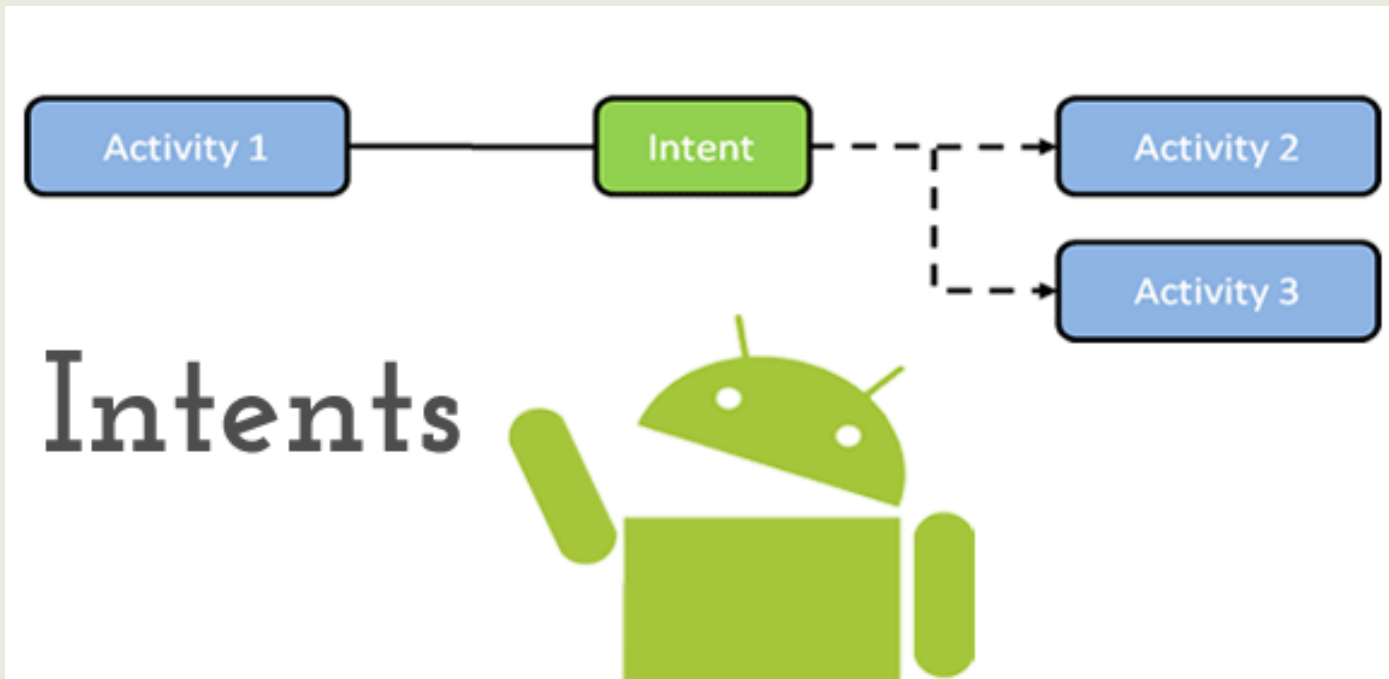


# GÉRER PLUSIEURS ACTIVITÉS

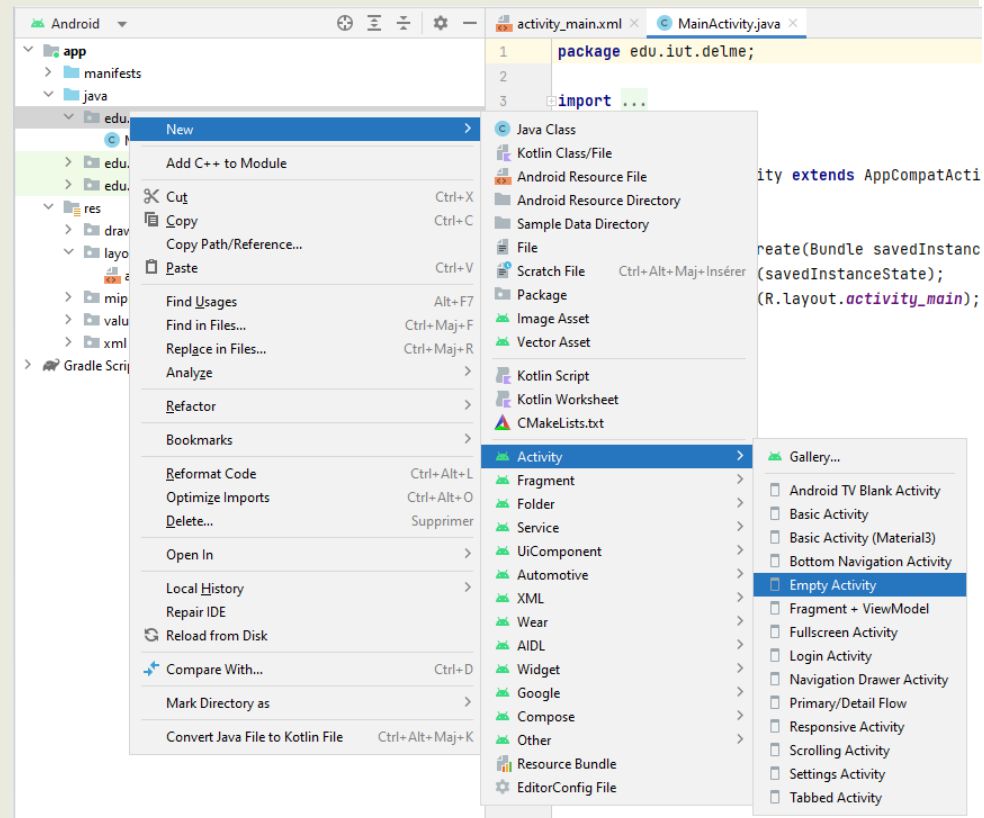
- ❑ Les Intents
- ❑ Déclarer et lancer une nouvelle activité
- ❑ Echange de données entre activités
- ❑ Echange de données parcelables



# CONTEXTE D'UTILISATION

On a créé 2 activités, chacune dispose :

- Une classe qui étend Activity
- Une vue dans un fichier xml
- Une déclaration dans le Manifest



# CONTEXTE D'UTILISATION

- Les *Intents explicites* permettent de gérer l'envoi et la réception de **messages** afin de faire coopérer les activités.
- Créer un *Intents*, c'est un peu comme acheter un billet de train :
  - **départ** : ici
  - **Arrivée** : *SecondActivity*
  - **Bagages** : la valeur 27



values	14	
colors.xml	15	
strings.xml	16	
themes (2)	17	
xml		
Gradle Scripts		

```
Intent intent = new Intent( getApplicationContext(), SecondActivity.class);
intent.putExtra( getString(R.string.VALUE), value: 27);
startActivity(intent);
```

# CONTEXTE D'UTILISATION

Ça fonctionne avec les types  
primitifs + String + ??

→ MAIS pas avec une Pizza

- MainActivity

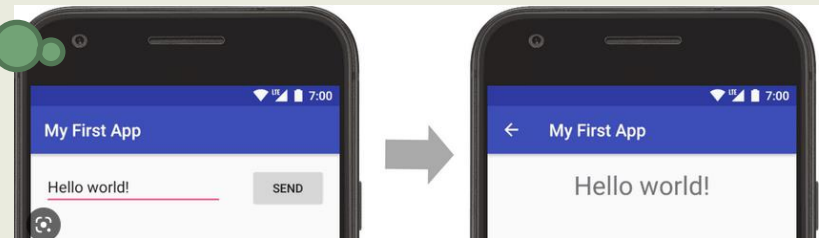
- **départ** : *ici*
- **Arrivée** : *SecondActivity*
- **Bagages** : la valeur 27

```
Intent intent = new Intent( getApplicationContext(), SecondActivity.class);  
intent.putExtra( getString(R.string.VALUE), value: 27);  
startActivity(intent);
```

- SecondActivity

- **Aller chercher le passager**
- **Récupérer son bagage**

```
Intent intent = getIntent();  
intent.getIntExtra( getString(R.string.VALUE), defaultValue: 0);
```



# CONTEXTE D'UTILISATION

Ça fonctionne avec les types  
primitifs + String + ??

→ MAIS pas avec une **Pizza**



- MainActivity
  - **départ** : ici
  - **Arrivée** : SecondActivity
  - **Bagages** : Une **Pizza**



```
class Pizza {  
    3 usages  
    private String name;  
    3 usages  
    private float price;  
    2 usages  
    private int picture;  
  
    9 usages  
    public Pizza(String name, float price, int picture) {  
        this.name = name;  
        this.price = price;  
        this.picture = picture;  
    }  
  
    public String getName() { return name; }  
    public float getPrice() { return price; }  
    public int getPicture() { return picture; }  
  
    @Override  
    public String toString() { return "nom='" + name + '\'' + ", prix=" + price; }  
}
```

# CONTEXTE D'UTILISATION

Ça fonctionne avec les types  
primitifs + String + ??

→ MAIS pas avec une **Pizza**

- MainActivity

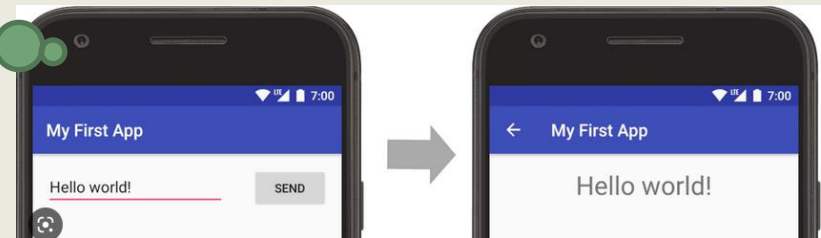
- **départ** : *ici*
- **Arrivée** : *SecondActivity*
- **Bagages** : Une **Pizza**

```
Intent intent = new Intent( getApplicationContext(), SecondActivity.class);  
intent.putExtra( getString(R.string.VALUE), value: 27);  
startActivity(intent);
```

- SecondActivity

- **Aller chercher le passager**
- **Récupérer son bagage**

```
Intent intent = getIntent();  
intent.getIntExtra( getString(R.string.VALUE), defaultValue: 0);
```



# CONTEXTE D'UTILISATION

```
public class Pizza implements Parcelable {  
    4 usages  
    private String name;  
    4 usages  
    private float price;  
    4 usages  
    private int picture;  
  
    public Pizza(String name, float price, int picture) {  
        this.name = name;  
        this.price = price;  
        this.picture = picture;  
    }  
  
    1 usage  
    protected Pizza(Parcel in) {  
        name = in.readString();  
        price = in.readFloat();  
        picture = in.readInt();  
    }  
}
```

→ MAIS pas avec  
une Pizza

→ ...sauf si elle est  
Parcelable !

# CONTEXTE D'UTILISATION

```
public static final Creator<Pizza> CREATOR = new Creator<Pizza>() {  
    @Override  
    public Pizza createFromParcel(Parcel in) { return new Pizza(in); }  
  
    @Override  
    public Pizza[] newArray(int size) { return new Pizza[size]; }  
};
```

```
public String getName() { return name; }  
public float getPrice() { return price; }  
public int getPicture(){ return picture; }
```

```
@Override  
public int describeContents() { return 0; }
```

```
@Override  
public void writeToParcel(@NonNull Parcel dest, int flags) {  
    dest.writeString(name);  
    dest.writeFloat(price);  
    dest.writeInt(picture);  
}
```

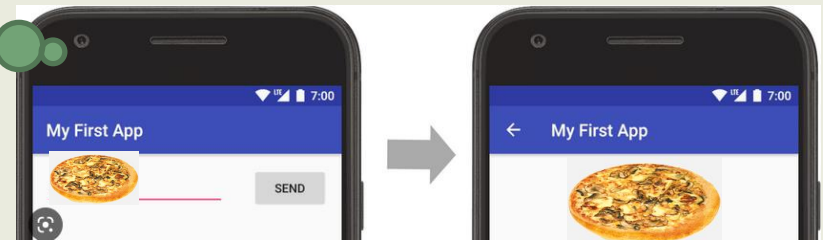
→ Redéfinir  
describeContents()  
n'est pas  
indispensable



# CONTEXTE D'UTILISATION

Ça fonctionne avec les types  
primitifs + String + ??

→ Et un **Parcelable**



- MainActivity
  - **départ** : ici
  - **Arrivée** : SecondActivity
  - **Bagages** : Une **Pizza**

```
Intent intent = new Intent( getApplicationContext(), SecondActivity.class);  
intent.putExtra( getString(R.string.VALUE), new Pizza( name: "Reine", price: 8.5f, R.drawable.pizza_reine));  
startActivity(intent);
```

- SecondActivity
  - **Aller chercher le passager**
  - **Récupérer son bagage**

```
Intent intent = getIntent();  
Pizza pizza = intent.getExtras().getParcelable(getString(R.string.VALUE));  
intent.getIntExtra( getString(R.string.VALUE), defaultValue: 0);
```