**Difference between Snackbar and Toast:**

**1. User Interaction**

**- Snackbar:**

- Example: A Snackbar can show an action button (e.g., “Undo”) that allows the user to reverse an action.

*```kotlin*

*val snackbar = Snackbar.make(findViewById(R.id.coordinatorLayout), "Item deleted", Snackbar.LENGTH\_LONG)*

*snackbar.setAction("UNDO") {*

*// Code to undo the delete action*

*}*

*snackbar*.show()

```

**- Toast:**

**- Example:** A Toast simply shows a message to the user without requiring any interaction.

```kotlin

Toast.makeText(applicationContext, "Message sent", Toast.LENGTH\_SHORT).show()

```

**2. Design**

**- Snackbar:**

**- Example:** Snackbars have a distinct design with a background and can incorporate action buttons.

```kotlin

val snackbar = Snackbar.make(findViewById(R.id.coordinatorLayout), "Item deleted", Snackbar.LENGTH\_LONG)

snackbar.setBackgroundTint(Color.BLUE) // Change the background color

snackbar.show()

```

**- Toast:**

**- Example:** Toasts have a simpler appearance without much styling.

```kotlin

Toast.makeText(applicationContext, "Network error", Toast.LENGTH\_SHORT).show()

```

**3. Customization**

**- Snackbar:**

- Example: Snackbars can be customized to change text colors, background, and action buttons.

```kotlin

val snackbar = Snackbar.make(findViewById(R.id.coordinatorLayout), "Item deleted", Snackbar.LENGTH\_LONG)

snackbar.setActionTextColor(Color.RED) // Change action button text color

snackbar.setAction("UNDO") {

// Undo action

}

snackbar.show()

```

**- Toast:**

- Example: Toasts have limited customization, though you can create a custom layout.

```kotlin

val customToast = Toast(applicationContext)

val layout = layoutInflater.inflate(R.layout.custom\_toast, null)

customToast.view = layout

customToast.duration = Toast.LENGTH\_SHORT

customToast.show()

```

**4. Visibility**

**- Snackbar:**

**- Example:** A Snackbar appears at the bottom of the screen and overlays other UI elements, making it more noticeable.

```kotlin

Snackbar.make(findViewById(R.id.coordinatorLayout), "Item added", Snackbar.LENGTH\_LONG).show()

```

**- Toast:**

- Example: A Toast appears without overlapping other UI elements.

```kotlin

Toast.makeText(applicationContext, "Item added to cart", Toast.LENGTH\_SHORT).show()

```

**Summary**

**- Snackbar:** Ideal for providing context-specific feedback that may require user action, such as confirming or undoing actions.

**- Toast:** Suitable for simple messages that inform users without needing interaction.

By using these examples, you can effectively decide when to implement Snackbar or Toast in your Android application based on the type of user feedback you want to provide.