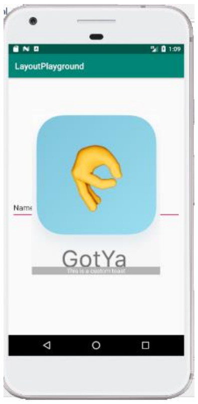
Challenge 0: Custom toast

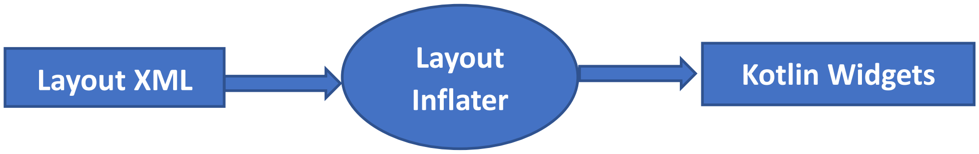
Create a custom toast

For this challenge, create a custom Toast, that shows a picture and a text of your choice.



- To change the UI you must create a layout (save it as a .xml file in the /res/layout folder) and define a layout, (linear, for example). Add an *ImageView* and a *TextView* widget to configure the layout of your Toast.

- To use the previous Layout as your toast view, you should use *LayoutInflater* class. This class converts XML into Kotlin widget objects and
  - Let you specify a chunk of layout, as XML, and then load it into Kotlin as needed.
  - It is similar to a fragment (we will study fragments in week 3), but without its own event or lifecycle.


Layout XML

Layout Inflater

Kotlin Widgets

Here is some code to help you get started

```
fun show_custom_toast(){
    val inflater = layoutInflater
    val container: ViewGroup? = name_of_your_xml_toast_file
    val layout: View = inflater.inflate(R.layout.name_of_your_xml_toast_file,
container)
    val textToShow : TextView = layout.id_of_your_layout_TextView
    textToShow.text = "This is a custom toast"
    with (Toast(applicationContext)) {
       //set gravity
        //set duration
         view = layout
         show()
     }
```

Toasts basics

A toast is a view containing a quick little message for the user. The toast class helps you create and show those.

When the view is shown to the user, appears as a floating view over the application. It will never receive focus. The idea is to be as unobtrusive as possible, while still showing the user the information you want them to see. An example is a brief message saying that the message has been sent. It is important to note that Toasts are not clickable. If user response to a status message is required, consider instead using a ***Notification*** (we will study Notifications later in the course).

First, instantiate a Toast object with one of the makeText() methods. This method takes three parameters: the application Context, the text message, and the duration for the toast. It returns a properly initialized Toast object. You can display the toast notification with show(), as shown in the following example:

```
val text = "This is the text of the toast!"
val duration = Toast.LENGTH_LONG
val toast = Toast.makeText(applicationContext, text, duration)
toast.show()
```

or alternatively:

```
Toast.makeText(context, text, duration).show()
```

Locate the Toast in your App

A standard toast notification appears near the bottom of the screen, centered horizontally. You can change this position with the setGravity(int, int, int) method. This accepts three parameters: a Gravity constant, an x-position offset, and a y-position offset.

For example, if you decide that the toast should appear in the top-right corner, you can set the gravity like this:

```
toast.setGravity(Gravity.TOP or Gravity.RIGHT, 0, 0)
```

If you want to nudge the position to the right, increase the value of the second parameter. To nudge it down, increase the value of the last parameter.