Section 3 Quiz - React Basics, Components, Props & JSX

1. Which kind of code do you write when using React.js?
    a. Definitive JSX Code
    b. Imperative JavaScript Code
    c. Declarative JavaScript Code

2. What is "JSX"?
    a. It's a standard JavaScript syntax
    b. It's a special, non-standard syntax which is enabled in React projects
    c. It's a special string which you can pass to React functions

3. Why is React all about "Components"?
    a. Every UI in the end up is made up of multiple building blocks (= components), hence it makes sense to think about user interfaces as "combinaations of components"
    b. React projects are configured to only work with components, hence you have to use them when writing React code.
    c. Components offer better performance than "standard user interfaces" that don't use components.

4. What does "declarative" mean?
    a. "Declarative" is the same as "imperative"
    b. You define the individual steps that need to be taken to achieve a desired outcome (e.g. a target UI).
    c. You define the desired outcome (e.g. a target UI) and let the library (React) figure out the steps.

5. What is a "React Component"?
    a. It's a JavaScript function which typically returns HTML (JSX) code that should be displayed.
    b. It's a replacement for standard HTML which is supported by modern browsers.
    c. It's a JavaScript function that must not return anything.

6. How many custom React components must a React app have?
    a. At least 2
    b. At most 99
    c. That’s totally up to you

7. Which statement is correct?
    a. With React, you build multiple sibling component trees that are then mounted into the same DOM node.
    b. With React, you build a component tree with one root component that's mounted into a DOM node.

c. With React, you always mount every component into it's own DOM node.

8. What does "component tree" mean?
    a. **It means that you have a root node which then has more components nested benaeath it.**
    b. It means that you must always return more than one component or HTML element per component function.
    c. It means that you can build multiple components.

9. How do you pass data between components?
    a. Via global JavaScript variables that are accessible in all files
    b. **Via "custom HTML attributes" (better known as "props")**
    c. Via standard HTML attributes wich you can use in non-React apps as well

10. How can you output dynamic data in React components (i.e. in the returned JSX code)?
    a. **You can use single curly braces (opening & closing) with any JS expression between them.**
    b. React has a special syntax that allows you to output variable values (i.e. values stored in variables) and nothing else: Opening & closing curly braces
    c. You can’t