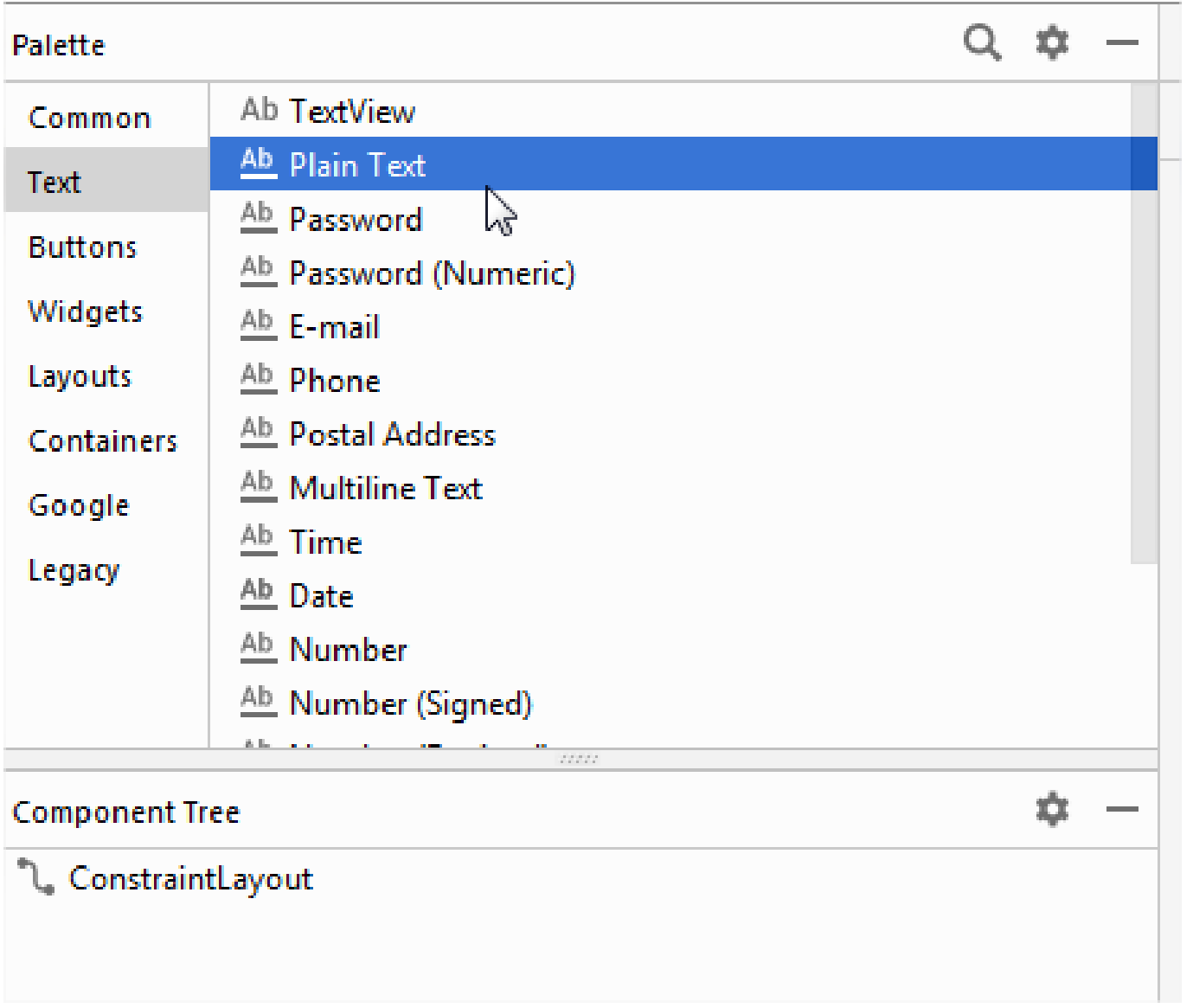
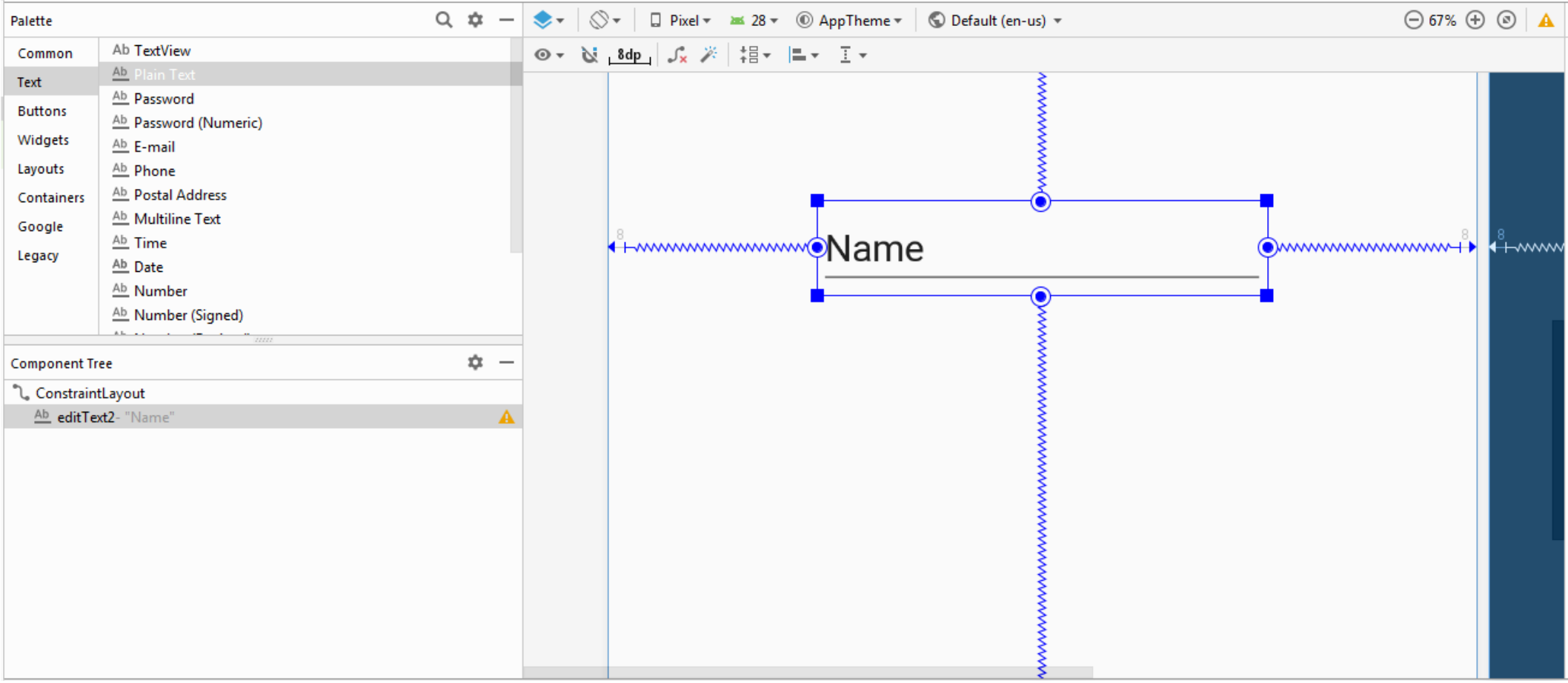
Utilisation des objets xml de type EditText

Android Studio

EditText



Android Studio



Android Studio

String strName = monEditTEXT.getText().toString();

on récupère un String strName grâce au getText(), mais getText est d'un genre particulier, il faut lui signaler le type par toString().

On peut faire un peu de contrôle de saisie avec un : if (strName.equals(" ")) return;

Pour que return soit "seul", il faut que la méthode qui l'englobe ne renvoie donc rien (void). Idéal dans un Public void onClick...

```
BTN1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        if (STRCFOND.equals("")) return;
```

si on clique sur le bouton, et que notre chaîne de caractère est vide on fait une sortie via return. Du coup on peut contrôler s'il y a bien eu une saisie ou pas. Le choix de votre EditTexte déterminera le type de données que l'utilisateur peut saisir.

```
int maValeurEnInt = Integer.parseInt(monString);
```

Dans le cas ou vous devez transformer votre "saisie" (qui est de type String) en entier.

```
double monDouble = Double.parseDouble(monString);
```

Pour aller plus loin… https://developer.android.com/reference/java/lang/Integer

PARSE = Analyser, cette fonction sera souvent utile...

```
String pseudo = et_pseudo.getText().toString().trim();
String email = et_email.getText().toString().trim();
String password = et_password.getText().toString().trim();
String password2 = et_password2.getText().toString().trim();
```

La méthode trim() permet d’enlever les espaces avant et après la saisie