contrôle Android Studio

# Presentation d’android

1. Android est un framework de développement mobile.  
   L’intérêt qu’il soit basé sur un noyau Linux est ………….
2. La version actuelle d’android est Nougat (v7)
3. Un émulateur permet de simuler le comportement d’une machine sur une machine d’un autre type.  
   Par exemple, dans le cas du développement Android, on va simuler un appareil mobile tournant sous l’OS pour voir comment l’application en développement fonctionne.
4. Je ne sais pas, je dirais car Android Studio est plus performant.

# Apk

1. Un APK est un exécutable de notre application (fichier binaire) qui permet d’installer celle-ci sur un appareil. Il contient toute les ressources du programme. Il existe les APK signé et non signé. L’inconvénient d’un APK non signé est qu’il ne peut pas être distribué.
2. Le KeyStore permet de sauvegarder les clés des applications créées.
3. Je ne sais pas.

# Composantes d’android

1. Une activité est, de manière basique, un écran de notre application sur le quel est représenté une fonctionnalité. Par exemple l’écran mosaïque de la galerie photo est une activité.
2. Un service est une exécution de commande qui ne nécessite pas forcement d’activité et qui sont appliqués en tache de fond (téléchargement, etc.)
3. Je ne sais pas.

# Cycle de vie d’une activité

1. Les 4 états d’une activité (pas forcément dans l’ordre):
   1. En fonction : L’activité est en train de tourner en plein écran.
   2. A l’arrêt : On a quitté l’application, elle existe toujours mais ne fais plus partie des propriétés de l’appareil, il peut s’en débarrasser s’il juge qu’il a besoin d’espace mémoire.
   3. En pause : L’application tourne en fond, elle n’est plus prioritaire dans l’utilisation de la mémoire de l’appareil. On peut y retourner à n’importe quel moment, son état est sauvegardé.
   4. Terminée : L’instance de l’application est vidée de la mémoire.
2. La méthode onCreate() qui est la première méthode appelée lorsque l’application est lancée.

# Contexte d’une application

1. Le contexte représente « l’environnement » de l’application. Grace à lui on a accès à différents éléments que contient l’application tel les layout, etc.
2. Les méthodes :
   1. getApplicationContext() : retourne le contexte de ladite application.
   2. getContext() : retourne le context de la vue en cours
   3. getBaseContext() : Je ne sais pas.
   4. this : retourne le context de l’application egalement.

# Le manifeste

1. Le manifeste contient les informations globales de notre application. C’est la fiche d’identité de l’application.
2. On le trouve dans le dossier app de notre application.
3. C’est ici que sont renseignés : Son nom, son icône, ses activités, etc.
4. Nom, Icône, Theme visuel de l’application, la liste des activités et des intent-filters.

# Architecture d’un projet

1. Le répertoire Java contient les classes java utiles au fonctionnement de l’appli
2. Le répertoire RES contient les ressources du projet.
   1. Drawable contient les éléments graphiques de l’application.
   2. Layout contient les écrans de l’appli
   3. Mipmap contient également des ressource graphiques (et sonore ?)
   4. Values contient tous les éléments de type textuels, de configuration (c’est un peu un dictionnaire de valeurs auxquels sont associés des identifiants).
3. Etant du xml, il est facile d’encapsuler les différentes « node » dans des balises de langues (par exemple <texte><francais></francais></texte>. Pour gérer les différentes langues, on peut supposer que l’on récupère la valeur de celle-ci pour appeler :

var\_langue + node\_correspondant\_au\_texte\_voulu