

Construïnt Layouts mitjançant un Adaptador

Per a que volem un Adaptador?

- Els controls de selecció, necessiten accedir a les dades mitjançant Adaptadors.
- Quan dissenyem un layout, no sempre coneguem quants elements el formaran.
- Si volem omplir una llista desplegable (ListView) amb els elements disponibles en una base de dades, a priori, no sabem quants elements ens tornarà la consulta.

Quan el contingut dels layouts és dinàmic o no el coneguem quan dissenyem el layout

Com s'utilitza un Adaptador?

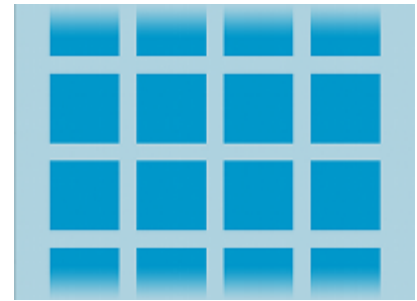
- **Sabem quants element s'han de mostrar?**
- Si no sabem quants elements hi hauran al layout, definim el layout com una subclasse d'[AdapterView](#).
- Una subclasse d'[AdapterView](#) utilitza un [Adapter](#) per enllaçar les dades amb el seu layout.
- L'[Adapter](#) es converteix en un intermediari entre qui ens proporciona les dades i el layout subclasse [AdapterView](#)
 - El [Adapter](#) rep les dades (des d'un array o d'una consulta a la base de dades) i transforma cada entrada en un view que es pot afegir al layout [AdapterView](#).

Exemple del resultat

- ListView



- GridView



- Mostra una llista desplegable **desplaçable** d'una sola columna

- Mostra una quadrícula **desplaçable** amb files i columnes

Treballant amb un Adaptador

- Els dos adaptadors més utilitzats per Android són:
- **ArrayAdapter**: Utilitzarem aquest adaptador quan les dades estiguen disponibles des d'un array.
- **SimpleCursorAdapter**: Quan les dades les rebem mitjançant un cursor (després de fer una consulta a una base de dades).

Treballant amb un Adaptador

- Hi ha un altre Adaptador anomenat
- **SimpleAdapter**: Utilitzarem aquest adaptador per a mapejar dades sobre els controls definits en un fitxer XML (un layout).
- Exemples:



Spinner



Spinner amb RadioButtons

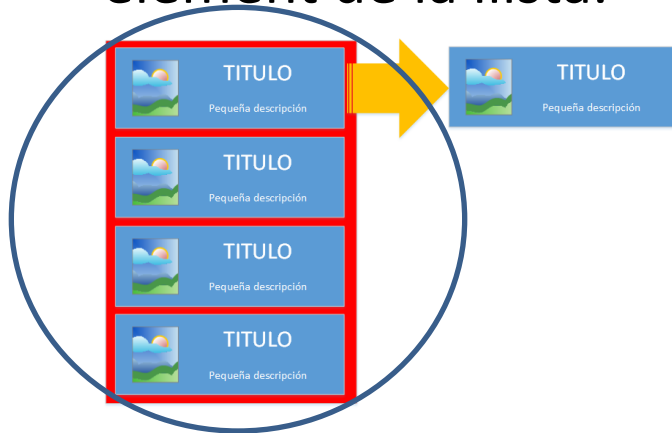
ArrayAdapter

```
1    final String[] dades = new
2        String[]{"Dilluns", "Dimarts", "Dimecres", "Dijous"};
3    ArrayAdapter<String> adaptador =
4        new ArrayAdapter<String>(this,
5            android.R.layout.simple_spinner_item, dades);
6
7    adaptador.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
```

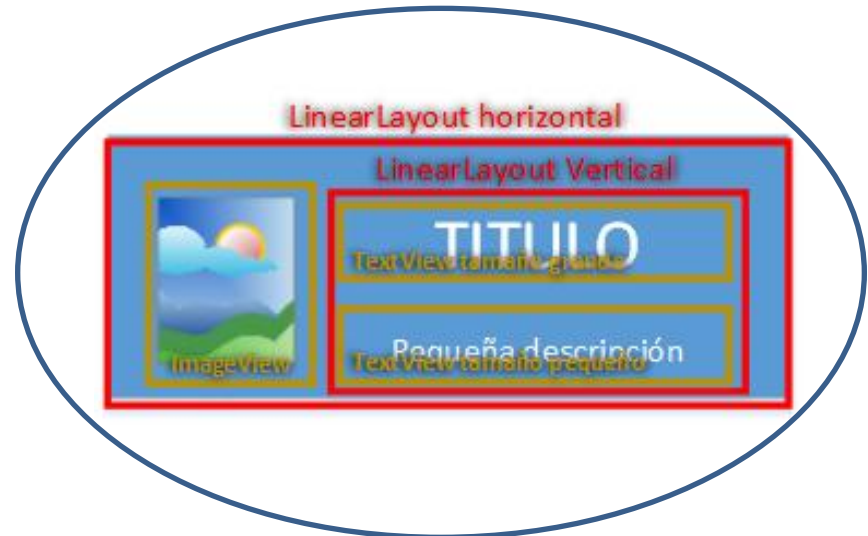
- El Constructor d'un [ArrayAdapter](#) sol rebre 3 parametres:
 1. Context: El Context de l'activity on es crea l'Adaptador.
 2. L' ID del *layout* sobre el qual es mostraran les dades.
 3. L'array amb les dades a mostrar
- A l'adaptador li indiquem quin layout ha d'aplicar quan desplegue la llista d'opcions.

SimpleAdapter

- Ens permet personalitzar la manera en que es mostraran cada element de la llista.



llista.xml



element.xml

- Crearem un layout xml per definir l'estructura que tindrà cada element. (element.xml)
- Crearem un layout xml per definir un ListView (llista.xml)

SimpleAdapter

- Començarem creant una classe per a gestionar la informació de cada element (Element.java):
- La informació que es mostrarà en cada element de la llista l'extraurem d'objectes de la classe “*Elements*”.
- Així, podríem tindre un ArrayList d'objectes Elements amb tota la informació que ha de mostrar la nostra llista, an l'Activity principal, clar.

Ara bé, qualsevol llista en Android, necessita d'un Adaptador que herete la classe BaseAdapter.

Anem a vore com el podem crear.

SimpleAdapter

- La classe que fem servir com Adaptador hereta BaseAdapter.
- Al construir la classe ens demana implementar els següents mètodes:

@Override

public int getCount () {}	//Numero d'entrades que hi ha en l'arrayList
public Object getItem (int position) {}	//Torna l'Element position de l'arrayList
public long getItemId (int position) {}	//L'identif. Del Item es la seua posició
public View getView (int position, View convertView, ViewGroup parent) {}	//Torna un View que relaciona dades amb element.xml

- Al constructor cal passar-li el context de l'aplicació, el id del layout de l'entrada ("R.layout.element"), y l'ArrayList que conté les dades.
- ```
public AdaptadorLlista (Context context, int R_layout_IdView, ArrayList<?> entradas) {
 super(); this.context = context; this.entradas = entradas; this.R_layout_IdView = R_layout_IdView;
}
```

# SimpleAdapter

- En l'Activity, creem l'arrayList d'Elements.
- Li afegim els Elements amb les dades que volem visualitzar.
- Obtenim l'objecte que referencia al desplegable (ListView, Spinner, GridView, etc).
- Li afegim un adaptador, que serà AdaptadorLlista.
- Implementem el mètode onEntrada que serà qui enllaçarà les dades a representar amb els elements del layout element.xml

# SimpleAdapter

- Obtindrem una cosa semblant a :



# SimpleAdapter

- Ejercicio: Modifiqueu el que calga per  
Obtenir un Spinner amb el  
Següent aspecte:

