Practical class # 7 – Tip Calculator – Part 2 – Compute the tip

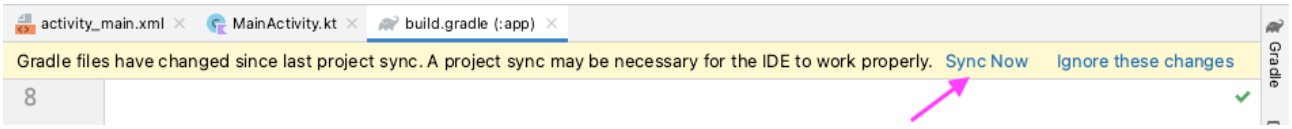
With many Views, the findViewById() can be hard to use. Android Studio adopts View Binding as a method to simplify the reference to views.

1) Enable View Binding

1. Open the app's build.gradle file (Gradle Scripts > build.gradle (Module: Tip_Time.app))
2. In the android section, add the following lines:

```
buildFeatures {
        viewBinding = true
}
```

3. Note the message Gradle files have changed since last project sync.
4. Press Sync Now.



5. After a few moments, you should see a message at the bottom of the Android Studio window, Gradle sync finished. You can close the build.gradle file if you want.

2) Initialize the binding object

1. Open MainActivity.kt (app > java > com.example.tiptime > MainActivity).
2. Replace all of the existing code for MainActivity class with this code to setup the MainActivity to use view binding:

```
class MainActivity : AppCompatActivity() {

        lateinit var binding: ActivityMainBinding

        override fun onCreate(savedInstanceState: Bundle?) {
                super.onCreate(savedInstanceState)
                binding = ActivityMainBinding.inflate(layoutInflater)
                setContentView(binding.root)
        }
}
```

3. This line declares a top-level variable in the class for the binding object. It's defined at this level because it will be used across multiple methods in MainActivity class.

```
lateinit var binding: ActivityMainBinding
```

4. The lateinit keyword is something new. It's a promise that your code will initialize the variable before using it. If you don't, your app will crash.
5. This line initializes the binding object which you'll use to access Views in the activity_main.xml layout.

```
binding = ActivityMainBinding.inflate(layoutInflater)
```

6. Set the content view of the activity. Instead of passing the resource ID of the layout, R.layout.activity_main, this specifies the root of the hierarchy of views in your app, binding.root.

```
setContentView(binding.root)
```

REMARK: different ways to binding objects

```
// Old way with findViewById()
val myButton: Button = findViewById(R.id.my_button)
myButton.text = "A button"

// Better way with view binding
val myButton: Button = binding.myButton
myButton.text = "A button"

// Best way with view binding and no extra variable
binding.myButton.text = "A button"
```

3) Calculate the tip

The first step is to add a click listener to specify what the **Calculate** button should do when the user taps it.

```
binding.calculateButton.setOnClickListener{ calculateTip() }
```

Inside the MainActivity, but outside the onCreate method, declare the function calculateTip().

1) Retrieve cost of service

```
fun calculateTip() {
        val stringInTextField = binding.costOfService.text.toString()
        val cost = stringInTextField.toDouble()
}
```

2) Get the percentage

```
val selectedId = binding.tipOptions.checkedRadioButtonId
val tipPercentage = when (selectedId) {
        R.id.option_twenty_percent -> 0.20
        R.id.option_eighteen_percent -> 0.18
        else -> 0.15
}
```

3) Calculate and check if it should be rounded up

```
val roundUp = binding.roundUpSwitch.isChecked

if (roundUp) {
        tip = kotlin.math.ceil(tip)
}
```

4) Format the tip

```
val formattedTip = NumberFormat.getCurrencyInstance().format(tip)
```

choose hoose **NumberFormat (java.text) as import package**

4) Display the tip

1. Open strings.xml (app > res > values > strings.xml)
2. Change the tip_amount string from Tip Amount to Tip Amount: %s.
3. In the calculateTip inser the display of the tip

```
binding.tipResult.text = getString(R.string.tip_amount, formattedTip)
```

5) Use a placeholder

1. Open activity_main.xml (app > res > layout > activity_main.xml).
2. Find the tip_result TextView.
3. Remove the line with the android:text attribute.
4. Add a line for the tools:text attribute set to Tip Amount: $10.

6) Handle errors

```
val cost = stringInTextField.toDoubleOrNull()
if (cost == null) {
        binding.tipResult.text = ""
        return
}
```

1 cup **128 g** 4.5 oz