the shortcut

Fullstack Web Development Tutorial Lesson 20

Today's lesson will cover

- **Object properties**
- **JavaScript Recap for React**
  - React and JavaScript Classes
  - Arrow Functions in React
  - Functions as Components in React
  - React Class Component Syntax
  - Template Literals in React
  - var, let, and const in React
  - map, reduce and filter
  - Ternary Operator in React

JavaScript fundamentals

Object property flags and descriptors

- Object properties, besides a **`value`**, have three special attributes (so-called “flags”):
  - **`writable`** – if `true`, the value can be changed, otherwise it’s read-only.
  - **`enumerable`** – if `true`, then listed in loops, otherwise not listed.
  - **`configurable`** – if `true`, the property can be deleted and these attributes can be modified, otherwise not.
- Syntax
  - ```
    let descriptor = Object.getOwnPropertyDescriptor(obj, propertyName)
    ```
- You can make object properties non-writable, non-enumerable and non-configurable
- Object.defineProperties(obj, descriptors) allows to define many properties at once.
    ```

Property getters and setters

- There are two kinds of object properties.
  - The first kind is *data properties*. All properties that we've been using until now were data properties.
  - The second type of properties is something new. It's *accessor properties*. They are essentially functions that execute on getting and setting a value, but look like regular properties to an external code.
- Accessor properties are represented by "getter" and "setter" methods. In an object literal they are denoted by `get` and `set`.
- The getter works when `obj.propName` is read, the setter – when it is assigned
- For accessor properties, there is no `value` or `writable`, but instead there are `get` and `set` functions. That is, an accessor descriptor may have:
  - **`get`** – a function without arguments, that works when a property is read,
  - **`set`** – a function with one argument, that is called when the property is set,
  - **`enumerable`** – same as for data properties,
  - **`configurable`** – same as for data properties.
- **Smarter getters/setters**: Getters/setters can be used as wrappers over "real" property values to gain more control over operations with them.

JavaScript Recap for React

React and Javascript Classes

- Being confronted with a React class component, requires the prior knowledge about JavaScript classes.
- A class describes an **entity** which is used as a blueprint to create an **instance** of this entity. Once an instance of the class gets created with the new statement, the constructor of the class is called which instantiates the instance of the class.
- Usually classes are used for inheritance in object-oriented programming. They are used for the same in JavaScript whereas the extends statement can be used to inherit with one class from another class.
- A JavaScript class is used for defining a React component, but as you can see, the React component is only a "React component" because it inherits all the abilities from the actual React Component class which is imported from the React package
- That's why the render() method is mandatory in React class components: The React Component from the imported React package instructs you to use it for displaying something in the browser. Furthermore, without extending from the React Component, you wouldn't be able to use other lifecycle methods.

Arrow Functions

- JavaScript arrow functions are often used in React applications for keeping the code concise and readable.
- They are one of JavaScript's language additions in ES6 which pushed JavaScript forward in functional programming.

Functions as Components

- Function components are the preferred way of defining components in React. They have less boilerplate, add less complexity, and are simpler to maintain than React class components. You can easily migrate your class components to function components with React Hooks.
- JavaScript arrow functions are a great way of keeping your function components in React concise. Even more when there is no computation in between and thus the function body and return statement can be left out.

Template Literals

- Template literals are another JavaScript language specific feature that came with JavaScript ES6.
- You only have to use backticks and the `${}` notation for inserting JavaScript primitives. However, string literals are not only used for string interpolation, but also for multiline strings in JavaScript

var, let and const

- These may be obvious but if you started learning JavaScript before ES6, rules of thumb when to use which variable declaration:
    - don't use var anymore, because let and const are more specific
    - default to const, because it cannot be re-assigned or re-declared
    - use let when re-assigning the variable
- While let is usually used in a for loop for incrementing the iterator, const is normally used for keeping JavaScript variables unchanged. Even though it is possible to change the inner properties of objects and arrays when using const, the variable declaration shows the intent of keeping the variable unchanged though.

map, reduce and filter

- There is no React specific API such as a custom attribute on a HTML tag which enables you to render multiple items in React. You can use plain JavaScript for iterating over the list of items and returning HTML for each item.
- React developer becomes used to the built-in JavaScript `map()` methods for arrays. It just makes so much sense to map over an array and return the rendered output for each item. The same can be applied for custom tailored cases where `filter()` or `reduce()` make more sense rather than rendering an output for each mapped item.

Ternary and conditional operators

- You cannot use an if-else statement directly in JSX, but you can return early from the rendering function. Returning null is valid in React when displaying nothing.
- However, if you want to use an if-else statement within the returned JSX, you can do it by using a JavaScripts ternary operator.
- Another way of doing it, if you only return one side of the conditional rendering anyway, is using the && operator.

the shortcut

Self Study Assignments

To Dos

- Try to use Github pages, Netlify or Heroku to showcase live projects
- Create a game of Rock, Paper and Scissors using JS which works on console, or with interactive UI using HTML, CSS and JS however you prefer ***(If you are working on your own project where you are using JS already, feel free to ignore this task but please share the project update with Lena.)***
- Continue freecodecamp (FCC) Javascript. Ideally finish before we resume after summer.
- Continue with FCC HTML, CSS lessons. Ideally finish all the lessons by end of this month.
- If you believe FCC exercises aren't the best for you as in if you are quite advanced already, please start working on your own project and reach out to mentors for help if needed.