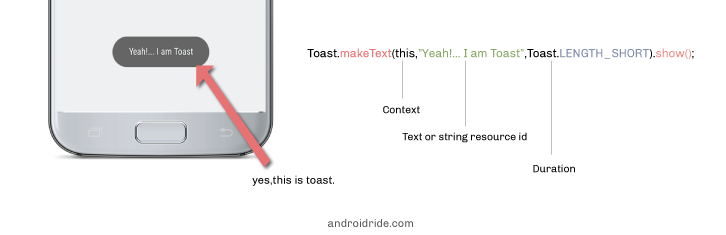
# Bài 1. Nhập xuất đơn giản

## Mục tiêu:

* Hiểu cách lấy tham chiếu đến 1 widget trên LayOut XML khi code xử lý ở file java **findViewByID**
* Hiểu cách xây dựng hàm sẽ được thực hiện (trong java file) đáp ứng sự kiện “OnClick” của nút nhấn
* Biết cách hiện thông báo đơn giản với lớp **Toast**

## Tài liệu



## Yêu cầu Thực hành

Say Hi!

Examples 1

Xin chao, Toi la ABC

Id?

**btnHello**

Sử dụng Lớp **Toast**

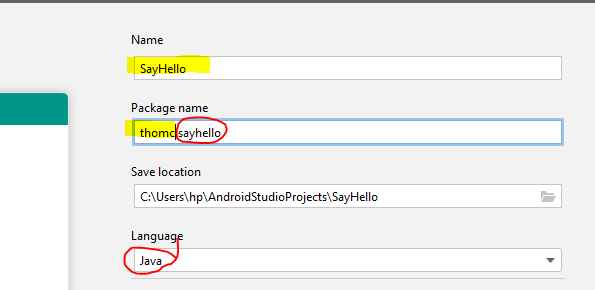
Toast.makeText( A, S, L ).show();

**L**: thời gian muốn hiện

**S**: chuỗi muốn hiện

**A**: Tên\_lớp\_hiện\_tại.**this**

**Bước 1.** Tạo dự án, Đặt tên dự án, tên Package <tên\_mình>.Tên\_Dự án



Khi này, Android Stu tạo sẵn 2 tệp đặc biệt

* Tệp mô tả giao diện (Activity) (**.xml**)
* Tệp chứa mã lệnh xử lý gắn với Activity ngay trên (**.java**)

(*Chú ý: Mỗi giao diện, đương nhiên cần có một file java tương ứng*)

( Khi chạy ứng dụng, Hàm OnCreate trong file Java được chạy “đầu tiên”, trong hàm này, thực hiện việc nạp file giao diện .xml, thông qua hàm setContentView(R.layout.Tên\_File\_XML); )

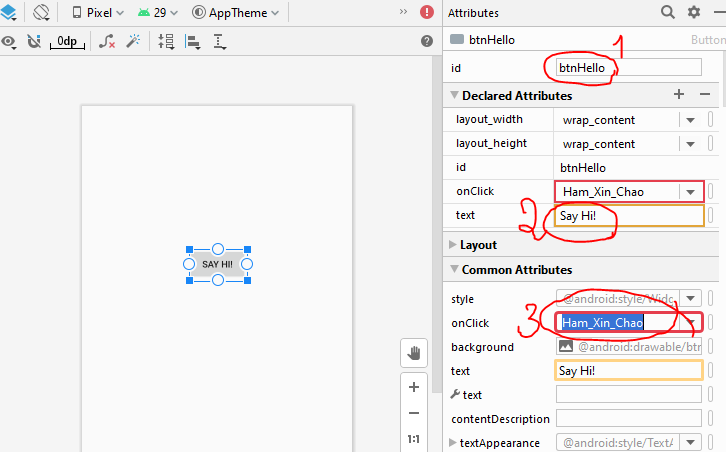
**Bước 2**. Thiết kế giao diện (Layout, file activity\_main.xml)

* Chú ý, đặt giá trị cho thuộc tính ID cho mỗi điều khiển

*Khuyên*: viết\_tắt\_của\_Wiget\_Tên

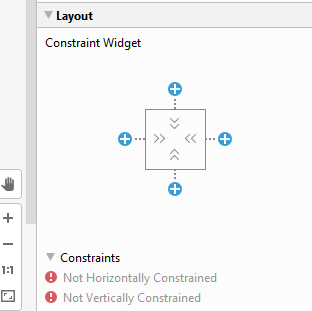
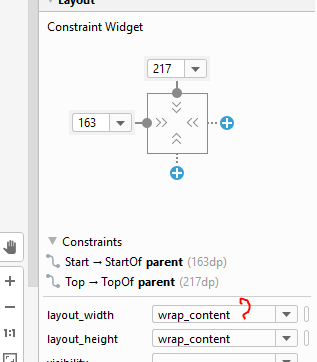
*Ví dụ*: btn**Hello**

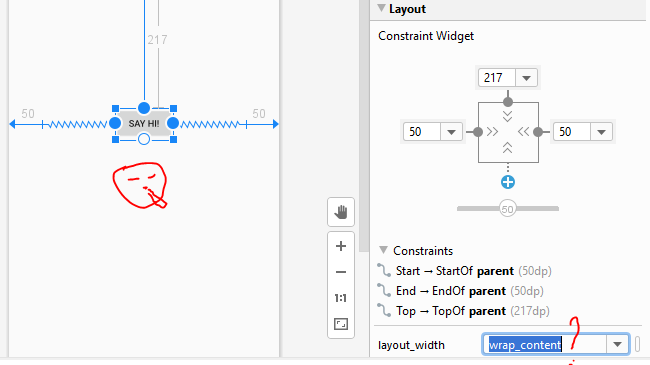
*Nhằm mục tiêu, ta xác định được đối tượng cần tương tác khi viết mã xử lý ở file Java*

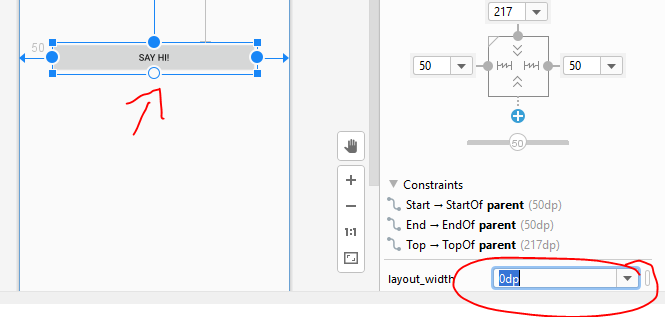


**!!** Vị trí **3**. Qui định tên hàm (ở file java) sẽ được gọi thực hiện khi User nhấn lên nút

Tiếp theo, **ta “neo”** đối tượng vào biên (hoặc vào đối tượng khác)





**Bước 3**. VIết mã ở file JAVA (file MainActivity.java)

