

Author: Alexander Adu-Sarkodie

LinkedIn: <https://www.linkedin.com/in/alexander-adu-sarkodie-063b4b4a/>

GitHub: <https://github.com/kukuu?tab=repositories>

Blog: <http://www.goldhawk-college.com/istudio/tech-delivery.html>

Question	Answer
What are the differences between the apply, bind and call methods of a function?	<p>Call runs the function in the context of the first argument, and subsequent arguments are passed in to the function to work with.</p> <p>Unfortunately, this is not good enough when you have multiple arguments to deal with. Its limitation becomes apparent when you want to write code that shouldn't know the number of arguments that the function needs.</p> <p>With apply the second argument needs to be an array which is unpacked into arguments which are passed to the function.</p> <p>The bind method enables you to pass arguments to a function without invoking it. Instead, it returns a new function with the arguments bound preceding any further arguments.</p>
Are there alternative syntaxes for them?	You can use the arrow function provided by es6, and also The filter method. It creates a new array with all elements that pass the test implemented by the provided function.
How do you handle errors in promises and async/await expressions?	<p>For best practices it is always useful to execute await in async calls. Using await makes sense even if an async function doesn't return anything.</p> <p>It's Promise is simply used as a signal for telling the caller that it is finished. However, if you just want to trigger a series of computation, you may not need await until after the process has completed.</p>
What happens if you don't?	<p>It can trickle a non-controlling pollution effect. The await guarantees that a previous step in the chain is completely finished before the remainder of the process is executed.</p> <p>On the other hand Promises allows you to connect legacy codes that cannot directly work with async function. By using Promises directly.</p>

<p>In what do arrow functions differ from normal functions? How do you bind one?</p>	<p>An arrow function expression has a shorter syntax than a function expression and does not bind its own <code>this</code>, <code>arguments</code>, <code>super</code>, or <code>new.target</code>.</p> <p>Arrow functions can return undefined as they can be bound to windows when working with say arrays and cannot use the <code>this</code> property. The focus becomes bound to the window instead of context.</p> <p>To resolve this will be to use the function expression or short hand syntax.</p>
<p>Why is $0.1 + 0.2 !== 0.3$?</p>	<p>This is because adding floating points is only an approximation. Best to use double precision.</p>
<p>How do you get the correct result?</p>	<p>You can obtain the correct result if you store the number in a different way. As humans we think in decimal, while computers think in binary. Computers normally don't prefer dealing in multiples of ten bits, but rather prefer eight.</p>
<p>Which CSS rules do you use for animations, if you care about performance?</p>	<p>Limiting to opacity and transforms will help boost performance.</p> <p>Also using <code>will-change</code> to ensure the browser knows that you intend to change an element's property. This allows the browser to put the most appropriate optimisations in place ahead of when you make the change.</p>
<p>What problems does the box-sizing rule solve?</p>	<p>The box sizing rule controls how the box model rule is handled for elements it applies to. For example, it allows you to retain the literal width of an element you set without having to perform any additional calculations and manage the complexity of widths that come from multiple properties.</p>
<p>What problems does CORS cause? How do you solve them?</p>	<p>CORS (Cross Origin Resource Sharing) allows the use of Web applications within browsers when domains aren't the same. To resolve them will be to use additional header parameters which may be missing in responses.</p> <p>Servers also need to be configured properly. API clouds provide comprehensive platforms to create and deploy RESTful services.</p>

How do you optimize the performance of a React component?	<ol style="list-style-type: none"> 1. By using state managing techniques including with REDUX. 2. Use life cycle methods properly: shouldComponentUpdate. 3. Allowing rendering and re-rendering only when state changes. 4. Make child components smarter and let them update themselves instead of having all elements respond to. 5. Target component should subscribe to store and track changes . 6. Also, no need to update app component if the list and order are the same.
In which component lifecycle method do you make load-time API calls?	componentDidMount(). Doing componentDidMount in a load-time API call will guarantee that there's a component to update.
How do you render a really long list in a performant way?	<ol style="list-style-type: none"> 1. Use caching and memorisation of all format instances. Especially when iterating through a loop. 2. By adding content dynamically as it is scrolled into view . Sort of lazy loading. 4. Use pagination . When an item changes in a page, React only has to call the main render function that iterates on the page, and only call the render function from a single page, which make a lot less iteration work. 5. Use observables. As the application will no longer require the parent component that holds the list to be re-rendered. 6. It might also help you in re-rendering the list faster as your component could just return false when the props change on shouldComponentUpdate. <p>As REACT does more than simple string templating , there will likely be additional overhead like setting up and maintaining references, stores and caches so that updates can be fast.</p>

How do you handle state?	<p>1. Using life cycle methods. For example you can handle states using <code>setState</code>.</p> <p>2. Also, it is bad practice to keep state calculated from props because you will have to update it with <code>componentWillReceiveProps()</code>, leading to duplication of source of truth. The same applies when you hold state based on props calculation. For example calculating say full name from name and surname.</p> <p>3. Don't hold state that you don't use for rendering. Leads to unneeded re-renders and other inconsistencies.</p>
When and where is it safe to use local state and when is it better to use a state management library?	<p>Local state is used when a state value changes and triggers a re-render. This state can then be passed down to children as props, allowing the separation of components between smart data-components (logic) and dumb ones (presentational components).</p> <p>In very complicated architectures it is preferred to use state management libraries like REDUX which uses global store. This store lives at the highest level of the application and passes data down to all children. For example, you can achieve this by using a connect wrapper and a <code>mapStateToProps</code> function.</p>