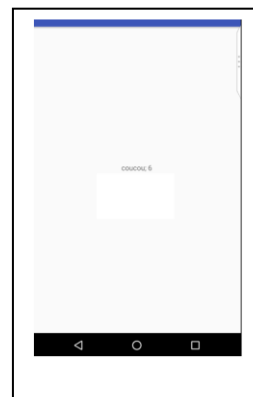
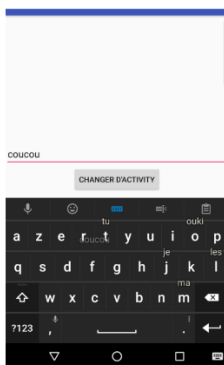


Exercice 1

l'objectif de l'exercice est de passer d'une activité 1 a une activité 2, tout en faisant passer des données de différents type, entre les deux activités, et on s'en servira des extra dans ceci ;



```
public void ChangeActivity(View view)
{
    EditText = (EditText) findViewById(R.id.et1);
    String str = editText.getText().toString();
```

nous allons pouvoir passer deux types de données différents : une String et un entier. Pour récupérer cet entier, il suffit d'appeler la méthode length sur notre String :

```
int nbLettre = str.length();
```

pour passer d'une activité à une autre, nous utilisons un Intent. Dans notre cas, pour passer de notre « MainActivity » à « MaDeuxiemeActivity », nous utilisons le code suivant (toujours dans la méthode ChangeActivity(l'évenement onClick du bouton)) :

```
Intent intent = new Intent(this, MaDeuxiemeActivity.class);
startActivity(intent);
```

Nous allons ajouter des extras à cet Intent (ce seront les données), à savoir la String et l'entier.

Comment fonctionne un Extra ? Un extra est un couple clé/valeur, basé sur le système Bundle :

- la clé est l'identifiant, tu peux mettre ce que tu veux, c'est simplement une chaîne de caractères ;
- la valeur est la valeur de ta donnée. Elle sera associée à la clé.

Ainsi, quand tu vas créer un extra, tu vas lui donner une clé et une valeur. En revanche, quand tu voudras récupérer ton extra, tu lui donneras la clé et il te renverra la valeur associée. comment faire ça concrètement ?

Tu vas pouvoir ajouter des extras à ton Intent entre le moment où tu as créé ton Intent et le moment où tu appelles startActivity. Pour cela, tu vas appeler la méthode putExtra sur ton Intent qui prend deux arguments :

- le premier argument : la clé ;
- le second : la valeur.

Tu ajoutes la String et l'entier comme extra, ce qui doit te donner :

```
Intent intent = new Intent(this, MaDeuxiemeActivity.class);
intent.putExtra("edittext", str);
intent.putExtra("nbLettre", nbLettre);
startActivity(intent);
```

Si tu avais voulu utiliser directement le système *bundle*, le code aurait plutôt ressemblé à ça : (Les bundles sont généralement utilisés pour transmettre des données entre diverses activités android)

```
Intent intent = new Intent(this, MaDeuxiemeActivity.class);
Bundle bundle = new Bundle();
bundle.putString("edittext ", str);
bundle.putString("nbLettre", nbLettre);
intent.putExtras(bundle);
startActivity(intent);
```

Récupérer les extras dans la seconde activité :

Dans ta seconde activité, dans la méthode onCreate et juste après l'appel de la méthode setContentView, nous allons pouvoir récupérer nos extras.

Pour cela, tu dois d'abord récupérer l'Intent de la manière suivante :

```
Intent intent = getIntent();
```

Si l'Intent récupéré n'est pas null, tu vas pouvoir effectuer les actions nécessaires pour récupérer les extras. A priori, Intent n'est pas null, mais c'est une bonne pratique de vérifier, quand tu utilises un objet, qu'il n'est pas null. Ça t'évitera des crashes si jamais il est null pour une raison x ou y. On le fait pour éviter les soucis plus tard.

```
if (intent != null) {  
    // Le code pour récupérer les extras ira ici  
}
```

On va vouloir, dans un premier temps, récupérer notre String. Tu vas donc créer la variable à laquelle sera assignée la valeur que l'on va récupérer :

```
String str = "";  
if (intent.hasExtra("edittext")) { // vérifie qu'une valeur est  
    associée à la clé "edittext"  
    str = intent.getStringExtra("edittext"); // on récupère la  
    valeur associée à la clé  
}
```

Si on a utilisé un bundle :

```
Bundle bundle = getIntent().getExtras();  
str = bundle.getString("edittext");
```

pour récupérer le int on utilise la méthode : « getIntExtra »

```
int nbLettre = intent.getIntExtra("nbLettre", 0);
```

on renvoi zero si la clé nbLettre n'existe pas

Afficher les données que nous avons récupérées

```
TextView textView = (TextView) findViewById(R.id.tv1);
```

Nous allons afficher la String récupérée, suivie d'un point-virgule et du nombre de lettres contenues dans la String :

```
textView.setText(str + "; " + nbLettre);
```