

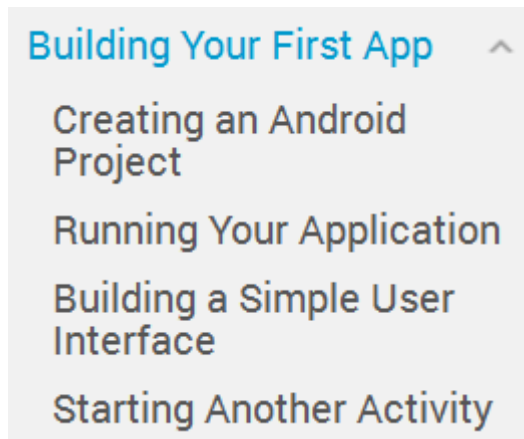
# Android

## Installation de l'environnement

### Premiere Application

Faire le tutorial Android

<http://developer.android.com/training/basics/firstapp/index.html>



### Questions

Quel fichier decrit les caracteristique fondamentale de l'application et de ses composants ?	
Dans quel repertoire le trouver ?	
Dans quel repertoire se trouve les sources de l'application ?	
A quoi servent les fichiers de "resource de l'application" ?	
Dans quel repertoire sont-ils ?	
A quoi sert le signe '@' devant un nom dans le fichier xml ? Ex: <code>android:id="@+id/edit_message"</code>	
A quoi sert le signe '+' devant un nom de resource ? Ex: <code>android:id="@+id/edit_message"</code>	
Un fichier R.java est généré quelque part dans le projet. Essayez de le localiser. Que contient-il ?	
Comment répondre au click sur un bouton ?	
A quoi sert un "Intent"	

## Application Palindrome

Vous allez développer une application permettant de vérifier qu'un mot est un palindrome. Un palindrome est un mot qui s'épelle de la même façon de droite à gauche ou de gauche à droite ('noyon', 'été' ).

L'application propose un écran principal contenant un 'EditText' pour entrer le mot à tester, un bouton 'Test' pour tester le texte, et un bouton 'Clear' pour effacer le texte. Le texte et le bouton 'send' sont sur une même ligne. Le bouton 'clear' est centré sur une autre ligne.

Quand l'utilisateur soumet un mot en entrant un texte puis en appuyant sur le bouton 'Test', l'application vérifie si le mot est un palindrome. Si le mot est un palindrome, l'application affiche un nouveau écran contenant un message "Bravo, le mot 'xxx' est un palindrome", ou 'xxx' doit être remplacé par le palindrome. Si le mot n'est pas un palindrome, l'application affiche un 3eme écran affichant le message "Désolé, le mot 'xxx' n'est pas un palindrome".

Le bouton 'Clear' du premier écran permet d'effacer le texte déjà entré.

### Quelques indications

- Pour effacer le texte d'un EditText, il faut appeler la méthode clear()
  - `editText.getText().clear();`
- Les Layout peuvent être imbriqués. Pour avoir deux lignes l'une en dessous de l'autre, il faut un LinearLayout 'vertical'. Pour avoir deux éléments l'un derrière l'autre dans une seule ligne, il faut un LinearLayout 'horizontal', imbriqué dans le LinearLayout 'vertical'