



React.1

▼ What is React?

React is a JavaScript library created and (mostly) maintained by Facebook.

React is a JavaScript library for building User Interfaces.

▼ Define React

React is a JavaScript library that is only responsible for the view layer of an application.

This means it is only responsible for rendering a User Interface and updating the UI whenever it changes.

▼ Is React a framework?

No, it's a library

▼ What is the difference between a library and a framework?

The difference between a library and a framework is that a library only helps you in one aspect whereas a framework helps you in many aspects.

- React is a library because it only takes care of your UI.
- Angular, on the other hand, is a framework because it handles much more than the UI (It handles Dependency Injection, CSS encapsulation, etc.)

▼ Every JavaScript file is a...

standalone module, meaning variables, functions, and imports in one file/module do not affect other files/modules

▼ What is a bare import?

an import without a file path

e.g. `import React from "react";`

▼ What does a code bundler do?

When running a code bundler such as webpack or Parcel (or create-react-app which uses Webpack), then bare imports are resolved from your node_modules folder.

It will look for

```
node_modules/react/
```

to figure out how to import that package.

▼ What is the current React version?

17.0.2

▼ Do you need to import React for version 17?

No

▼ What is the current JavaScript or ECMAScript version?

ES11

JavaScript 2020

▼ What is the size of a React import?

approximately 12.5KB.

▼ What is document.createElement(tagName)?

a Web API provided by browsers that let you programmatically create an HTML element

```
const element = document.createElement("h2");

// this creates: <h2></h2>
// which you can then insert in the DOM:
document.body.appendChild(element);
```

▼ What does console.dir(element) show?

a method provided by browsers to list all the properties of a certain object

```
console.dir(lesson) // given element with variable lesson

div#lesson
accessKey: ""
align: ""
ariaAtomic: null
ariaAutoComplete: null
ariaBusy: null
ariaChecked: null
...
```

▼ How can you change the style of an element with element.style?

```
element.style = "color: red; background-color: blue";
```

▼ How can you change the class of an element with element.className?

```
element.className = "name-of-class";

element.className = "container center"; // multiple classes
```

▼ Why don't use element.classList.add("name-of-class") with React?

in React you will need to avoid changing the DOM directly (for example with classList.add) and instead **define the elements that you'd like to render**

▼ In React you avoid changing the DOM directly and instead define what?
the elements that you'd like to render

▼ document.createElement vs. React.createElement

document.createElement returns a DOM element (for example a div or an h1)

React.createElement returns **an object that represents the DOM element**

▼ How to React.createElement(type, options, children)?

```
const element = React.createElement("h1");
//returns an object similar to this one:
{
  type: 'h1',
```

```
  props: {}  
}
```

- ▼ Why does `React.createElement` return an object rather than the DOM element?
because React operates a Virtual DOM

- ▼ What is a virtual DOM?

a representation of the UI is kept in memory and synced with the DOM

this is a performance optimization

- ▼ How to change the class/style of `React.createElement`?

```
React.createElement("h1", {className: "center", style: "color: red"})
```

- ▼ How to write text for a `React.createElement`?

Notice how we're saying that we don't want to set a `className` or `style` (or other options) so we pass `{}` as the second parameter (you could also pass `null`).

```
React.createElement("h1", {}, "Hello World")
```

- ▼ What is the 3rd parameter for `React.createElement`?

children (so it also accepts other Elements for later on)

- ▼ What is a React element?

an Element is a representation of what the smallest piece of a User Interface will look like

- ▼ What is the glue between React and the DOM?

ReactDOM

- ▼ What is ReactDOM?

the library that receives instructions from React and efficiently updates the DOM based on that Virtual DOM

- ▼ Why does the virtual DOM exist?

to figure out which parts of the UI need to be updated and then batch these changes together

▼ ReactDOM binds the idea of React to what?

a web browser (example: Firefox, Chrome, Safari, Edge, etc.)

▼ What is React Native?

the glue between React and native apps

▼ What is reconciliation?

React is creating the virtual representation of the UI in memory, and then ReactDOM receives that and syncs the UI to the actual DOM.

▼ How to install React and ReactDOM?

npm install react react-dom

▼ How to import React?

```
import React from 'react';
```

▼ How to import ReactDOM?

```
import ReactDOM from "react-dom";  
  
import {render} from "react-dom"; // render method
```

▼ Where should elements be rendered?

within a root element aka the container

```
<div id="root"></div>  
  
import {render} from "react-dom";  
  
const root = document.querySelector("#root");  
render(React.createElement("p", {}, "Hello World"), root);
```

The render(element, root) method expects the element to render as a first argument and the root as the second one.

▼ Apps built with React usually have how many root elements?

One. The whole application is rendered inside that root element.

▼ The root element is completely managed by what?

ReactDOM

▼ The JSX syntax looks similar to HTML but it is not what?

HTML

▼ JSX syntax

```
import React from "react";

// this code will be translated
const title = <h1>Hello World</h1>

// into this
const title = React.createElement("h1", {}, "Hello World");
```

▼ the JSX that you write is being converted to what?

React.createElement

▼ JSX requires what?

React. It creates JSX

you need to import React in every file that uses JSX or else it won't work

▼ You need to convert JSX with babel to what?

JavaScript

▼ JSX with attributes. Note className not class

```
const title = <h1 id="brand-title" className="primary-color">Supermarket</h1>;
```

▼ KEY: Since JSX is being converted into a React.createElement(...) that returns an object, you can essentially think of a JSX element as if it was what?

an object

As a reminder, this object is a representation of a piece of your UI that React maintains in its Virtual DOM. This lets it efficiently know when it updates and re-render it using ReactDOM.

```
// given:
<h1 className="title">Supermarket</h1>

// we can think of it as an object like this:
{
  type: 'h1',
  props: {
    className: "title",
    children: "Supermarket"
  }
}
```



