View Binding

set the `viewBinding` build option to `true` in the module-level `build.gradle` file, as shown in the following example:

```
android {
    ...
    buildFeatures {
        viewBinding true
    }
}
```

If you want a layout file to be ignored while generating binding classes, add the `tools:viewBindingIgnore="true"` attribute to the root view of that layout file:

```
    <LinearLayout
        ...
        tools:viewBindingIgnore="true" >
        ...
    </LinearLayout>
```

a binding class is generated for each XML layout file that the module contains. Each binding class contains references to the root view and all views that have an ID. The name of the binding class is generated by converting the name of the XML file to camel case and adding the word "Binding" to the end.

this is a pre made function you can easily just call it onCreate fun of your activity

```
private lateinit var binding: ActivityMainBinding
private fun initializeBinding() {
        Log.d("MAIN", "Binding Initialized!")
        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)
    }
```

you can manipulate elements immediately, to call the UI elements you just use:

```
binding.tvMain //change tvMain to your UI element ID
```

another example:

```
val loc = binding.etLocation.text.toString()
```