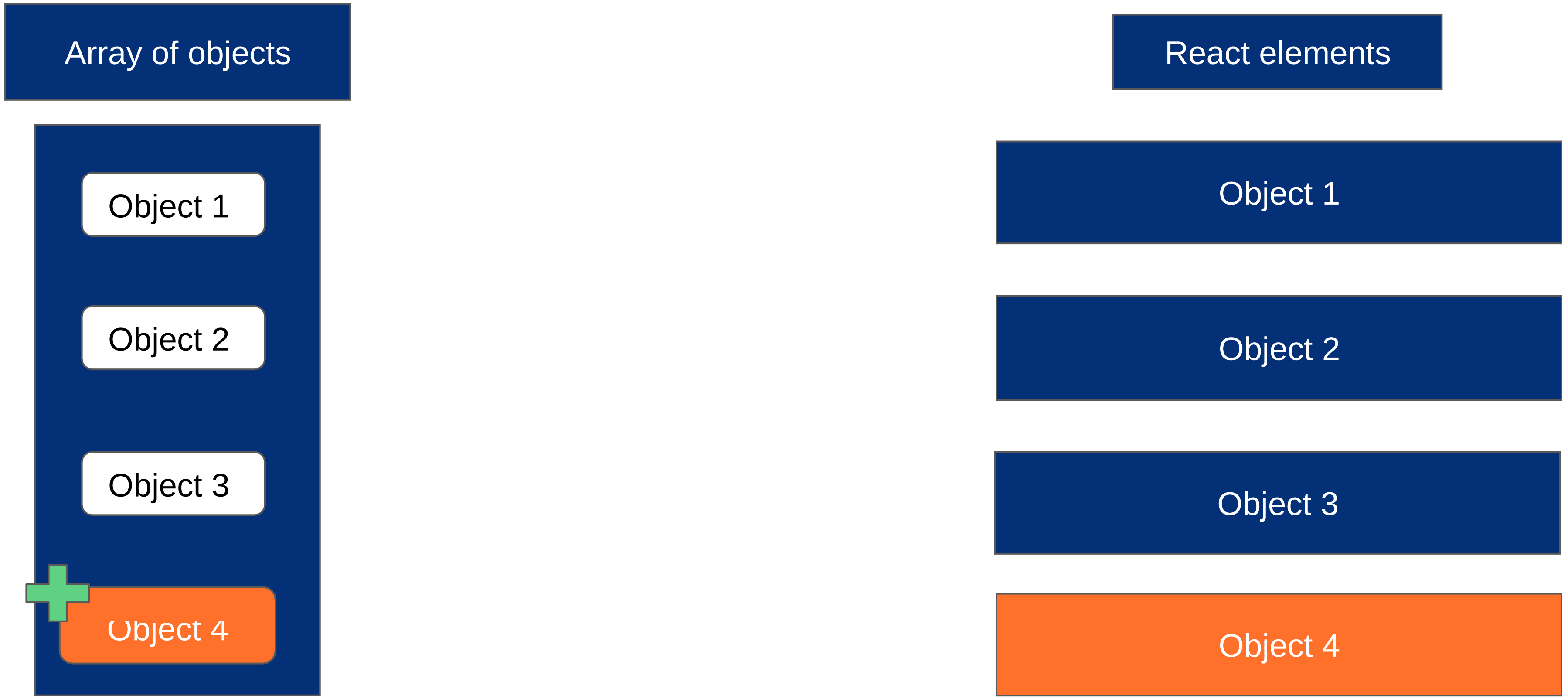
Johnny Doe

Vice President - Insider Trading

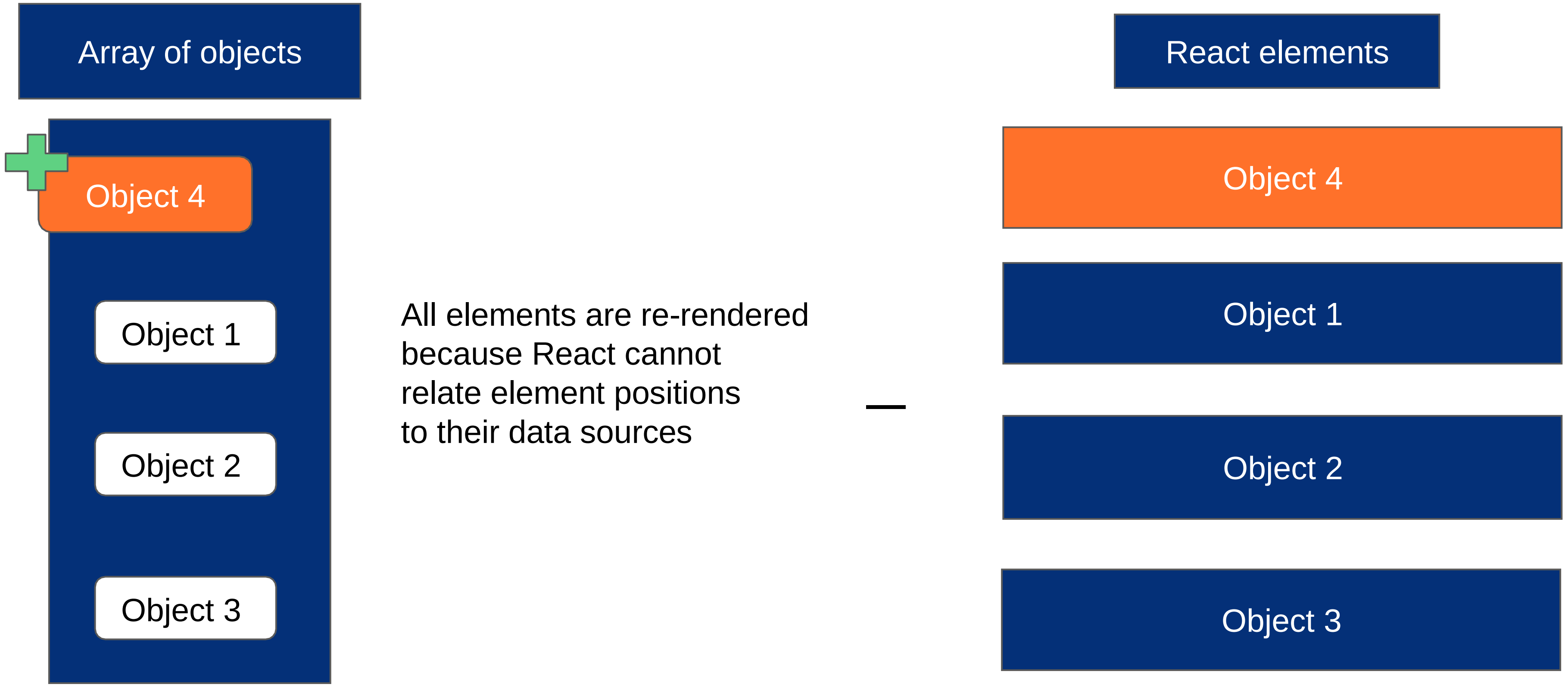
ARRAY OF OBJECTS TRANSLATE TO A LIST OF REACT ELEMENTS



ADDING ELEMENT AT THE END OF THE LIST IS EASILY HANDLED

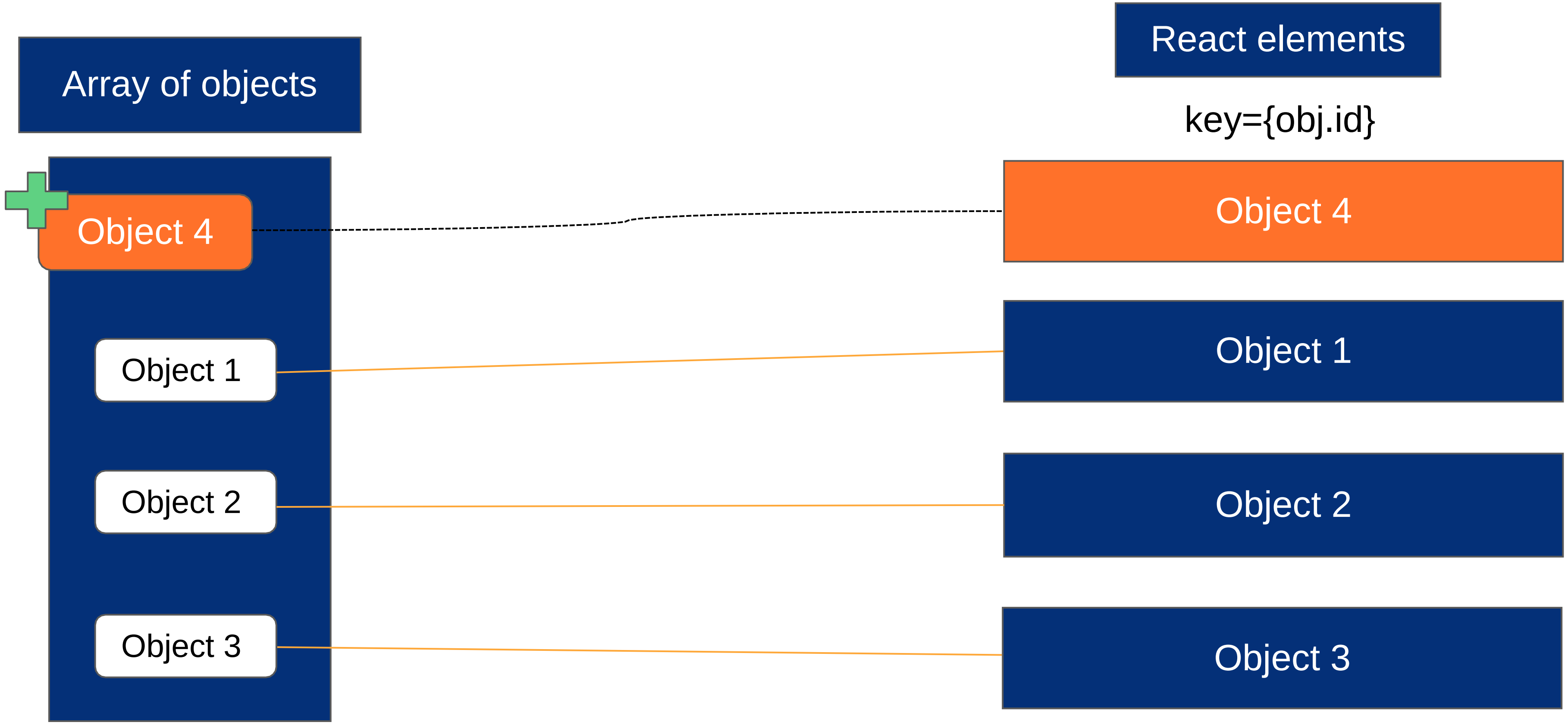


ADDING AN ITEM AT THE TOP CREATES A PROBLEM

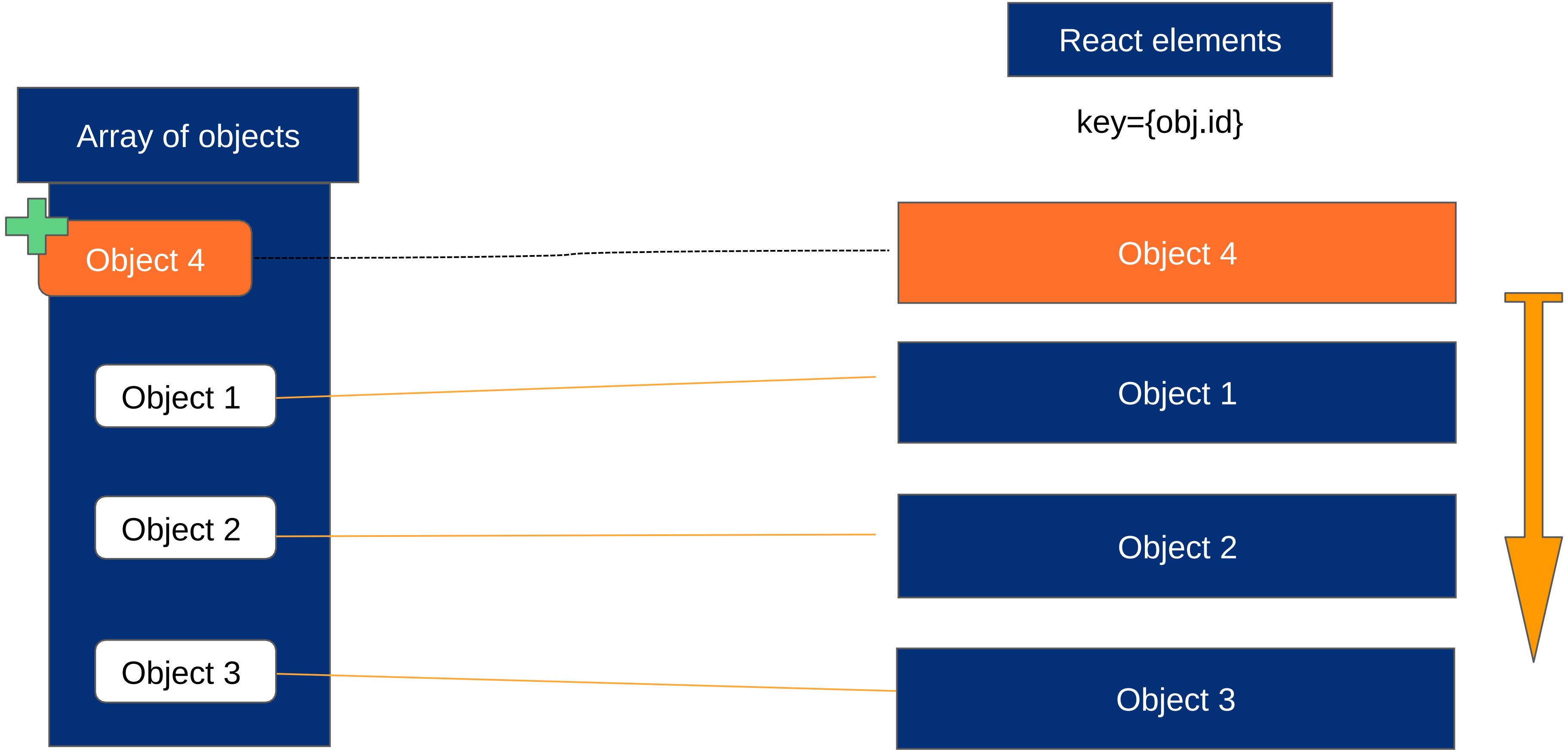


All elements are re-rendered because React cannot relate element positions to their data sources

CORRELATION BETWEEN DATA & ELEMENTS USING KEYS



EXISTING ELEMENTS ARE REPOSITIONED & A NEW ONE ADDED



SYNTAX

```
employees.map(e => <Employee name={e.fullName} title={e.jobTitle} key={e.empId})
```

- Should be a String
- Should be unique

Hands-On

Hands-On

Never use the element’s index value as the key because it isn’t consistent and cannot guarantee the position of the element.

USE THE INDEX VALUE

You can use the index value if your data source for the list is static and won't update over time. For instance, displaying images & captions from a gallery that won't update in real-time.

```
employees.map((e, index) => <Employee name={e.fullName} title={e.jobTitle} key={index})
```

SUMMARY

- Dynamic Lists → Always use a **unique String** as key
- When rendering lists "key" attribute helps React perform optimal updates

Rendering Lists

Using Fragments

RENDERING LISTS OF REACT COMPONENTS

```
const employees = [
  {
            empId: 1,
            fullName: "Jimmy Kushner",
            jobTitle: "CEO"
  }, {
            empId: 2,
            fullName: "Richard Hammond",
            jobTitle: "Vice President - Sales"
  }, {
            empId: 3,
            fullName: "Johnny Doe",
            jobTitle: "Vice President - Insider
              Trading"
  }
]
```

```
employees.map(e =>
            <Employee

name={e.fullName}
            title={e.jobTitle}

key={e.empId})
```

Jimmy Kushner
CEO

Richard Hammond
Vice President - Sales

Johnny Doe
Vice President – Insider
Trading

RENDER ONE ROOT ELEMENT

```
render() {
  return <Component />
}
```

You can only render ONE root element in a Component

RENDERING MULTIPLE TOP-LEVEL ELEMENTS IS NOT ALLOWED

```
render() {
 return (



   );
}
```

This isn't allowed

WRAPPING MULTIPLE INSTANCES OF ELEMENTS

```
render() {
  return (
    <div>
      <ComponentOne />
      <ComponentTwo />
      <ComponentThree />
    </div>
  );
}
```

So, we end up rendering an enclosing element such as a <div>

WRAPPING MULTIPLE INSTANCES OF ELEMENTS

```
render() {
  return (
   <div>
     <ComponentOne />
     <ComponentTwo />
     <ComponentThree />
   </div>
  );
}
```

So, we end up rendering an enclosing element such as a <div>

- The wrapper acts as the root and this works
- Good if you're using the wrapper for styling

RENDERING A LIST MAY INTRODUCE A LARGE NUMBER OF WRAPPERS

```
<div>
  <ComponentOne />
  <ComponentTwo />
  <ComponentThree />
</div>
<div>
  <ComponentOne />
  <ComponentTwo />
  <ComponentThree />
</div>
<div>
  <ComponentOne />
  <ComponentTwo />
  <ComponentThree />
</div>
<div>
  <ComponentOne />
  <ComponentTwo />
  <ComponentThree />
</div>
```

Fragments save you from rendering wrapper DOM nodes

```
render() {
  return (
    <div>
      <ComponentOne />
      <ComponentTwo />
      <ComponentThree />
    </div>
  );
}
```

```
import React, {Fragment} from "react";
render() {
  return (
    <Fragment>
      <ComponentOne />
      <ComponentTwo />
      <ComponentThree />
    </Fragment>
  );
}
```

THE IMPORTACE OF THE FRAGMENTS OPERATOR

```
render() {
  return (
    <>
      <ComponentOne />
      <ComponentTwo />
      <ComponentThree />
    </>
  );
}
```

The Fragments operator doesn’t render any extra wrapper in the DOM.

SHORT-HAND SYNTAX

```
import React, {Fragment} from "react";

render() {
  return employees.map(emp => (
    <Fragment key={emp.empId}>
      <Name value={emp.fullName} />
      <Job value={emp.jobTitle} />
    </Fragment>
  ));
}
```

The short-hand Fragments operator <></> does not permit attributes/props. Use <Fragment> when the key attribute needs to be set.

Only the key attribute is permitted at this time!

If your wrapper needs to implement event listeners or any other attribute besides key, then you'll have to use a renderable node such as a <div> as the wrapper.

FEATURES OF FRAGMENTS

```
render() {
 return (
   <ComponentOne />
   <ComponentTwo />
   <ComponentThree />
  );
}
```

- Incredibly easy way to render multiple top-level nodes
- Doesn’t render itself
- Doesn’t tax the DOM

thank you!