

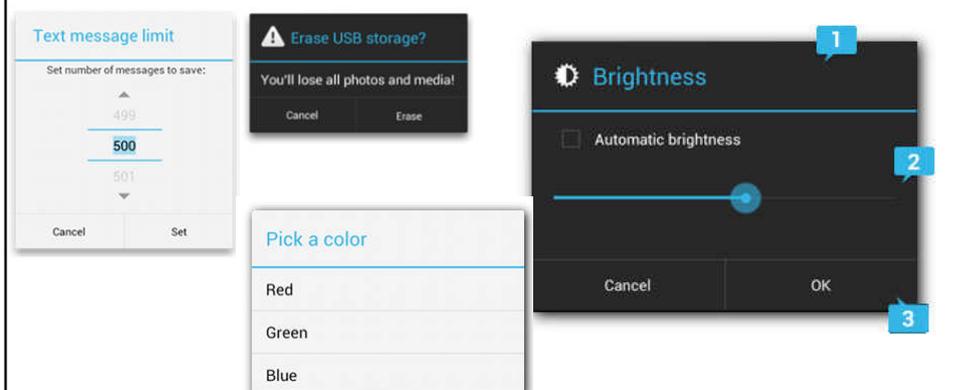
Département d'Informatique, Licence S06 Cours développement mobile

Les boites de dialogue



Les boites de dialogue

- Une application a parfois besoin de s'exprimer.
- En utilisant les messages , l'application peut communiquer des informations à l'utilisateur concernant les traitements lancés et aussi les possible erreurs qui peuvent avoir lieu.

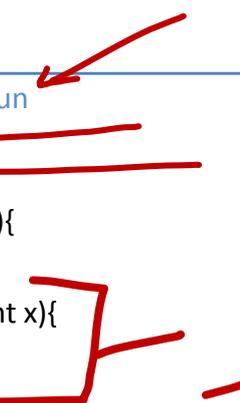


Les toasts

- Se sont des messages qui affichent des informations et disparaissent sans intervention de l'utilisateur.
- Exemple :
- **Toast.makeText(this, "Exemple de texte", Toast.LENGTH_SHORT).show();**

Les alerts

```
AlertDialog.Builder(this).setTitle(" ce ci est un  
test").setMessage("le contenu de  
msg").setNeutralButton("OK",  
    new DialogInterface.OnClickListener(){  
  
    public void onClick(DialogInterface dlg, int x){  
        // evenement du button  
    }  
}
```



```
AlertDialog diag=AlertDialog.Builder(this);
diag.setTitle("Message de confirmation");
diag.setMessage("confirmez-vous cette action");

DialogInterface.OnClickListener lst=
new DialogInterface.OnClickListener(){
    public void onClick(DialogInterface dlg, int x){

        //ici on va traiter l'action de clique sur le bouton Oui.
    }
}
diag.setNeutralButton( "Oui",lst);
```

Dialogue avec plusieurs valeurs pré-saisîtes

- Ce type de dialogue, offre au développeur la possibilité de permettre une interaction avec l'utilisateur sans saisie (ce dernier doit simplement sélectionner une valeur parmi des valeurs pré-saisîtes dans une liste de choix).
- Ce type d'interaction permettent d'éviter les **erreurs** de saisi et aussi permet de réduire la **charge cognitive**.

Un exemple

```
public Dialog onCreateDialog(Bundle savedInstanceState) {
    AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
    builder.setTitle(Pick a Color)

    .setItems(R.array.colors_array, new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            // The 'which' argument contains the index position
            // of the selected item
        }
    });
    return builder.create();
}
```

La fonction **setItems** prend en argument le tableau de valeurs pour remplir la liste.

Comme résultat on obtient l'index de la valeur sélectionnée.

Dialogue de type Oui / Non

```
public void open(View view){
    AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
    alertDialogBuilder.setMessage("Are you sure,You wanted to make decision");

    alertDialogBuilder.setPositiveButton("yes", new DialogInterface.OnClickListener()
        @Override
        public void onClick(DialogInterface arg0, int arg1) {
            Toast.makeText(MainActivity.this,"You clicked yes button",Toast.LENGTH_LO
        }
    });

    alertDialogBuilder.setNegativeButton("No",new DialogInterface.OnClickListener()
        @Override
        public void onClick(DialogInterface dialog, int which) {
            finish();
        }
    });

    AlertDialog alertDialog = alertDialogBuilder.create();
    alertDialog.show();
}
```

Dialogue de type Oui / Non

- Notez que la fonction **setMessage**, permet de définir la chaîne de caractère qui forme le message de la question.
- **setPositiveButton** : permet de définir le bouton qui prend le rôle de la réponse « oui » , donc positive
- **setNegativeButton**: permet de définir le bouton de la réponse « non » (négative).