

TP1 :

Contexte :

Le Prof du Module « Programmation Mobile » demande de ses étudiants à réaliser une application Mobile sous Android pour saisir les informations concernant les étudiants de 3^{ème} année (Matricule, Nom, Prénom, Groupe et la moyenne du module).

Objectif

- Créer un formulaire par XML.
- Utiliser les différents types de données.
- Suggérer les manques dans IHM.
- Proposer les améliorations.

Travail demandé :

(I). Le démarrage de l'Android

1. Démarrez éclipse
2. Créez un nouveau projet et l'appelé par « TP01_NomPrenonEtudiant »

(II). L'interface utilisateur

3. Ouvrir le fichier MainActivity.xml du projet
4. (Reproduire cet interface IHM), c'est-à-dire : créer le formulaire représentant les informations d'un étudiant de 3^{ème} année suivantes :

(III) Les types de données suivants :

Utiliser la propriété : android:inputType = « type de données »

- Un identifiant (Matricule de l'étudiant) de Type **Numérique** (**Number**)
- Le nom de l'étudiant de type **Text** (**textCapCharacters**)
- Le prénom de l'étudiant de type **Text** (**textCapWords**)
- Le groupe de l'étudiant de Type **Numérique** (**Number=1, 2, 3, 4 , 5**)
- La moyenne du module Applications Mobile de Type **Décimal** (**numberDecimal**)
- Un bouton « Aficher »
- Un TextView (Text) contenant toutes ces informations (toute la liste d'étudiants de 3^{ème} année).

Utilisez le fichier TP1 pour vous permettre de modifier les contraintes des composantes graphique

Détail

Objectif du TP :

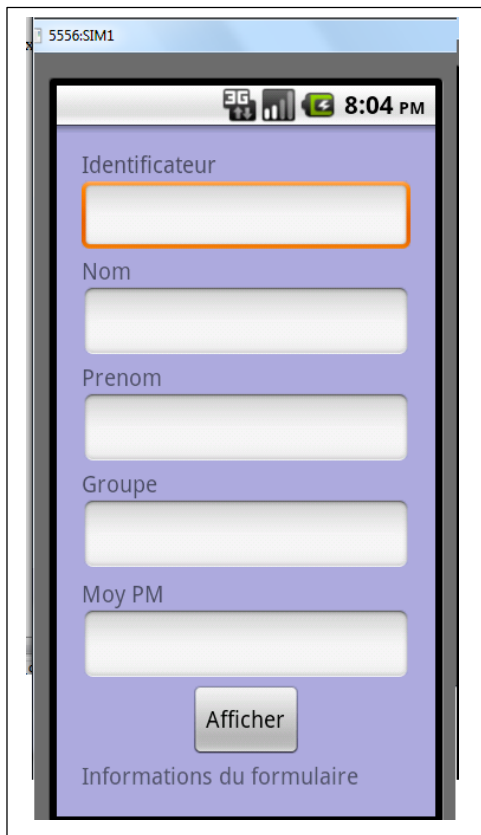
Pour créer cette application et montrer les différentes propriétés des vues graphique (wedgets)

1. Créer un nouveau projet du nom : TP1
 - a. Nommer l' Activity (fichier source) par **MainTp1**
 - b. Nommer le Layout (fichier xml) par **maintp1**
2. Ajouter les vues suivantes: TextView1, EditText1, TextView2, EditText2, TextView3, EditText3, TextView4, EditText4, et 2 boutons (Button1, Button2).

3. Renommer les respectivement par:

Matricule, Mat, Nom Etudiant, Nom, Prenom Etudiant, Prenom , Moyenne PM, Moy, Afficher, Suivant.

4. Changer le type du EditText (Matricule) en **number**
5. Changer le type du EditText (Moyenne) en **numberDecimal**
6. Changer la couleur de RelativeLayout en **#abc**
7. Ajouter autre Activity et la nommée par **Affichage**
8. Ajouter autre Layout et la nommée par **affichage**



Questions & objectifs :

- Quel est le manque remarqué dans ce formulaire (coté informationnel).

- Comment faire pour fixer le nombre de groupes saisi (ne permettre que les chiffres 1, 2, 3, 4 ou 5).
- Comment faire pour fixer le Matricule d'étudiant aux **5 chiffres maximum**.
- Comment changer la couleur du formulaire (background). `android:background="#abdfffff"`
- Supprimer les **textView** qui contiennent Identificateur, Nom, prénom, Groupe et Moyenne et utiliser la propriété **Hint**
- Proposer des améliorations du formulaire conçu pour compléter le manque.

TD & TP (Homework)

Contexte :

Créer une application Android : « examen sous forme QCM », où le formulaire xml comporte :

- 3 Questions ; chaqueune dans une Layout (Activity),
- Chaque question avoir 2 réponses (une seule réponse est juste)
- Chaque activité comporte 3 boutons ; une pour **suivant**, autre **précédent** et la dernière pour **valider**.
- Ajouter une autre activité (activity2).
- Le résultat s'affichera dans une **TextView** dans la 2^{ème} activité après l'appui sur le bouton valider.

Un message Toast s'affichera pour donner la correction et le score

```
Toast.makeText(this, "message à afficher", Toast.LENGTH_LONG).show();
```

- Le score se calculera par la méthode suivante :
 - Chaque réponse vraie a un score d'un point en plus, par contre si la réponse est fausse le score est démunie par 1 (-1).
 - Un message confirme le score obtenu avec une félicitation ou avertissement.

Aide :

Comment adapter le type de données à vos besoins

Dans ce TP, l'utilisateur aura peut-être besoin de saisir quelques données de types différents. Dans ce cas, il faut faciliter le travail à l'utilisateur en faisant apparaître un clavier en fonction du type de champ à remplir (adresse mail, numéro de téléphone etc...).

Pour cela dans votre "**EditText**", il faut simplement spécifier l'option **android:inputType**. Voici les différentes valeurs disponibles :

- **text** (par défaut) : clavier normal
- **textCapCharacters** : Clavier tout en majuscule
- **textCapWords** : Première lettre automatiquement en majuscule
- **textCapSentences** : Phrase en majuscule
- **textAutoCorrect** : Activer la correction automatique
- **textAutoComplete** : Activer la correction automatique forcée
- **textMultiLine** : Texte sur plusieurs lignes
- **textNoSuggestions** : Pas de suggestion de correction
- **textUri** : Saisie d'une url web
- **textEmailAddress** : Adresse mail
- **textEmailSubject** : Sujet de mail
- **textShortMessage** : Active le raccourci smiley sur le clavier
- **textPersonName** : Saisie du nom d'une personne (affichage de speech to text en bas a gauche du clavier)
- **textPostalAddress** : Saisie d'une adresse postale (affichage de speech to text en bas a gauche du clavier)
- **textPassword** : Saisie de mot de passe
- **textVisiblePassword** : Saisie d'un mot de passe visible
- **textWebEditText** : Text Web (activation de raccourci tabulation et speech to text)
- **number / numberSigned / numberDecimal / phone / datetime / date / time** : Clavier numérique

Pouvoir mettre plusieurs valeurs en les séparant simplement par des "|".