



UNIVERSITÀ DEGLI STUDI DELL'AQUILA

Laboratorio di Programmazione ad Oggetti

Ph.D. Juri Di Rocco
juri.dirocco@univaq.it
<https://jdirocco.github.io>





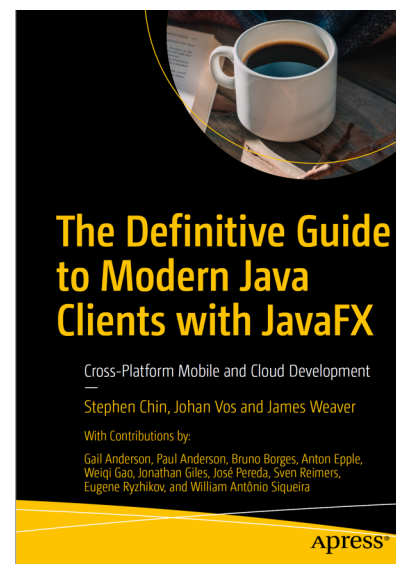
Sommario

- › Introduzione
- › Download ed installazione
- › Fondamenti di JavaFX
- › Esempio: HelloWorld
- › Elementi di controllo del layout
 - StackPane
 - AnchorPane
 - GridPane
 - BorderPane
 - VBox e HBox, ButtonBar
- › FXML
- › Scene Builder



Risorse

- › <https://openjfx.io/>
- › <http://tutorials.jenkov.com/javafx/index.html>
Buon tutorial in Inglese
- › The Definitive Guide to Modern Java Clients with JavaFX





Introduzione (1)

- › JavaFX permette la realizzazione di applicazioni con una interfaccia grafica (user interface - UI) molto accattivante in Java
- › Successore di Java Swing
- › Sfrutta appieno tutte la potenza dei processori grafici attuali GPU
- › Tramite FXML pemette di creare in modo semplice ed elegante le interface grafiche con uno stile di programmazione dichiarativo

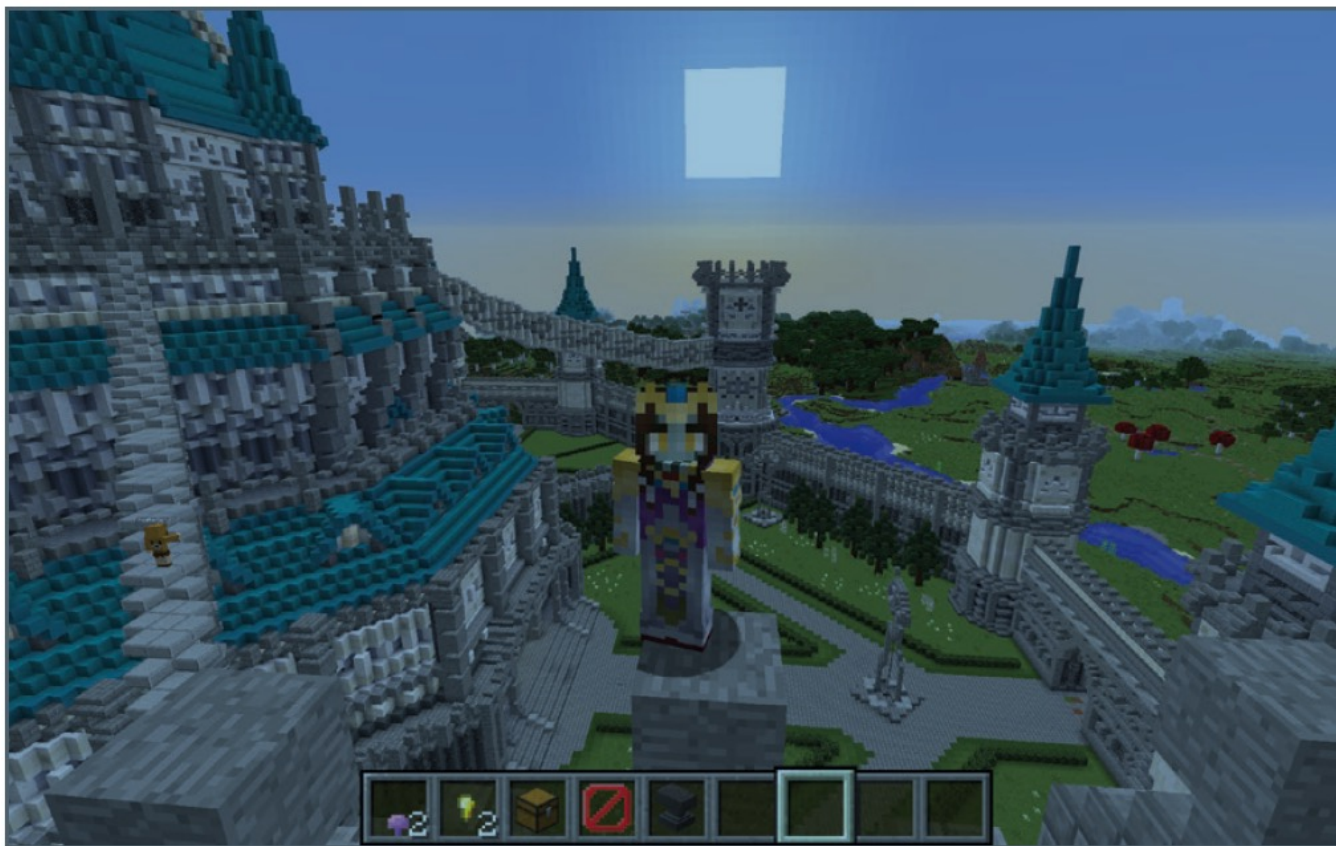


- [illegible]



Introduzione (3)

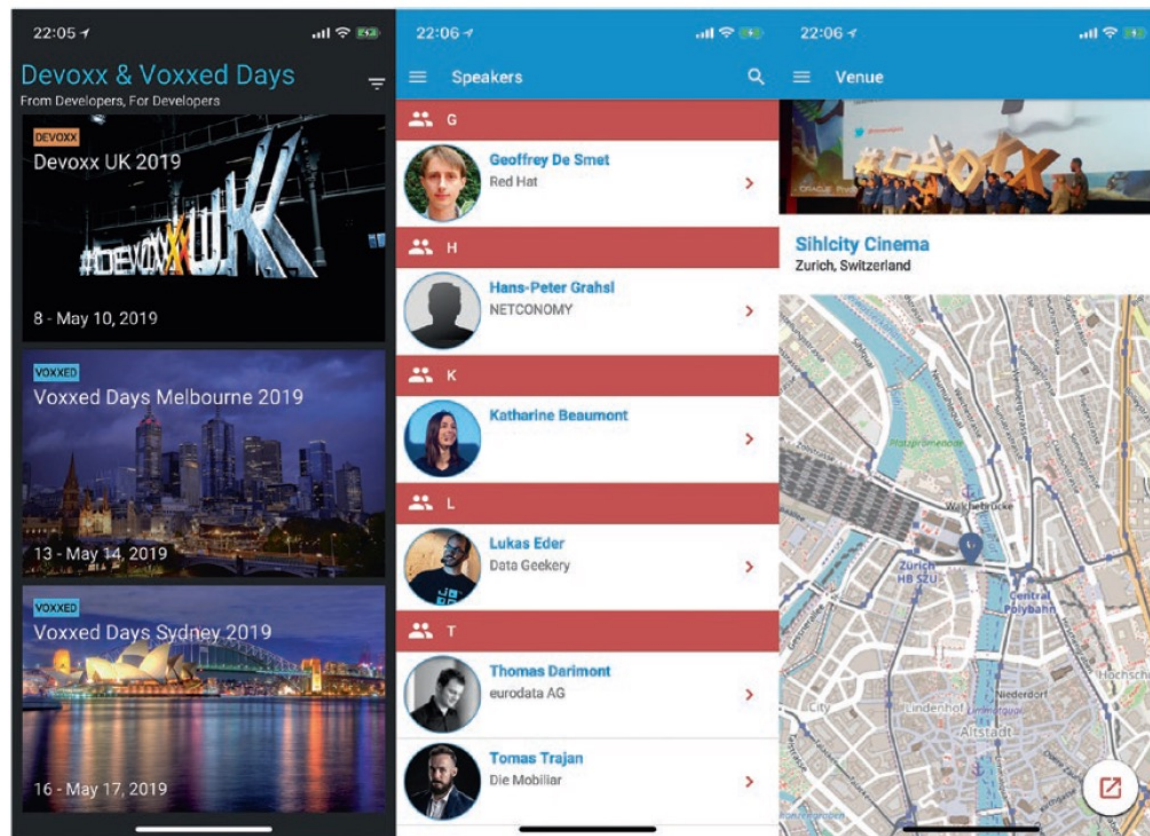
› Esempio: gaming minecraft





Introduzione (4)

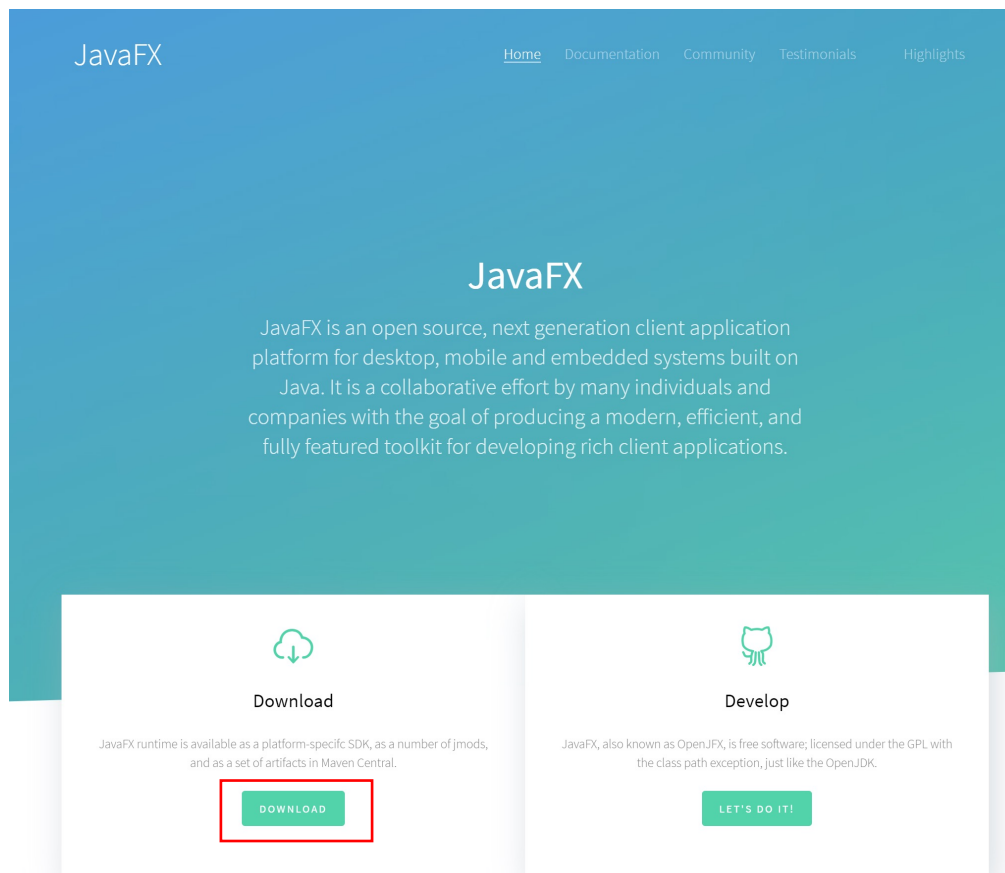
› Esempio: Mobile della conferenza Devovx UK 2019





Download (1)

› <https://openjfx.io/>


A screenshot of the JavaFX website homepage. The page has a blue header with the "JavaFX" logo on the left and navigation links "Home", "Documentation", "Community", "Testimonials", and "Highlights" on the right. The main content area has a teal background with the "JavaFX" title and a paragraph describing it as an open source, next generation client application platform for desktop, mobile and embedded systems built on Java. Below this, there are two white boxes. The left box is titled "Download" and features a cloud download icon, a paragraph about the runtime availability, and a green "DOWNLOAD" button which is highlighted with a red rectangle. The right box is titled "Develop" and features a GitHub icon, a paragraph about the software being free and licensed under the GPL, and a green "LET'S DO IT!" button.

JavaFX

[Home](#) [Documentation](#) [Community](#) [Testimonials](#) [Highlights](#)

JavaFX


JavaFX is an open source, next generation client application platform for desktop, mobile and embedded systems built on Java. It is a collaborative effort by many individuals and companies with the goal of producing a modern, efficient, and fully featured toolkit for developing rich client applications.



Download

JavaFX runtime is available as a platform-specific SDK, as a number of jmods, and as a set of artifacts in Maven Central.

[DOWNLOAD](#)



Develop

JavaFX, also known as OpenJFX, is free software; licensed under the GPL with the class path exception, just like the OpenJDK.

[LET'S DO IT!](#)



Download (2)

Latest Release

JavaFX 16 is the latest release of JavaFX. We will support it until the release of JavaFX 17.

The JavaFX 16 runtime is available as a platform-specific SDK, as a number of jmods, and as a set of artifacts in maven central.

The Release Notes for JavaFX 16 are available in the OpenJFX GitHub repository: [Release Notes](#).

This software is licensed under GPL v2 + Classpath (see <http://openjdk.java.net/legal/gplv2+ce.html>).

Product	Version	Platform	Download
JavaFX Windows x64 SDK	16	Windows x64	Download [SHA256]
JavaFX Windows x64 jmods	16	Windows x64	Download [SHA256]
JavaFX Windows x86 SDK	16	Windows x86	Download [SHA256]
JavaFX Windows x86 jmods	16	Windows x86	Download [SHA256]
JavaFX Mac OS X SDK	16	Mac	Download [SHA256]
JavaFX Mac OS X jmods	16	Mac	Download [SHA256]
JavaFX Linux SDK	16	Linux	Download [SHA256]
JavaFX Linux jmods	16	Linux	Download [SHA256]
JavaFX Documentation	16	Javadoc	Download [SHA256]





Download (3)


Organizza


Nuovo


java > javafx-sdk-16



Nome

 bin

 legal

 lib



Fondamenti di JavaFX (1)

- › Applicazione JavaFX è controllata dalla piattaforma JavaFX
- › Piattaforma JavaFX è un sistema di runtime che costruisce gli oggetti dell'applicazione e costruisce il thread dell'applicazione JavaFX
- › Per costruire una applicazione JavaFX è necessario estendere dalla classe `Application`
- › Sistema di runtime controlla il ciclo di vita dell'applicazione e invoca il metodo `start()`



Fondamenti di JavaFX (2)

- › JavaFX utilizza la metafora del teatro
 - Contenitore top-level è il palco (**Stage**) ed è costruito dalla piattaforma
 - Ambiente desktop lo stage è la finestra (**Window**)
 - Chiaramente la finestra varia a seconda del Sistema operativo
 - E' decorata con controlli che riducono, minimizzano e chiudono l'applicazione
- › E' possibile specializzare la classe `Application` per altri ambienti
- › Esempio il framework Gluon Mobile Application definisce la classe `MobileApplication` per la scrittura di applicazioni per dispositivi mobili



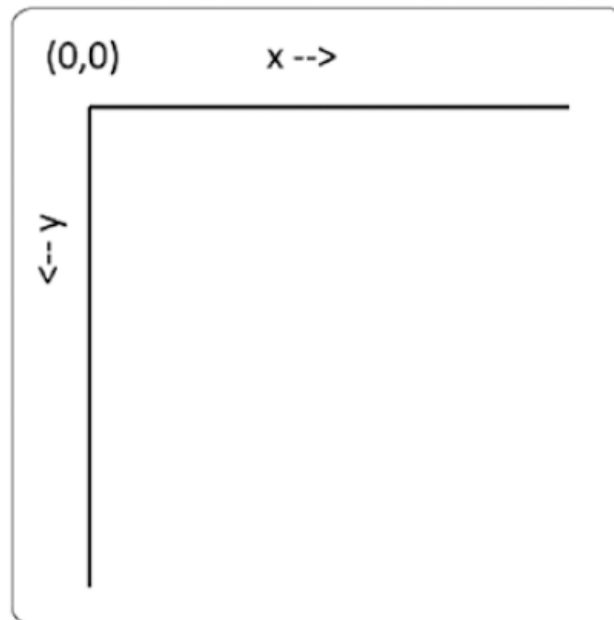
Fondamenti di JavaFX (3)

- › Il palco (**Stage**) contiene una scena (**Scene**)
- › Scena consiste di elementi JavaFX di cui uno è l'elemento **root**
- › root elemento **top** della scena e contiene il cosiddetto grafo della scena (**scene graph**)
- › Grafo è una struttura gerarchica di elementi detti **Nodi**
- › Un nodo ha esattamente un padre (**parent**) e può avere uno o più figli
- › Un nodo che non ha figli è detto **foglia**
- › Nodi devono essere aggiunti ad un grafo per poter essere visualizzati nella scena
- › Un nodo può essere aggiunto soltanto ad **una** scena



Fondamenti di JavaFX (4)

- › Vi sono delle regole e vincoli di layout quando una scena deve essere visualizzata
- › JavaFX utilizza un sistema di coordinate bi-dimensionale per gli elementi grafici 2D
- › L'asse-x va da sinistra a destra e l'asse-y dall'alto verso il basso





Fondamenti di JavaFX (5)

- › JavaFX supporta la grafica 3D e rappresenta la terza dimensione con l'asse-z che fornisce la profondità
- › JavaFX ha un sistema assoluto di coordinate che è in aggiunta al sistema locale delle coordinate relative al padre
- › In ogni caso l'origine del Sistema di coordinate è l'angolo in alto a sinistra del componente padre
- › Gli elementi di controllo del layout (**layout controls**) nascondono la complessità del posizionamento dei componenti all'interno della scena e gestiscono il posizionamento dei figli automaticamente
- › E' possibile **annidare** elementi di controllo del layout



Fondamenti di JavaFX (6)

- › JavaFX utilizza un modello **single-thread** per l'esecuzione
- › Applicazione costruisce e modifica lo stage e le sue scene all'interno del thread dell'applicazione
- › Anche la risposta agli eventi dell'utente, animazioni o cambiamenti al grafo della scena viene effettuato all'interno del thread dell'applicazione
- › L'esecuzione di task long-running (ovvero lunga computazione) dovrebbe avvenire in threads separati per mantenere l'UI responsive



Esempio (1)

```
public class HelloWorld extends Application {  
    public static void main(String[] args) {  
        launch(args);  
    }  
    @Override  
    public void start(Stage stage) throws Exception {  
        String javaVersion = System.getProperty("java.version");  
        String javafxVersion = System.getProperty("javafx.version");  
        Label l = new Label("Hello, JavaFX " + javafxVersion +  
                             ", running on Java " + javaVersion + ".");  
        Scene scene = new Scene(new StackPane(l), 640, 480);  
        stage.setScene(scene);  
        stage.show();  
    }  
}
```

Palco ovvero finestra dell'applicazione

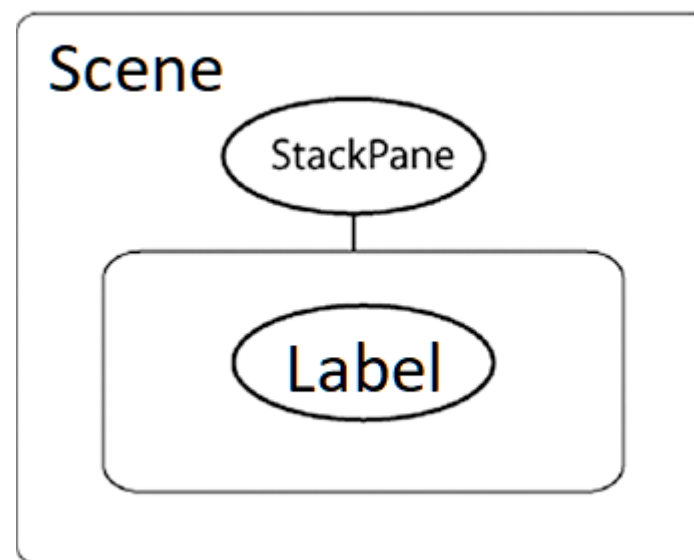
Creazione di una scena

Componente di root della scena

Figlio del componente di root

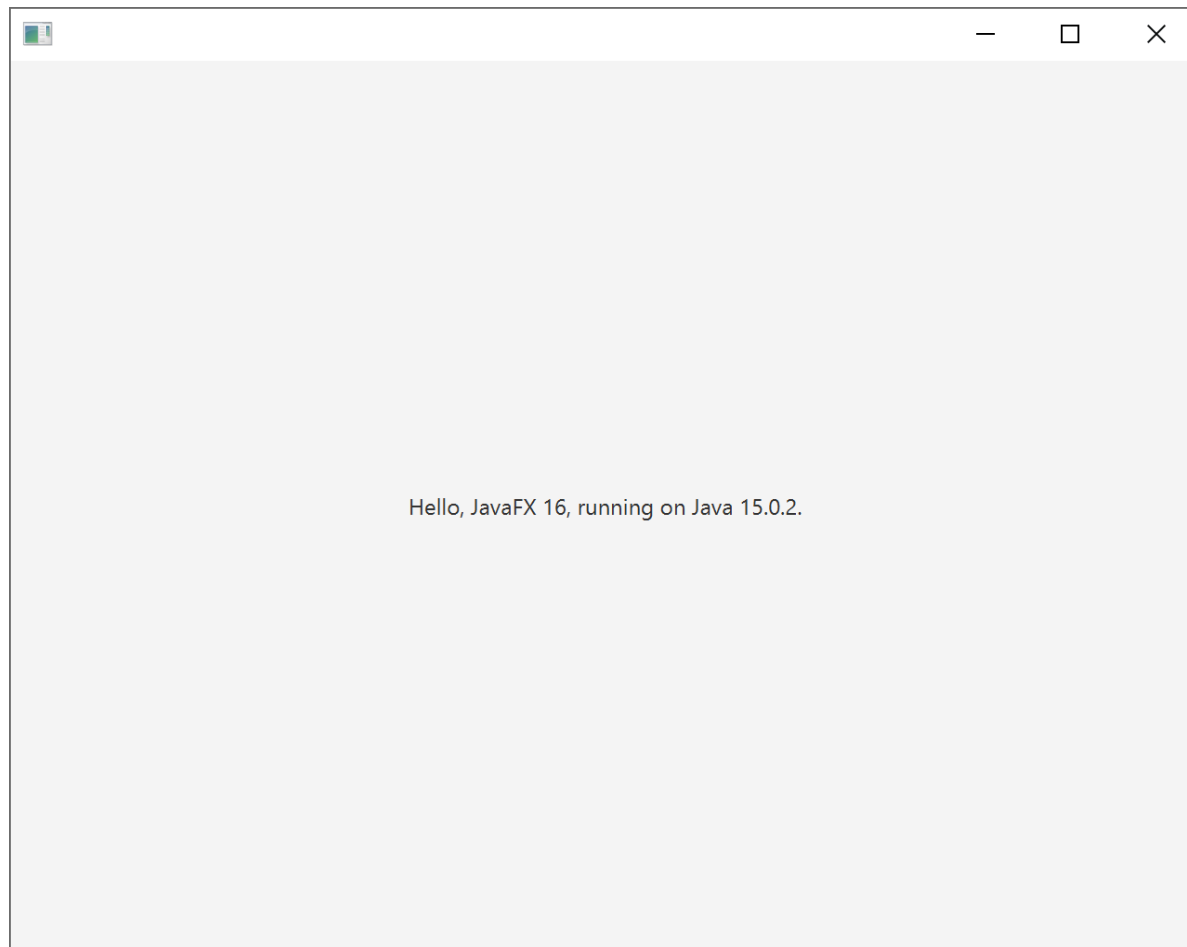


Esempio (2)





Esempio (3)





Elementi di controllo del layout

- › Per gestire nodi all'interno di una scena è necessario utilizzare degli elementi di controllo del layout (detti anche **layout controls**)
- › E' possibile annidare tali elementi utili durante il ridimensionamento della scena
- › E' possibile specificare diverse modalità di allineamento, margini, padding
- › Diversi modi per aggiungere nodi `getChildren().add()` e `getChildren().addAll()`



StackPane

- › Layout control che sovrappone i suoi figli da dietro a davanti nell'ordine in cui vengono aggiunti
- › Nell'esempio successivo l'ellisse appare prima del testo poiché è visualizzata dietro in quanto è stata aggiunta per prima
- › In caso opposto l'ellisse avrebbe oscurato il testo
- › Per default `StackPane` mette al centro tutti i suoi figli
- › E' possibile fornire un allineamento differente
- › Esempio

```
stackPane.setAlignment(text, Pos.BOTTOM_CENTER);
```



Esempio (1)

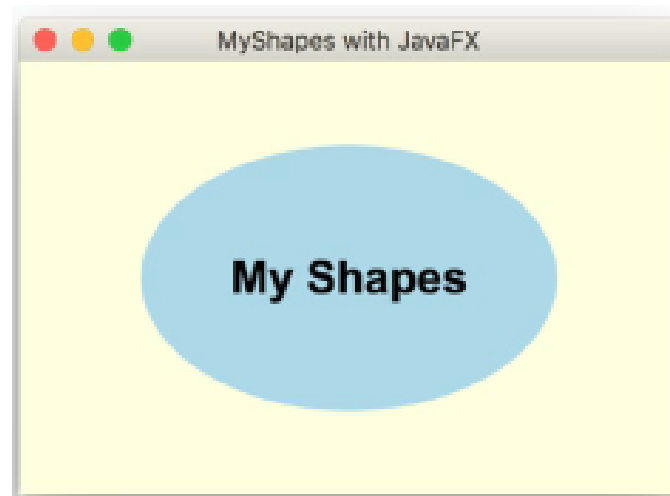
```
public class MyShapes extends Application {  
    @Override  
    public void start(Stage stage) throws Exception {  
        // Create an Ellipse and set fill color  
        Ellipse ellipse = new Ellipse(110, 70);  
        ellipse.setFill(Color.LIGHTBLUE);  
        // Create a Text shape with font and size  
        Text text = new Text("My Shapes");  
        text.setFont(new Font("Arial Bold", 24));  
        StackPane stackPane = new StackPane();  
        stackPane.getChildren().addAll(ellipse, text);  
        Scene scene = new Scene(stackPane, 350, 230, Color.LIGHTYELLOW);  
        stage.setTitle("MyShapes with JavaFX");  
        stage.setScene(scene);  
        stage.show();  
    }  
}
```




Esempio (2)

.....

```
public static void main(String[] args) {  
    launch(args);  
}  
}
```





AnchorPane

- › Gestisce i figli considerando dei punti preconfigurati anche quando viene ridimensionato
- › Si può specificare un offset
- › Tipicamente viene utilizzato come layout control top-level per controllare i margini

```
AnchorPane anchorPane = new AnchorPane();  
Label label = new Label("My Label");  
anchorPane.getChildren().add(label);  
AnchorPane.setLeftAnchor(label, 10.0);  
AnchorPane.setBottomAnchor(label, 10.0);
```



GridPane

- › I figli vengono posti in una griglia bi-dimensionale
- › I componenti possono coprire righe e/o colonne, ma la dimensione della riga è coerente per tutti i componenti in una data riga
- › Analogamente accade per una Colonna

```
GridPane gridPane = new GridPane();  
gridPane.add(new Label("Label1"), 0, 0);  
gridPane.add(new Label("Label2 is very long"), 1, 0, 2, 1);
```



BorderPane

- › Composto di un'area in **alto**, a **sinistra** al **centro**, a **destra** ed in **basso**
- › E' particolarmente utile per le applicazioni desktop dove in alto si può inserire una toolbar, in basso una status bar, al centro l'area di lavoro e due aree sia a sinistra che a destra
- › Ogni sezione può essere vuota
- › Allineamento di default al centro per l'area di centro e a sinistra per l'area top
- › E' possibile settare anche i margini

```
BorderPane borderPane = new BorderPane();  
Label colorLabel = new Label("Color: Lightblue");  
colorLabel.setFont(new Font("Verdana", 18));  
borderPane.setTop(colorLabel);  
Rectangle rectangle = new Rectangle(100, 50, Color.LIGHTBLUE);  
borderPane.setCenter(rectangle);  
BorderPane.setAlignment(colorLabel, Pos.CENTER);  
BorderPane.setMargin(colorLabel, new Insets(20,10,5,10));
```



Hbox e VBox, ButtonBar

- › Permettono di inserire i component figli con un orientamento orizzontale e verticale
- › Annidando un `HBox` dentro un `VBox`, o viceversa, si ottiene un effetto griglia
- › `ButtonBar` permette di inserire una riga di pulsanti della stessa dimensione in un contenitore orizzontale



FXML

- › E' possibile creare le scene (viste) utilizzando un approccio alternativo ovvero tramite FXML
- › FXML è una notazione markup basata su XML
- › Permette di descrivere e configurare il grafo della scena in modo dichiarativo
- › Supporta l'architettura Model-View-Controller (MVC) che fornisce una migliore struttura per applicazioni di grandi dimensioni
- › FXML riduce la scrittura di codice JavaFX



FXML

- › Si può disegnare la vista mediante **Scene Builder**
- › Chiaramente è possibile editare i file FXML tramite un qualsiasi editor di testi
- › All'interno del codice Java, JavaFX utilizza oggetto `FXMLLoader` che legge il file FXML e costruisce gli oggetti JavaFX che rappresentano la scena
- › Associato ad ogni file FXML (vista) vi è una classe **controller** che ha il compito di
 - Gestire gli eventi generati dall'utente
 - Aggiornare dinamicamente la vista
 - Invocare la logica di business di una specifica vista



Sviluppo rapido con Scene Builder

- › Tool che permette costruire rapidamente applicazioni moderne client
- › E' un tool WYSIWYG (what you see is what you get)
- › Fornisce una rappresentazione visuale dell'applicazione
- › Opera direttamente sul file FXML per specificare in modo dichiarativo l'interfaccia grafica senza avere un impatto sul codice Java



Scene Builder: Download ed installazione

<https://gluonhq.com/products/scene-builder/>



Integrated

Scene Builder works with the JavaFX ecosystem – official controls, community projects, and Gluon offerings including [Gluon Mobile](#), [Gluon Desktop](#), and [Gluon CloudLink](#).



Simple

Drag & Drop user interface design allows for rapid iteration. Separation of design and logic files allows for team members to quickly and easily focus on their specific layer of application development.



Supported

Scene Builder is free and open source, but is backed by Gluon. [Commercial support offerings](#) are available, including [training](#) and [custom consultancy services](#).

Download Scene Builder

Scene Builder **16.0.0** was released on **Mar 26, 2021**.

*You can use this Scene Builder version together with **Java 11 and higher**.*

Product	Platform	Download
Scene Builder	Windows Installer	Download
Scene Builder	Mac OS X dmg	Download
Scene Builder	Linux RPM	Download
Scene Builder	Linux Deb	Download
Scene Builder Kit info	Jar File	Download



Scene Builder
