



# Windows OS



## Setting, Build or Running the App



### Running Android Emulator Instead of iOS

- Do **NOT** use `npx react-native run-ios`
- Instead use `npx react-native run-android`


### Pod Installation for iOS on Windows

- Do **NOT** install pods
  - During the course, if you observe me executing commands like `cd ios` followed by `pod install`, please do **NOT** replicate this action on Windows.



### Gradlew Command on Windows

- on macOS some of the commands that are executed inside the `android` folder start with `./gradlew`
  - Examples - `./gradlew clean` `./gradlew build`
- Do **NOT** use `./gradlew` on Windows, instead use `gradlew`

### Custom Fonts and Windows

- **Different Font Management Systems on macOS & Windows**
  -  Each operating system manages fonts in its own way. Windows and macOS have different ways of registering and referencing font weights and styles, which might be affecting how React Native interprets them.
  - The videos in this course are going to be edited soon to account for this, however, for students taking the course now - this is a resource on how to resolve your issues.

- **Font Files**

-  Some font families, including "Inter", come with separate font files for each weight or style (e.g., Inter-Bold.ttf, Inter-Regular.ttf). When you specify a `fontWeight`, React Native will try to load the appropriate font file. If the platform doesn't handle it correctly or if there's an issue with the font file naming or registration, it won't be applied as expected which is the case on **Windows**.
-  **Windows** does not recognize `fontFamily: 'Inter'` directly without specifying the variation of it such as `fontFamily: 'Inter-Bold'`

Given these factors, here are a few suggestions and workarounds:

- **Use Specific Font Variations**

- Using specific font variations like "Inter-Bold" can work.
- Here's a breakdown chart of which font to use when we're specifying font weight's and fonts in windows instead of macOS.

Inter Font Variations	Corresponding Font Weight
Inter-Thin	'100'
Inter-ExtraLight	'200'
Inter-Light	'300'
<b>Inter-Regular</b>	<b>'400' (Default Value of <code>fontWeight</code> in React Native)</b>
Inter-Medium	'500'
Inter-SemiBold	'600'
Inter-Bold	'700'
Inter-ExtraBold	'800'
Inter-Black	'900'

- Better solution if you're up for creating it into your project
  - inside `src/assets/fonts` folder:
    - Create the fonts folder if it doesn't exist already
    - Create `helper.js` file

```

/**
 * Returns the appropriate font family based on the weight.
 * @param {string} baseFont - The desired font in case you're using something
                             different in your later projects
 * @param {string} weight - The desired font weight(e.g., 'normal', 'bold', '600')
 * @returns {string} - The appropriate font family for the platform.
 */
export const getFontFamily = (baseFont='Inter', weight) => {
  switch (weight) {
    case '100':
      return `${baseFont}-Thin`;
    case '200':
      return `${baseFont}-ExtraLight`;
    case '300':
      return `${baseFont}-Light`;
    case 'normal':
    case '400':
      return `${baseFont}-Regular`;
    case '500':
      return `${baseFont}-Medium`;
    case '600':
      return `${baseFont}-SemiBold`;
    case 'bold':
    case '700':
      return `${baseFont}-Bold`;
    case '800':
      return `${baseFont}-ExtraBold`;
    case '900':
      return `${baseFont}-Black`;
    default: // Default to regular if the weight is not recognized
      return `${baseFont}-Regular`;
  }
}

// Usage example in a stylesheet file
import {StyleSheet} from 'react-native';
//import getFontFamily in your style.js file

const style = StyleSheet.create({
  text: {
    fontFamily: getFontFamily('bold'),
    fontSize: 16, //use whichever font size fits you after
  }
});

```