



Développement Graphique

Comportement

Programmation Séquentielle vs Évènementielle

Programmation Séquentielle:

Le code est exécuté dans un ordre prédéfini et immuable,

Programmation Évènementielle:

Le code est exécutée en fonction de l'apparition d'évènements provoqués par :

- l'utilisateur (action sur un bouton)
- Le système (durée d'un timer terminé)



Développement Graphique

Comportement

La Programmation Graphique est basée sur le fonctionnement évènementiel.

Ce fonctionnement nécessite de mettre en place des **listeners** (écouteurs d'évènements) associés à des **handle** (méthodes à exécuter) lorsque l'événement se produit.

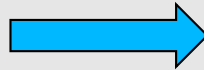
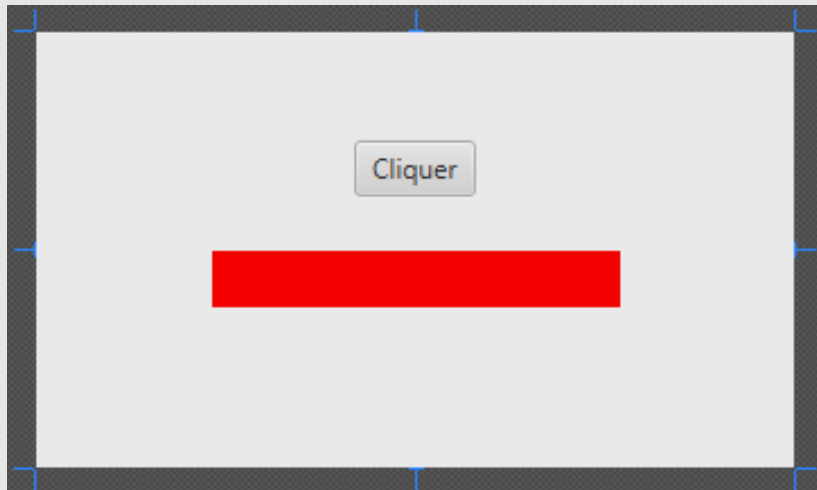


Développement Graphique

Comportement

Exemple:

Lorsqu'on clique sur le bouton, un message s'affiche dans le label



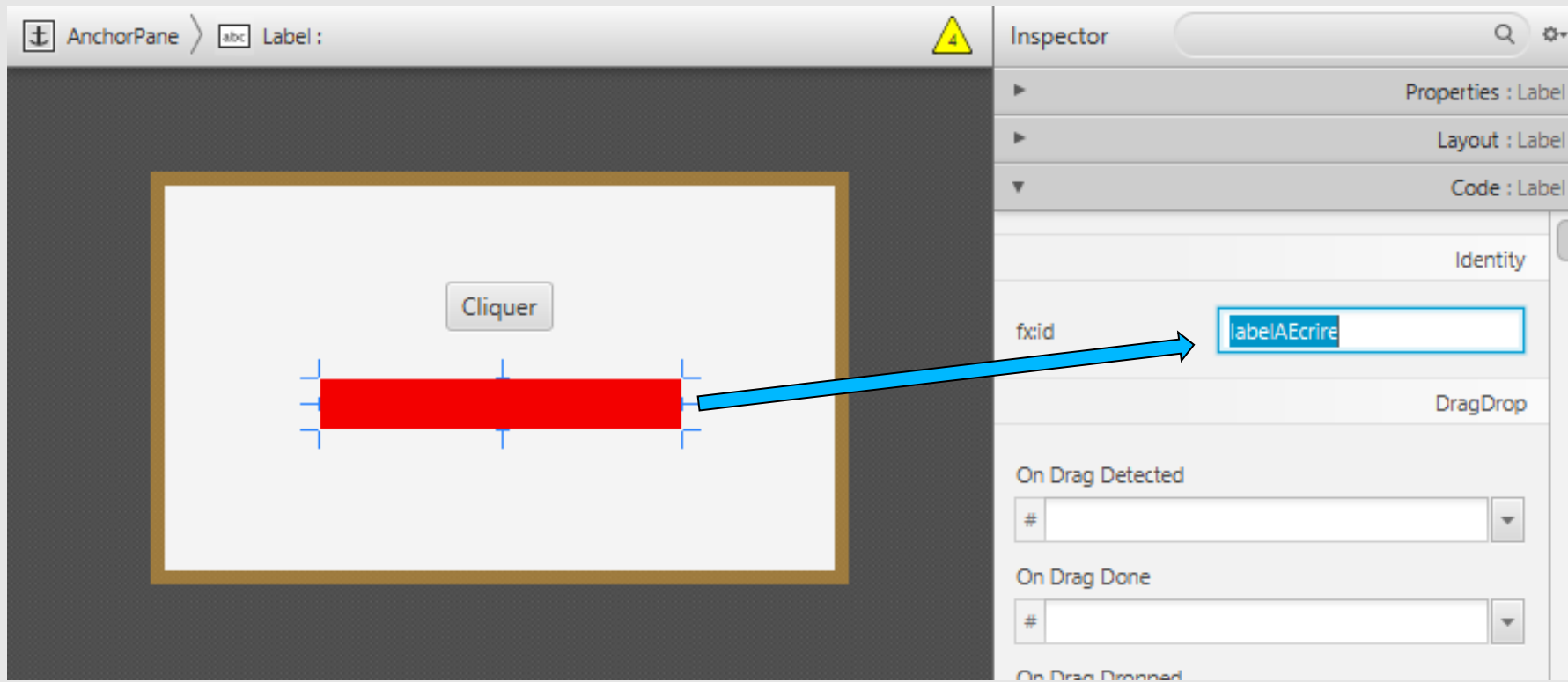


Développement Graphique

Comportement

Exemple: méthode

Associer au *label* un identifiant (*fx:id*) qui sera **l'identifiant de l'objet** associé au *label*.



Faire de même pour tous les composants dont on veut récupérer la référence.



Développement Graphique

Comportement

Exemple: méthode

Dans la classe contrôleur:

Déclarer l'objet *Label* en utilisant son identifiant *fx:id*

Chaque déclaration est précédée de l'annotation *@FXML*

```
public class FXMLDocumentController implements Initializable {  
  
    @FXML  
    private Label labelAEcrire;  
  
    @Override  
    public void initialize(URL url, ResourceBundle rb) {
```

Faire de même pour tous les composants dont on veut récupérer la référence.



Développement Graphique

Comportement

Exemple: méthode

Dans la classe contrôleur:

Créer la méthode contenant le code à exécuter lorsque l'événement « bouton cliqué » est détecté.

Syntaxe:

Annotation permettant de
faire le lien avec le FXML

@FXML

private void methodeAExecuter(ActionEvent event){

// code à exécuter

.....

}

Référence de l'évènement



Développement Graphique

Comportement

Exemple: méthode
classe contrôleur:

```
public class FXMLDocumentController implements Initializable {  
  
    @FXML  
    private Label labelAEcrire;  
  
    @FXML  
    private void methodeAExecuter(ActionEvent event) {  
        labelAEcrire.setText("Vous avez Cliqué");  
    }  
  
    @Override  
    public void initialize(URL url, ResourceBundle rb) {
```



Développement Graphique

Comportement

Exemple: méthode

Associer à l'évènement *On Action* (click du bouton) la méthode précédemment créée.

The screenshot displays a graphical user interface (GUI) design environment. On the left, a canvas shows a button labeled 'Cliquer' with a red rectangular area below it. A blue arrow points from the button to the 'On Action' event handler in the properties panel on the right. The properties panel is titled 'Properties : Button' and includes sections for 'Layout : Button' and 'Code : Button'. The 'Code : Button' section shows the 'On Action' event handler set to 'methodeAExecuter'.

Properties : Button	
Layout : Button	
Code : Button	
Identity	
fx:id	button_cliquer
Main	
On Action	
#	methodeAExecuter
DragDrop	
On Drag Detected	



Développement Graphique


Comportement

Exemple:

Conséquences sur le Fichier fxml

```
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.layout.AnchorPane?>

<AnchorPane id="AnchorPane" prefHeight="193.0" prefWidth="336.0" xmlns="http://javafx.com/javafx/8.0.6" xmlns:fx="http://javafx.com/fxml/1">
  <children>
    <Button fx:id="button_cliquer" alignment="CENTER" layoutX="141.0" layoutY="48.0" onAction="#methodeAExecuter" text="Cliquer" />
    <Label fx:id="labelAEcrire" layoutX="78.0" layoutY="97.0" prefHeight="25.0" prefWidth="181.0" style="-fx-background-color: red;" />
  </children>
</AnchorPane>
```



Un *listener* est créé sur l'évènement *Action* (click) du bouton, la méthode *MethodeAExecuter* lui est associée.



Développement Graphique

Comportement

Exemple: Autre méthode

Le *listener* sur l'évènement *Action* (click) du bouton peut être créé différemment dans la classe controleur:

```
public class FXMLDocumentController implements Initializable {
    @FXML
    private Label labelAEcrire;
    @FXML
    private Button button_cliquer;

    @Override
    public void initialize(URL url, ResourceBundle rb) {
        button_cliquer.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                labelAEcrire.setText("Vous avez Cliqué");
                System.out.println(event.getSource());
                System.out.println(event.getEventType());
            }
        });
    }
}
```