

آموزش زبان برنامه نویسی جاوا

گرافیک در جاوا - پکیج Swing

جلسه دوم

کلاس JButton

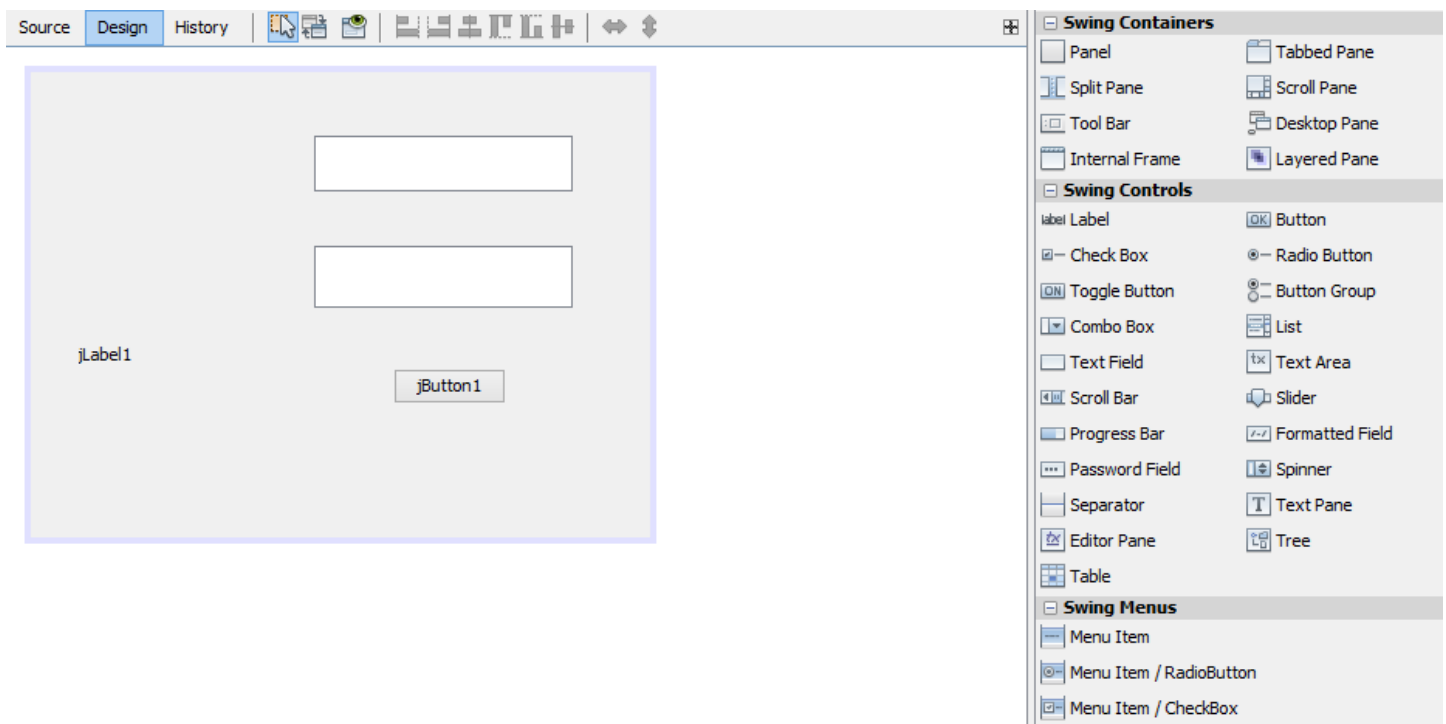
نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



قبل از پرداختن به این جلسه لازم است یک نکته را یادآورم شویم!

کار با گرافیک در جاوا در IDE (ویرایشگر) Netbeans ساده تر از IDE (ویرایشگر) Eclipse می باشد. در نتبیز شما تنها با Drag and drop (کشیدن و رها کردن) اجزای گرافیکی می توانید رابط گرافیکی کاربری را طراحی کنید. اما در این آموزش ما بصورت کد نویسی اجزای گرافیکی خود را طراحی می کنیم که خب نسبت به نتبیز یخورده کار بیشتری می برد. در تصویر (۱) محیط نتبیز را مشاهده می کنید که سمت راست آن اجزای گرافیکی قرار داده که با drag کردن (کشیدن) هر کدام و drop کردن (رها کردن) آنها در قسمت سمت چپ می توانید اجزای گرافیکی دلخواه خودتون رو ایجاد کنید.

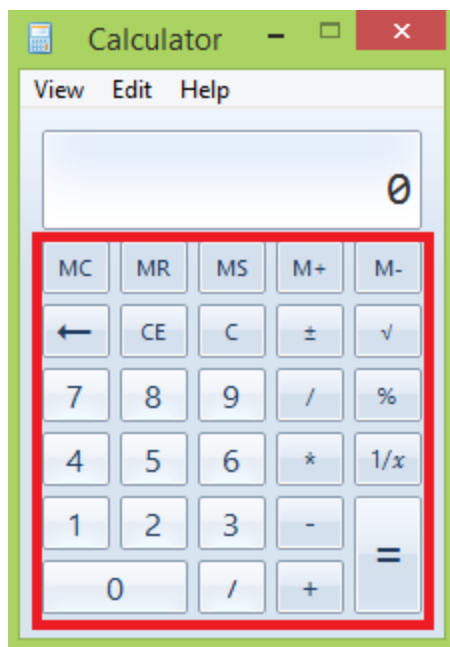


تصویر (۱)

خب شاید بگید این چکاره! کار خودمون داریم سخت تر می کنیم! چرا مستقیم سراغ netbeans نمی رویم؟! ما ابتدا کار با گرافیک در محیط eclipse را کار می کنیم چون منابع فارسی کمتری در این زمینه وجود دارد، در مرحله بعد به سراغ netbeans می رویم. البته ما سعی می کنیم این مفاهیم را طوری آموزش بدهیم که به آسانی قابل فهم باشد. یک نکته دیگه ای هم که هست شما می توانید با نصب پلاگین (plugin) مربوط به اجزای گرافیکی در ایکلیپس، به راحتی شبیه نتبیز در ایکلیپس اجزای گرافیکی خود را طراحی کنید.

## کلاس JButton:

یکی از اجزای گرافیکی یا component ما کلاس JButton یا همون دکمه می باشد. مثالی که از button برای درک بیشتر می توانیم بزنیم، دکمه های برنامه کاربردی ماشین حساب می باشد. تصویر (۲)



تصویر (۲)

- بخش قرمز مشخص شده همان دکمه های ماشین حساب یا button های ماشین حساب می باشند.

خب در این جلسه آموزشی قصد داریم یاد بگیریم که چطور این button ها را در جاوا طراحی کنیم. Button ها اجزای گرافیکی هستند که هنگام فشردن دستورات و عملیات خاصی را اجرا می کنند. مثلا در تصویر (۲) هنگام فشردن دکمه صفر عدد شماره صفر در صفحه نمایش ماشین حساب نمایش داده می شود یا با فشردن دکمه جمع (+) عملیات جمع زدن اجرا می شود و... پس تا اینجا یاد گرفتیم که :

۱. Button ها همان نقش دکمه در برنامه ها را برای ما بازی می کنند.
۲. هر button یا دکمه یک شکل ظاهری دارد مثلا در برنامه ماشین حساب دکمه های اعداد بصورت مربع نمایش داده شده اند و همچنین هر دکمه ( button ) یک دستور و عملیاتی خاص را اجرا می کند، مثلا دکمه شماره یک ماشین حساب عدد شماره یک را به ما می دهد و دکمه جمع (+) عملیات جمع زدن اعداد را برای ما اجرا می کند و...
۳. برای اجرای دستورات و عملیات مربوط به هر دکمه یا button باید button مربوطه فشرده شود.

خب ما برای این که یک دکمه (button) را در برنامه خود طراحی کنیم باید دو کار زیر را انجام دهیم:

۱. طراحی ظاهر دکمه که برای کاربر در برنامه نمایش داده می شود.

۲. طراحی دستورات و عملیات مربوط به دکمه مورد نظر

- به دستورات و عملیاتی که هر دکمه (button) بعد از فشردن اجرا می شود یک رویداد یا event می گویند. در ادامه چگونگی طراحی ظاهر و رویداد (event) یک button (دکمه) را بررسی می کنیم.
- ما برای ظاهر دکمه (button) خود می توانیم تصویر نیز قرار دهیم که به آن آیکون می گویند. در تصویر (۳) در برنامه تلگرام چهار دکمه را با رنگ قرمز مشخص کرده ایم که به این دکمه ها یا button ها تصویر یا آیکون اضافه شده است.



تصویر (۳)

- در همین آموزش روش تصویر گذاری بر روی دکمه ها (button) را بررسی می کنیم.

## نحوه استفاده از کلاس JButton در برنامه خود:

برای استفاده از کلاس JButton در برنامه مون باید پکیج زیر را در کلاس خود import کنیم:

```
import javax.swing.JButton;
```

```
package swing_javalike;
import javax.swing.JButton;
public class Test {

}
```

- بعد از این کار حالا مجاز هستید از کلاس `JButton` در کلاس خود شی ایجاد کنید، و از متدهای آن استفاده کنید.

**نکته:** کلاس `JButton` کلاس `AbstractButton` را به ارث برده است و کلاس `AbstractButton` اینترفیس `Accessible` را در خود `implements` (پیاده سازی) کرده است. به دستور زیر نگاه کنید:

```
public class JButton extends AbstractButton implements Accessible
```

یا به گونه ای دیگر:

```
public class JButton
extends AbstractButton
implements Accessible
```

اصلا نگران نشوید!! این نکته رو کامل فراموش کنید! این رو گفتم که کلاس `JButton` متدها و ویژگی های خود را کلاس `AbstractButton` گرفته است و متدهای اینترفیس `Accessible` در کلاس `AbstractButton` پیاده سازی شده است همین صرفا جهت اطلاع بود 😊

- متداول ترین سازنده های (Constructors) کلاس `JButton` که از شون استفاده می کنیم در زیر آورده ایم:

سازنده (Constructor)	توصیف
<code>JButton()</code>	برای ایجاد یک دکمه ( <code>button</code> ) بدون متن یا آیکون
<code>JButton(String s)</code>	برای ایجاد یک دکمه ( <code>button</code> ) با متن دلخواه
<code>JButton(Icon i)</code>	برای ایجاد یک دکمه ( <code>button</code> ) با شی از یک آیکون
<code>JButton(String text, Icon icon)</code>	برای ایجاد یک دکمه ( <code>button</code> ) که دارای متن و آیکون می باشد

- این سازنده ها را در همین جلسه آموزشی در قالب مثال بررسی می کنیم.

## متدهای کلاس `JButton`:

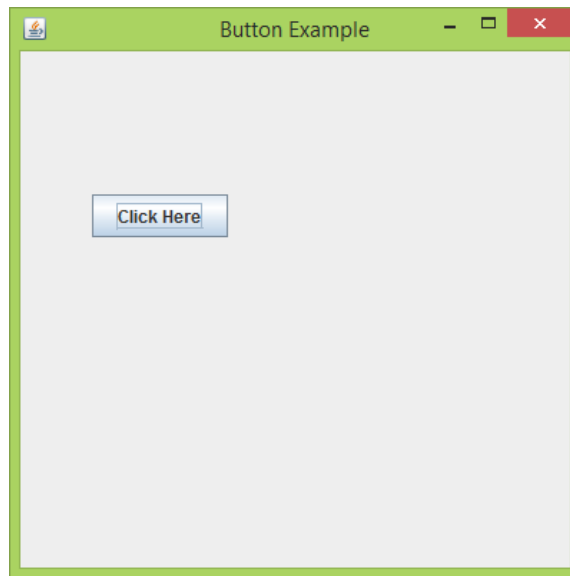
متد	توصیف
<code>void setText(String s)</code>	این متد برای قرار دادن متد روی دکمه ( <code>button</code> ) استفاده می شود

<code>String getText()</code>	این متد برای برگرداندن متن روی <code>button</code> استفاده می شود
<code>void setEnabled(boolean b)</code>	این متد برای نمایش یا عدم نمایش <code>button</code> در برنامه استفاده می شود. اگر مقدار پارامتر این متد <code>true</code> باشد <code>button</code> نمایش داده می شود و در صورت <code>false</code> بودن <code>button</code> نمایش داده نمی شود
<code>void setIcon(Icon b)</code>	این متد برای قرار دادن آیکون روی <code>button</code> استفاده می شود
<code>Icon getIcon()</code>	این متد آیکون روی <code>button</code> را برای ما بر می گرداند
<code>void addActionListener(ActionListener a)</code>	برای افزودن یک شی از اینترفیس <code>ActionListener</code> از این متد استفاده می شود. چرا و چگونگی استفاده از متد <code>addActionListener</code> و اینترفیس <code>ActionListener</code> را در ادامه بررسی می کنیم

مثال از ساخت یک دکمه (button) در جاوا:

```
package swing_javalike;
import javax.swing.*;
public class ButtonExample {
public static void main(String[] args) {
    JFrame f=new JFrame("Button Example");
    JButton b=new JButton("Click Here");
    b.setBounds(50,100,95,30);
    f.add(b);
    f.setSize(400,400);
    f.setLayout(null);
    f.setVisible(true);
}
}
```

خروجی: بعد از کامپایل و اجرای برنامه خروجی بصورت زیر خواهد بود. تصویر (۴)



تصویر (۴)

- در این مثال قصد داریم یک `button` در برنامه خود طراحی کنیم. ابتدا پکیج زیر را در کلاس خود `import` می کنیم:

```
import javax.swing.*;
```

- در گام بعدی نیاز به یک شاسی یا `frame` داریم که `button` ما بر روی آن قرار گیرد. برای این کار از کلاس `JFrame` شی ایجاد کرده کرده و متنی را جایگزین پارامتر سازنده آن می کنیم:

```
JFrame f=new JFrame("Button Example");
```

- در گام بعد برای داشتن یک `Button` یا دکمه از کلاس `Jbutton` شی ایجاد کرده و متنی را جایگزین پارامتر سازنده آن می کنیم:

```
JButton b=new JButton("Click Here");
```

- با استفاده از شی ایجاد شده از کلاس `JButton` که `b` نام دارد، متد `setBounds` را صدا میزنیم. این متد برای تنظیم کردن محل قرار گیری (مختصات) و ابعاد (عرض و ارتفاع) `button` در `frame` استفاده می شود. در کد زیر مختصات `button` ما برابر است با `x=50` و `y=100` و عرض دکمه ما برابر ۹۵ و ارتفاع آن برابر ۳۰ می باشد.

```
b.setBounds(50,100,95,30);
```

- از طریق شی `f` که از نوع کلاس `JFrame` هستش متد `add` را برای افزودن دکمه (`button`) به `frame` صدا می زنیم:

```
f.add(b);
```

- برای تعیین اندازه **frame** خود از دستور زیر استفاده می کنیم:

```
f.setSize(400,400);
```

- چون قصد نداریم از طرح بندی خاصی استفاده کنیم مقدار پارامتر متد زیر را **null** قرار می دهیم. طرح بندی یا **Layout** برای مدیریت محل قرارگرفتن اجزای گرافیکی در فریم استفاده می شود.

```
f.setLayout(null);
```

- در پایان برای نمایش **frame** در برنامه خود باید مقدار پارامتر متد زیر را **true** کنیم:

```
f.setVisible(true);
```

- دوستان برای استفاده از این دستورات ثابت در گرافیک که تا اینجا مثال زدیم اصلا نیاز به حفظ کردن نیست!! تنها یادگیری که چه موقع نیاز به این مفاهیم دارید ، و از آنها استفاده کنید! یادتون هم رفت سرچ کنید!! هیچ وقت اینارو حفظ نکنید که خسته میشید 😊

❖ خب ما تا اینجا با روش طراحی ظاهر دکمه یا **button** خود آشنا شدیم. بخش دیگر **button** که اجرای دستورات و عملیات مربوط به فشردن دکمه یا **button** می شود را در ادامه بررسی می کنیم.

❖ در مثال قبل بعد از اجرای برنامه در **frame** خود شاهد یک **button** بودیم ، اما با فشردن یا کلیک کردن بر روی **button** هیچ اتفاقی نمی افتاد. چرا؟ همان طور که از قبل هم گفتیم هر **button** علاوه بر شکل ظاهر کاربری خود ، دستورات و عملیاتی را نیز دارد که هنگام کلیک کردن بر روی آن اجرا می شود. حالا در ادامه قصد داریم کاری کنیم که وقتی بر روی یک **button** کلیک می شود دستور خاص و مربوط به آن اجرا شود.

## اجرای دستور یک **button** بعد از فشردن شدن

### اینترفیس **ActionListener**:

برای اجرای دستوراتی که به یک **button** نسبت می دهیم از این اینترفیس کمک می گیریم! خب چگونه؟! اینترفیس **ActionListener** یک متد بصورت زیر در بدنه خود دارد:



### `void actionPerformed(ActionEvent e)`

- دستورات و عملیاتی که قصد داریم یک دکمه (`button`) بعد از فشردن شدن (کلیک کردن) اجرا کند درون این متد قرار می گیرد.
- حتما سوال براتون پیش اومده چطور متد `actionPerformed` از اینترفیس `ActionListener` را به `button` خود ارتباط دهیم!! صبر داشته باشید 😊

متد `void addActionListener(ActionListener a)` رو که یادتون هست! در همین جلسه گفتیم که این متد برای افزودن یک شی از اینترفیس `ActionListener` به `button` خود از آن استفاده می کنیم. پس راه ارتباطی `button` ما با اینترفیس `ActionListener` استفاده از متد `addActionListener` می باشد. به شبه کد زیر دقت کنید:

```

JButton b = new JButton("Click Here");

b.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {

    }

});

```

ما در کد بالا قصد داریم یک `button` ایجاد کنیم و سپس دستورات و عملیات مورد نظر را به `button` خود نسبت دهیم به گونه ای که وقتی بر روی `button` کلیک شود دستورات مربوطه اجرا شود. برای این کار:

۱. ابتدا برای ایجاد دکمه یا `button` خود از کلاس `JButton` شی ایجاد می کنیم:

```
JButton b = new JButton("Click Here");
```

۲. از طریق شی ایجاد شده از کلاس `JButton` متد زیر را صدا می زنیم:

```
b.addActionListener(ActionListener a)
```

۳. بجای پارامتر متد `addActionListener` یک شی از اینترفیس `ActionListener` مستقیم ایجاد می کنیم:

```
b.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
    }
});
```

- وقتی مستقیم از اینترفیس `ActionListener` شی ایجاد می کنیم متد درون بدنه آن نیز بصورت خودکار پیاده سازی می شود. و در کل شکل شی سازی از اینترفیس `ActionListener` بصورت کد بالا می باشد.

۴. حالا دستورات و عملیاتی که قصد داریم به `button` مورد نظر نسبت دهیم را درون متد `actionPerformed` قرار می دهیم:

```
b.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.out.println("salam");
    }
});
```

- همان طور که در کد بالا مشاهده می کنید دستور چاپ پیام "salam" در محیط کنسول را در متد `actionPerformed` قرار داده ایم، حال با کلیک کردن یا فشردن دکمه `button` ما دستور چاپ پیام "salam" اجرا می شود.
- میدونم الان یکم گیج شدید! سخت نیست این مفهوم فقط یکم در هم بر هم شده 😊 درستش می کنیم!

❖ عملکرد کد بالا که باعث میشه با فشردن یا کلیک کردن روی دکمه `button` دستوری خاص اجرا شود بصورت زیر هستش:

وقتی ما دکمه `button` را فشار می دهیم یک رویداد (`event`) رخ می دهد که توسط متد `actionPerformed` درون اینترفیس `ActionListener` دریافت می شود و دستورات درون آن اجرا می شود. پل ارتباطی میان `button` ما و متد `actionPerformed` واقع در اینترفیس `ActionListener` متد `addActionListener` می باشد. یعنی تمام رویداد هایی که با فشردن یک دکمه `button` رخ میدهد از طریق راه ارتباطی `addActionListener` صورت می گیرد. این راه ارتباطی میان `button` ما و اینترفیس `ActionListener` می باشد.

در کل قالب دستوری که موجب اجرا شدن عملیات مربوط به یک `button` می شود بصورت زیر است:

```
b.addActionListener(new ActionListener() {  
  
    public void actionPerformed(ActionEvent e) {  
  
        System.out.println("salam");  
  
    }  
});
```

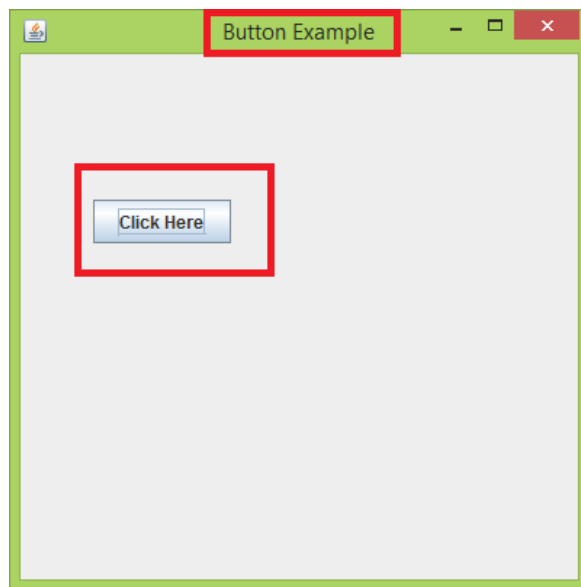
```
addActionListener(new ActionListener() {  
  
    public void actionPerformed(ActionEvent e) {  
  
        محل قرار گرفتن دستورات و عملیات مربوط به باتن یا دکمه ما  
  
    }  
});
```

مثال:

```
package swing_javalike;  
  
import java.awt.event.*;  
import javax.swing.*;  
  
public class ButtonExample2 {  
    public static void main(String[] args) {  
        JFrame f = new JFrame("Button Example");  
  
        JButton b = new JButton("Click Here");  
        b.setBounds(50, 100, 95, 30);  
        b.addActionListener(new ActionListener() {  
            public void actionPerformed(ActionEvent e) {  
                f.setTitle("Welcome to @Javalike");  
            }  
        });  
        f.add(b);  
    }  
}
```

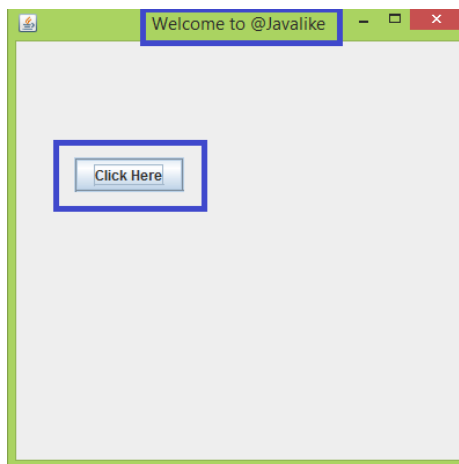
```
f.setSize(400, 400);  
f.setLayout(null);  
f.setVisible(true);  
}  
}
```

خروجی: تصویر (۵)



تصویر (۵)

- دکمه button و عنوان frame برنامه رو با رنگ قرمز مشخص کرده ایم، کار این برنامه به این صورت است که وقتی بر روی دکمه باتن "Click Here" کلیک می کنیم عنوان frame تغییر می کند، بعد فشردن یا کلیک کردن روی دکمه button خود برنامه بصورت زیر تغییر می کند. تصویر (۶)



تصویر (۶)

- عنوان **frame** ما که تغییر کرده است را با رنگ ابی مشخص کرده ایم. وقتی که بر روی دکمه **button** کلیک می کنیم یک رخداد ایجاد می شود که نتیجه آن اجرای دستور باتن ما که درون متد **actionPerformed** قرار دارد می باشد.

```
JFrame f = new JFrame("Button Example");

    JButton b = new JButton("Click Here");
    b.setBounds(50, 100, 95, 30);
```

- ایجاد یک شی از کلاس **JFrame** و مقداردهی به پارامتر سازنده آن. پارامتر سازنده آن عنوان **frame** ما را تشکیل می دهد.
- ایجاد یک شی از کلاس **JButton** و مقداردهی پارامتر سازنده آن. پارامتر سازنده آن نام دکمه **button** ما را تشکیل می دهد.
- با استفاده از متد **setBounds** مختصات و ابعاد دکمه **button** خود را مشخص می کنیم.

```
b.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        f.setTitle("Welcome to @Javalike");
    }
});
```

- در کد بالا ما یک رویداد را برای دکمه **button** خود ایجاد کرده ایم. به این صورت که با فشردن دکمه **button** دستورات مربوطه که برایش در متد **actionPerformed** تعیین کرده ایم اجرا می شود.
- برای این کار با استفاده از شی ایجاد شده از کلاس **JButton** خود متد **addActionListener** را صدا زده ایم.
- بعد بجای پارامتر متد **addActionListener** مستقیم از اینترفیس **ActionListener** شی ساخته ایم.
- وقتی از اینترفیس **ActionListener** شی ایجاد می کنیم بصورت خودکار باید متد درون بدنه این اینترفیس پیاده سازی شود.
- متد **actionPerformed** در بدنه اینترفیس **ActionListener** قرار دارد و به عنوان پارامتر یک رویداد را دریافت می کند، این رویداد زمانی رخ میدهد که بر روی دکمه **button** کلیک شود.
- بعد از رخ دادن رویداد یا همون کلیک بر روی دکمه **button** ما دستورات درون متد **actionPerformed** که همان عملیات دکمه **button** ما هستند اجرا می شود.

```
f.setTitle("Welcome to @Javalike");
```

• متد `setTitle` برای جایگزین کردن متن عنوان `frame` استفاده می شود.

چطور بود؟! امیدوارم که توانسته باشم مفهوم این مبحث رو روشن بیان کرده باشم 😊

## قرار دادن تصویر یا آیکون بر روی دکمه `button`

### کلاس `ImageIcon` :

برای قرار دادن یک تصویر یا آیکون روی دکمه `button` خود از این کلاس استفاده می کنیم.

### سازنده های پر کاربرد کلاس `ImageIcon` :

#### `ImageIcon(String filename)`

هنگام شی سازی از کلاس `ImageIcon` می توانیم از این سازنده بعد از کلمه کلیدی `new` استفاده کنیم.

```
ImageIcon im=new ImageIcon(String filename);
```

آدرس فایل تصویری که قراره بر روی دکمه `button` ما قرار گیرد جایگزین پارامتر این سازنده می کنیم. برای مثال من آدرس تصویر خود را در پوشه پروژه رو جایگزین این پارامتر کرده ام:

```
ImageIcon im=new ImageIcon("icon.png");
```

همان طور که ابتدای این جلسه آموزشی گفتیم ، کلاس `JButton` متدهای گوناگونی داره که پرکاربردترین آنها را بررسی کردیم. یکی از این متدها بصورت زیر است:

```
void setIcon(Icon b)
```

ما می توانیم از طریق شی ایجاد شده از کلاس `JButton` این متد را صدا بزنیم و شی `im` که از نوع کلاس `ImageIcon` را جایگزین پارامتر این متد کنیم:

```
JButton b = new JButton();
b.setIcon(im);
```

با این کار پروسه اضافه کردن تصویر یا آیکون به دکمه `button` ما انجام می شود.

مثال:

```
package swing_javalike;

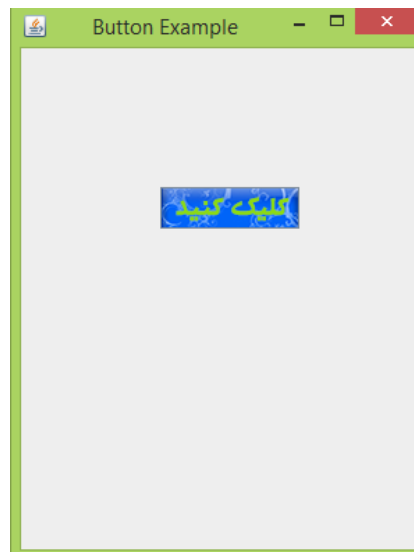
import javax.swing.*;

public class ButtonExample {
    ButtonExample() {
        JFrame f = new JFrame("Button Example");
        ImageIcon im=new ImageIcon("icon.png");

        JButton b = new JButton();
        b.setIcon(im);
        b.setBounds(100, 100, 100, 30);
        f.add(b);
        f.setSize(300, 400);
        f.setLayout(null);
        f.setVisible(true);
    }

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

خروجی: تصویر (۷)



تصویر (۷)

- همان طور که در تصویر (۷) مشاهده می کنید ، تصویر ما که نقش یک آیکون را بازی می کند روی دکمه `button` ما قرار گرفته است.

```
ImageIcon im=new ImageIcon("icon.png");

JButton b = new JButton();
b.setIcon(im);
```

- همان طور که در کد بالا مشاهده می کنید برای درج تصویر یا آیکون روی دکمه `button` ابتدا از کلاس `ImageIcon` شی ایجاد کردیم سپس آدرس تصویر در کامپیوتر را ( در اینجا آدرس پوشه محل پروژه های ما می باشد) را جایگزین پارامتر سازنده آن کرده ایم. سپس از کلاس `JButton` شی ایجاد کرده و از طریق شی ایجاد شده متد `setIcon` را صدا زده و به عنوان پارامتر شی `im` که از نوع کلاس `ImageIcon` هستش قرار داده ایم.
- ما می توانیم مستقیم و بدون صدا زدن متد هنگام شی سازی از کلاس `JButton` بر روی آن تصویر یا آیکون درج کنیم:

مثال:

```
package swing_javalike;

import javax.swing.*;

public class ButtonExample {
    ButtonExample() {
        JFrame f = new JFrame("Button Example");
        JButton b = new JButton(new ImageIcon("icon.png"));

        b.setBounds(100, 100, 100, 30);
        f.add(b);
        f.setSize(300, 400);
        f.setLayout(null);
        f.setVisible(true);
    }

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



خروجی: تصویر (۸)



تصویر (۸)

- همان طور که مشاهده میکنید در مثال بالا ما مستقیم و بدون استفاده از متدی آیکون خود را روی دکمه `button` درج کردیم.

```
JButton b = new JButton(new ImageIcon("icon.png"));
```

- در شبه کد بالا ما مستقیم از کلاس `ImageIcon` شی ایجاد کرده و جایگزین پارامتر سازنده کلاس `JButton` کرده ایم.

آموزش این جلسه چطور بود؟ آیا واضح و روشن بیان شد؟ امیدوارم که مورد پسند شما واقع شده باشد، نظرات خودتون رو در مورد این جلسه به ای دی تلگرام زیر ارسال کنید.

[@rzutab](https://t.me/rzutab)

همچنین می توانید نظرات خودتون رو به ایمیل زیر ارسال کنید:

[rahman.zarie92@gmail.com](mailto:rahman.zarie92@gmail.com)

ما بعد از پایان کار با گرافیک در جاوا آموزش ساخت بازی دوبعدی در جاوا را خواهیم داشت، سعی بر این داریم گام به گام به مفاهیم جاوا بپردازیم پس با ما همراه باشید قطعاً حمایت شما موجب انرژی بیشتر خواهد شد 😊

پیروز و موفق باشید

سایت آموزش زبان جاوا به زبان ساده، آسان و شیرین!!!

[www.JAVAPRO.ir](http://www.JAVAPRO.ir)

آموزش جاوا SE را با تجربه شخصی و به زبان خودمونی یاد بگیرید!!!!

# بازدید از کانال

# بازدید از سایت

هر روز مفاهیم و مثال های جدید به سایت اضافه می شود برای اطلاع از مطالب جدید روی سایت عضو کانال شوید.