

# 30 Days Of React: Mapping Arrays



LinkedIn



Follow @asabeneh

776

Author: [Asabeneh Yetayeh](#)

October, 2020

<< [Day 5](#) | [Day 7](#) >>

- [Mapping arrays](#)
  - [Mapping and rendering arrays](#)
    - [Mapping array of numbers](#)
    - [Mapping array of arrays](#)
    - [Mapping array of objects](#)
    - [Key in mapping arrays](#)
- [Exercises](#)
  - [Exercises: Level 1](#)
  - [Exercises: Level 2](#)
  - [Exercises: Level 3](#)

## Mapping arrays

An array is the most frequently used data structure to handle many kinds of problems. In React, we use `map` to modify an array to list of JSX by adding a certain HTML elements to each element of an array.

### Mapping and rendering arrays

Most of the time data is in the form of an array or an array of objects. To render this array or array of objects most of the time we modify the data using `map`. In the previous section, we have rendered the techs list using

a map method. In this section, we will see more examples.

In the following examples, you will see how we render an array of numbers, an array of strings, an array of countries and an array of skills on the browser.

```
import React from 'react'
import ReactDOM from 'react-dom'
const App = () => {
  return (
    <div className='container'>
      <div>
        <h1>Numbers List</h1>
        {[1, 2, 3, 4, 5]}
      </div>
    </div>
  )
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)
```

If you check the browser, you will see the numbers are attached together in one line. To avoid this, we modify the array and change the array elements to JSX element. See the example below, the array has been modified to a list of JSX elements.

## Mapping array of numbers

```
import React from 'react'
import ReactDOM from 'react-dom'

const Numbers = ({ numbers }) => {
  // modifying array to array of li JSX
  const list = numbers.map((number) => <li>{number}</li>)
  return list
}

// App component

const App = () => {
  const numbers = [1, 2, 3, 4, 5]

  return (
    <div className='container'>
      <div>
        <h1>Numbers List</h1>
        <ul>
          <Numbers numbers={numbers} />
        </ul>
      </div>
    </div>
  )
}
```

```

    )
  }

  const rootElement = document.getElementById('root')
  ReactDOM.render(<App />, rootElement)

```

## Mapping array of arrays

Let's see how to map array of arrays

```

import React from 'react'
import ReactDOM from 'react-dom'

const skills = [
  ['HTML', 10],
  ['CSS', 7],
  ['JavaScript', 9],
  ['React', 8],
]

// Skill Component
const Skill = ({ skill: [tech, level] }) => (
  <li>
    {tech} {level}
  </li>
)

// Skills Component
const Skills = ({ skills }) => {
  const skillsList = skills.map((skill) => <Skill skill={skill} />)
  console.log(skillsList)
  return <ul>{skillsList}</ul>
}

const App = () => {
  return (
    <div className='container'>
      <div>
        <h1>Skills Level</h1>
        <Skills skills={skills} />
      </div>
    </div>
  )
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

## Mapping array of objects

## Rendering array of objects

```
import React from 'react'
import ReactDOM from 'react-dom'

const countries = [
  { name: 'Finland', city: 'Helsinki' },
  { name: 'Sweden', city: 'Stockholm' },
  { name: 'Denmark', city: 'Copenhagen' },
  { name: 'Norway', city: 'Oslo' },
  { name: 'Iceland', city: 'Reykjavík' },
]

// Country component
const Country = ({ country: { name, city } }) => {
  return (
    <div>
      <h1>{name}</h1>
      <small>{city}</small>
    </div>
  )
}

// countries component
const Countries = ({ countries }) => {
  const countryList = countries.map((country) => <Country country={country} />)
  return <div>{countryList}</div>
}

// App component
const App = () => (
  <div className='container'>
    <div>
      <h1>Countries List</h1>
      <Countries countries={countries} />
    </div>
  </div>
)

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)
```

## Key in mapping arrays

Keys help React to identify items which have changed, added, or removed. Keys should be given to the elements inside the array to give the elements a stable identity. The key should be unique. Mostly data will come with as an id and we can use id as key. If we do not pass key to React during mapping it raises a warning on the browser. If the data does not have an id we have to find a way to create a unique identifier for each element when we map it. See the following example:

```
import React from 'react'
import ReactDOM from 'react-dom'

const Numbers = ({ numbers }) => {
  // modifying array to array of li JSX
  const list = numbers.map((num) => <li key={num}>{num}</li>)
  return list
}

const App = () => {
  const numbers = [1, 2, 3, 4, 5]

  return (
    <div className='container'>
      <div>
        <h1>Numbers List</h1>
        <ul>
          <Numbers numbers={numbers} />
        </ul>
      </div>
    </div>
  )
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)
```

Let's also add in key in countries mapping example.

```
import React from 'react'
import ReactDOM from 'react-dom'

const countries = [
  { name: 'Finland', city: 'Helsinki' },
  { name: 'Sweden', city: 'Stockholm' },
  { name: 'Denmark', city: 'Copenhagen' },
  { name: 'Norway', city: 'Oslo' },
  { name: 'Iceland', city: 'Reykjavík' },
]

// Country component
const Country = ({ country: { name, city } }) => {
  return (
    <div>
      <h1>{name}</h1>
      <small>{city}</small>
    </div>
  )
}

// countries component
```

```
const Countries = ({ countries }) => {
  const countryList = countries.map((country) => (
    <Country key={country.name} country={country} />
  ))
  return <div>{countryList}</div>
}

const App = () => (
  <div className='container'>
    <div>
      <h1>Countries List</h1>
      <Countries countries={countries} />
    </div>
  </div>
)

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)
```

## Exercises

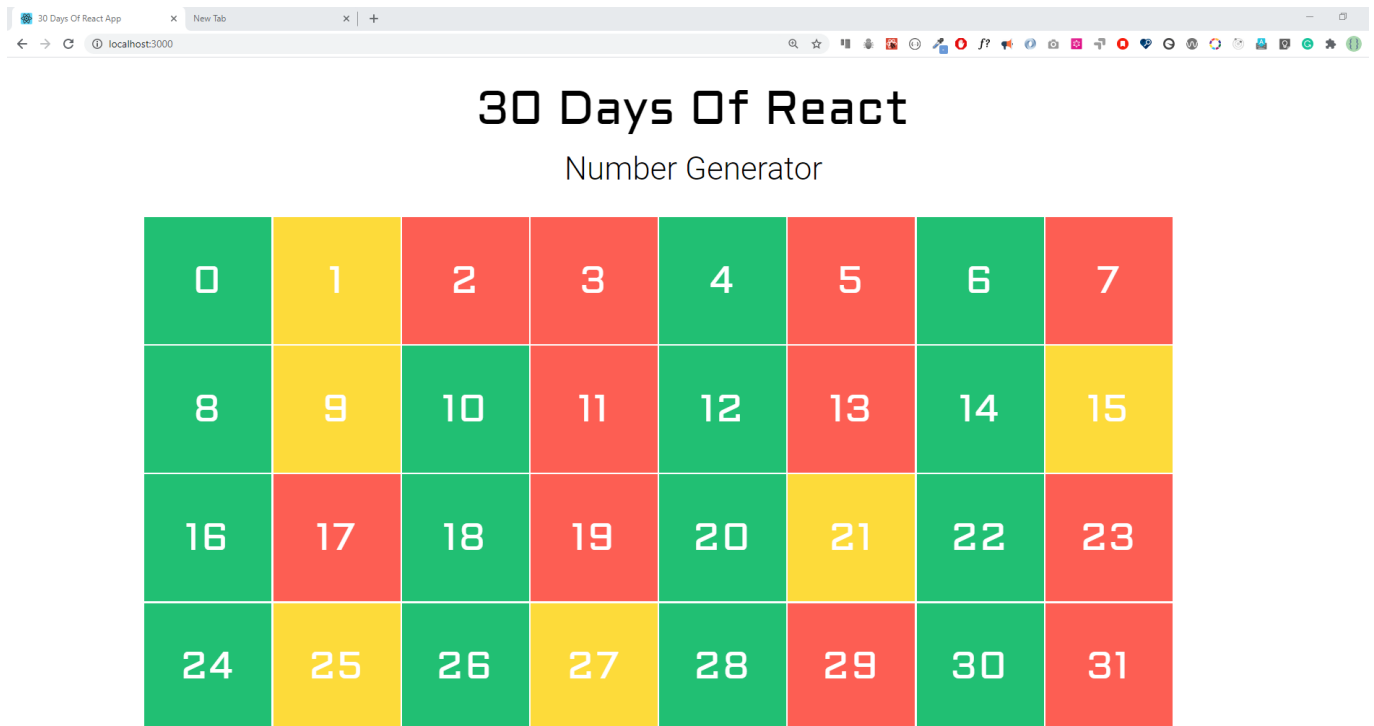
---

### Exercises: Level 1

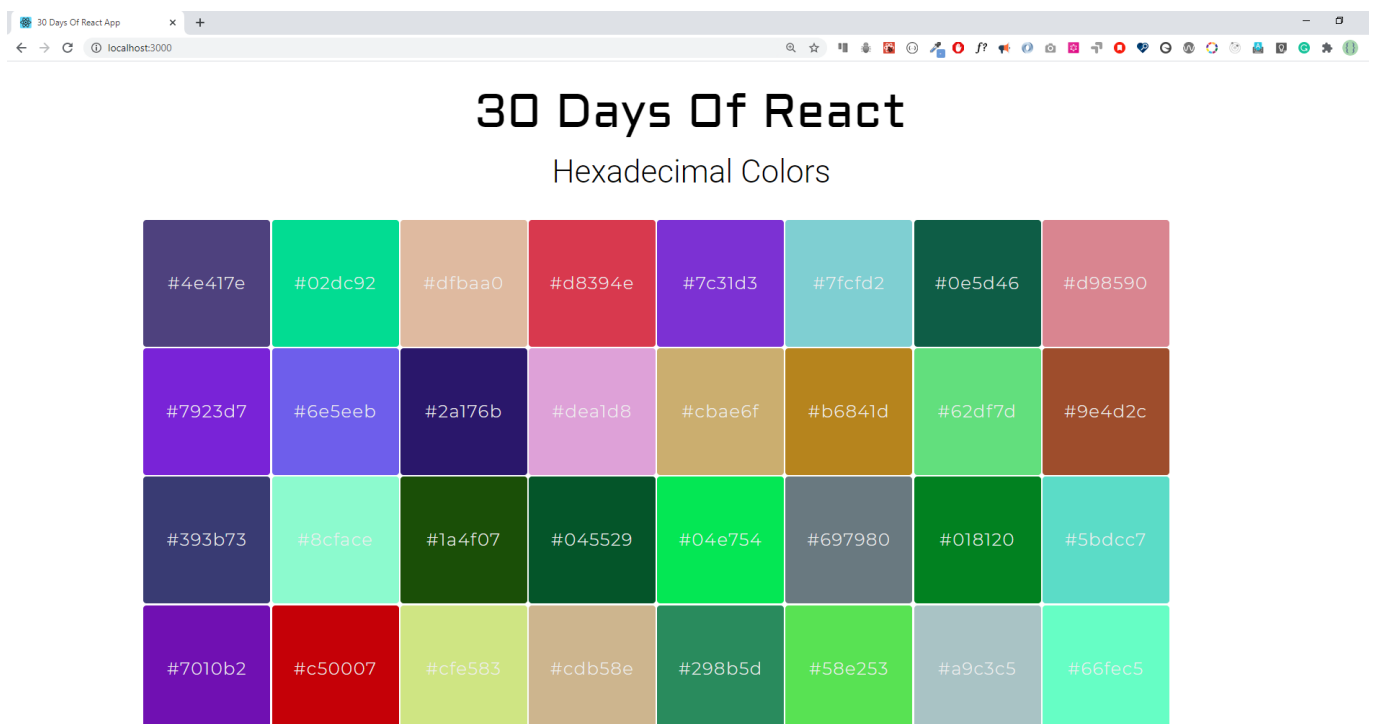
1. Why you need to map an array ?
2. Why we need keys during mapping an array ?
3. What is the importance of destructuring your code ?
4. Does destructuring make your code clean and easy to read ?

### Exercises: Level 2

1. In the following design, evens are green, odds are yellow and prime numbers are red. Build the following colors using React component

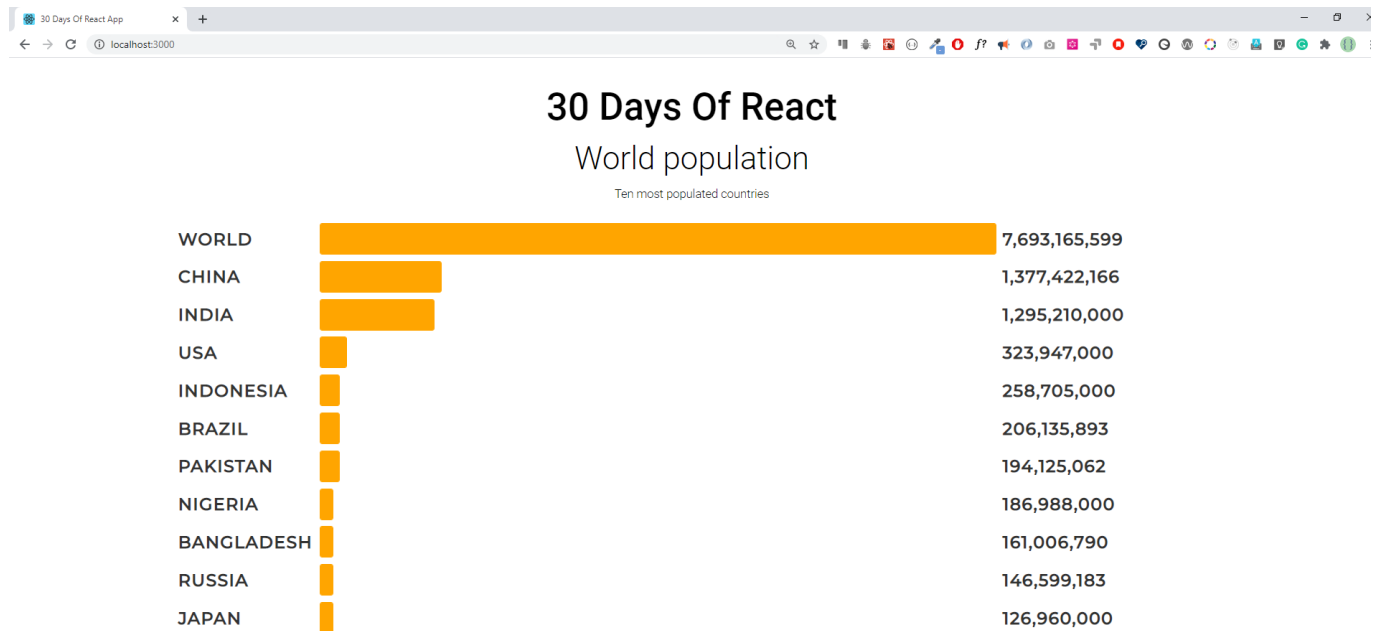


2. Create the following hexadecimal colors using React component



## Exercises: Level 3

1. Make the following bar group using the given [data](#)



🎉 CONGRATULATIONS ! 🎉

[<< Day 5](#) | [Day 7 >>](#)