<https://facebook.github.io/react-native/docs/getting-started> docs  
<https://docs.nativebase.io/Components.html> native base  
<https://www.tutorialspoint.com/react_native/>  
<https://oblador.github.io/react-native-vector-icons/> vector icon  
<https://www.fastfwd.com/custom-component-in-react-native/> custom components  
<https://github.com/react-native-community/react-native-device-info> can get any info about user device  
<https://github.com/react-native-community/react-native-maps> map docs  
<https://www.youtube.com/watch?v=4qq0GQPkfjI&t=428s> map lecture  
<https://www.youtube.com/watch?v=Hn2acItzQBk> redux lecture

1. React native life sycle   
   A component’s lifecycle can be divided into 4 parts:

* Mounting —  an instance of a component is being created and inserted into the DOM.
* Updating — when the React component is born in the browser and grows by receiving new updates.
* Unmounting — the component is not needed and gets unmounted.
* Error handling — called when there is an error during rendering, in a lifecycle method, or in the constructor of any child component.  
  <https://www.netguru.com/codestories/react-native-lifecycle>  
  https://medium.com/@amanpreet\_singh/lifecycle-of-react-native-component-2020-edition-1cec4762859b

1. Promises vs await
2. Interceptors .  
   **Interceptors** are a feature that allows an application to *intercept* requests or responses before they are handled by the .then() or the .catch().

https://blog.bitsrc.io/setting-up-axios-interceptors-for-all-http-calls-in-an-application-71bc2c636e4e

Interceptors are used to   
<https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Intercept_HTTP_requests>

1. Back button off
2. React components
3. Deep linking
4. OId update
5. Event loop in java script
6. React native config
7. Pagination
8. Require vs import and npm:  
   Import and Export are ES6 features(Next gen JS).  
   Require is old school method of importing code from other files

Major difference is in require, entire JS file is called or imported. Even if you don't need some part of it.

var myObject = require('./otherFile.js'); //This JS file will be imported fully.

Whereas in import you can extract only objects/functions/variables which are required.

import { getDate }from './utils.js';

//Here I am only pulling getDate method from the file instead of importing full file

Another major difference is you can use require anywhere in the program where as import should always be at the top of file

<https://www.educba.com/require-vs-import/>

1. How use just one loader for all screes: Ans: Will on App.js or root
2. If two reducers then which loader will called
3. When function called then how functions pass value into params by deep copy or shallow copy
4. Purpose of default in redux.
5. If we do not initialize values into store then redux work fine or not?
6. Compare two json objects.
7. If we access particular key in object but this key does not exist then what’s response
8. How to accurate map directions.
9. How to pass tokens into fetch API
10. query params vs path params.
11. Difference between fetch and xios  
    https://blog.logrocket.com/axios-or-fetch-api/#:~:text=To%20send%20data%2C%20fetch(),an%20argument%20to%20fetch()%20.  
    difference between git pull and git fetch  
    nativagate vs push  
    navigation.goback and navigation.pop  
    buttom