



תפריט נשלף

Options menu



בגרסאות 1-2 של Android

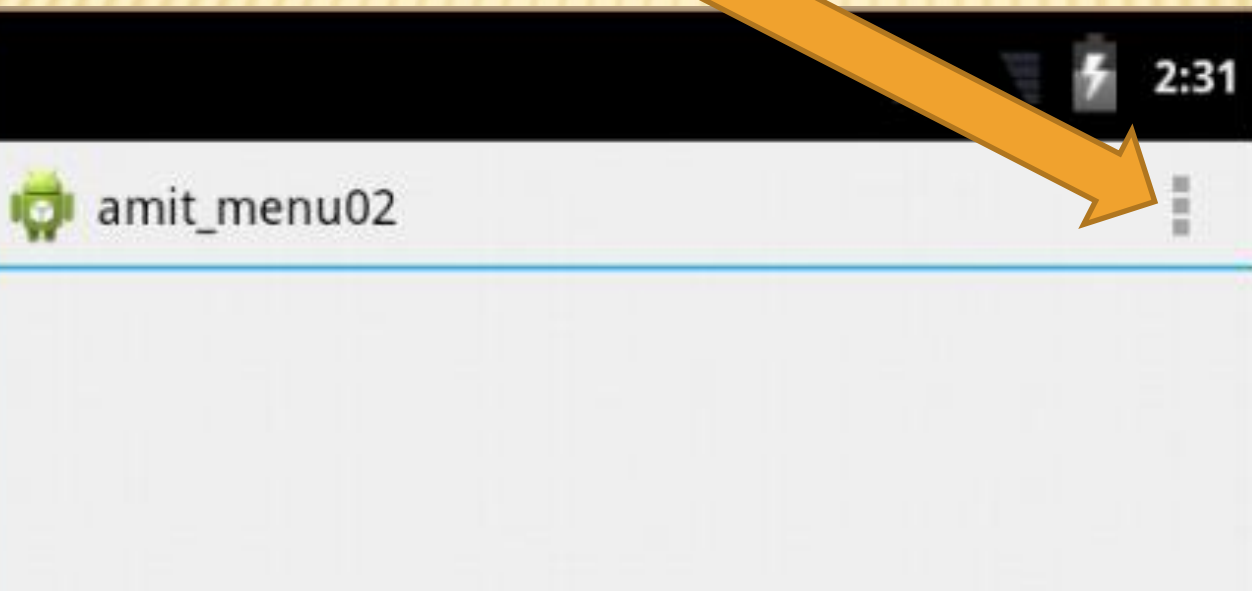
לקבלת תפריט

היה לחצן על מכשיר עצמו





בגרסאות 3-4 של Android לקבלת תפריט משמש סימן על פני המסך





פעולה ליצירת התפריט הנשלף (נוצרת באופן אוטומטי)

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
  
    // Inflate the menu; this adds items to the ac  
    getMenuInflater().inflate(R.menu.main, menu);  
  
    return true;  
}
```

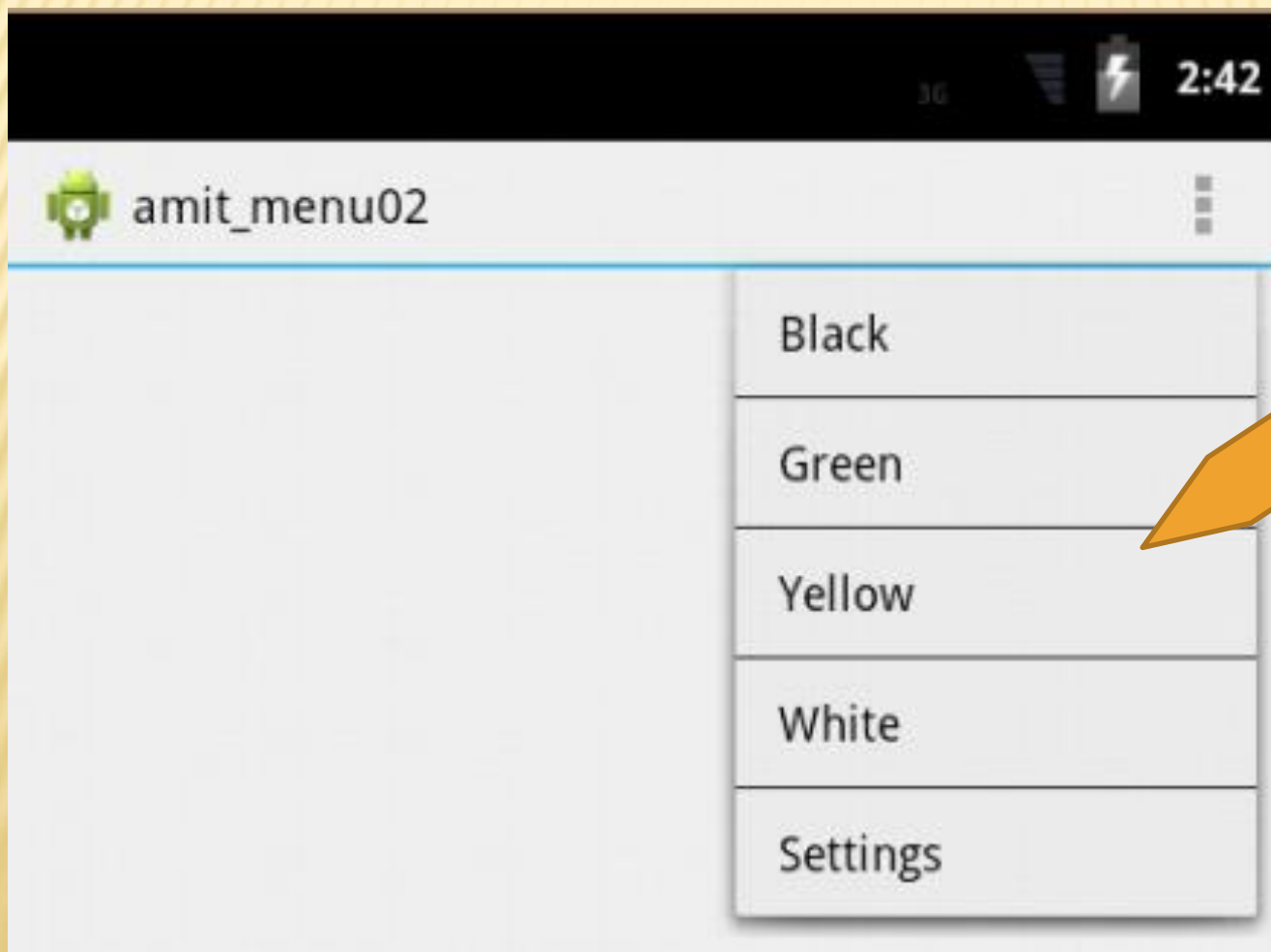


להוסיף שורות לתפריט די פשוט...

```
public boolean onCreateOptionsMenu(Menu menu) {  
  
    // Inflate the menu; this adds items to the adapter  
    getMenuInflater().inflate(R.menu.main, menu);  
    menu.add("Black");  
    menu.add("Green");  
    menu.add("Yellow");  
    menu.add("White");  
  
    return true;  
}
```

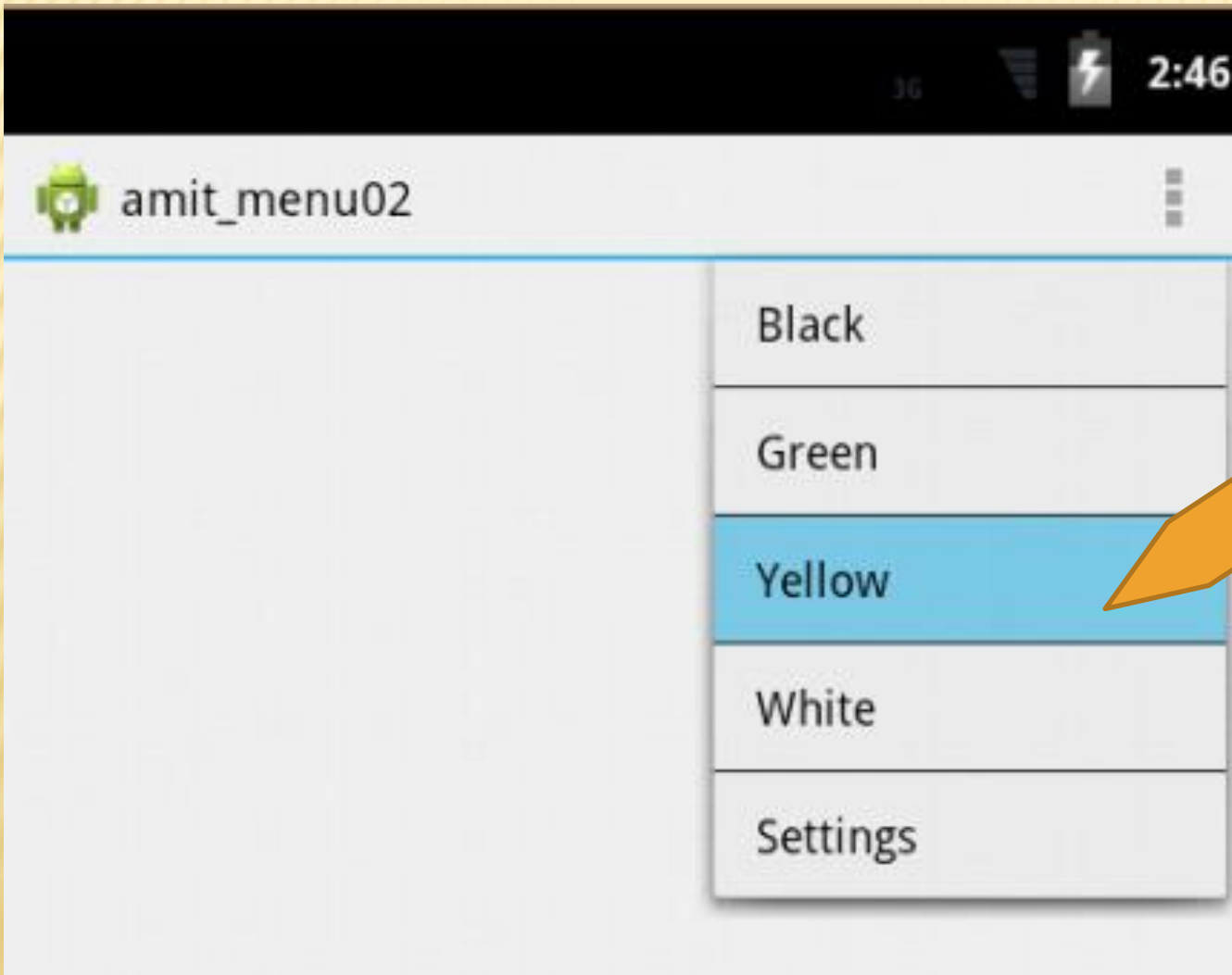


להוסיף שורות לתפריט די פשוט...





אבל בחירת שורה בתפריט לא מבצעת שום דבר...





להפעלת סעיפי התפריט משתמשים בפעולה נוספת



```
public boolean onOptionsItemSelected(MenuItem item) {  
    // Handle action bar item clicks here. The action bar will  
    // automatically handle clicks on the Home/Up button, so long  
    // as you specify a parent activity in AndroidManifest.xml.  
    int id = item.getItemId();  
    if (id == R.id.action_settings) {  
        return true;  
    }  
    return super.onOptionsItemSelected(item);  
}
```




בפעולת זאת רושמים פקודות לתגובה על בחירת כל סעיף

```
public boolean onOptionsItemSelected(MenuItem item) {  
    String st=item.getTitle().toString();  
    if(st.equals("Black"))  
        RL.setBackgroundColor(Color.BLACK);  
    if(st.equals("Green"))  
        RL.setBackgroundColor(Color.GREEN);  
    if(st.equals("Yellow"))  
        RL.setBackgroundColor(Color.YELLOW);  
    if(st.equals("White"))  
        RL.setBackgroundColor(Color.WHITE);  
    return true;  
}
```



פעולה מקבלת כפרמטר
משתנה מסוג "שורת התפריט"

```
public boolean onOptionsItemSelected(MenuItem item) {  
    String st=item.getTitle().toString();  
    if(st.equals("Black"))  
        RL.setBackgroundColor(Color.BLACK);  
    if(st.equals("Green"))  
        RL.setBackgroundColor(Color.GREEN);  
    if(st.equals("Yellow"))  
        RL.setBackgroundColor(Color.YELLOW);  
    if(st.equals("White"))  
        RL.setBackgroundColor(Color.WHITE);  
    return true;  
}
```



על המשתנה אשפר להפעיל
פעולות שונות על מנת לקבל
מאפיינים (תכונות) שלו

```
public boolean onOptionsItemSelected(MenuItem item) {  
    String st=item.getTitle().toString();  
}
```



בעזרת התכונה ניתן להפעיל
תגובות שונות לבחירת שורות
שונות בתפריט

```
public boolean onOptionsItemSelected(MenuItem item) {  
    String st=item.getTitle().toString();  
    if(st.equals("Black"))  
        RL.setBackgroundColor(Color.BLACK);  
    if(st.equals("Green"))  
        RL.setBackgroundColor(Color.GREEN);  
    if(st.equals("Yellow"))  
        RL.setBackgroundColor(Color.YELLOW);  
    if(st.equals("White"))  
        RL.setBackgroundColor(Color.WHITE);  
    return true;  
}
```

פעולה צריכה להחזיר true



קיימת גם דרך נוספת ליצירת תפריט מסוג Options menu



שלבי עבודה

1. ליצור משתנים וערכים ב-`strings.xml`
2. ליצור "סעיפים" ב"קובץ לתפריט" `main.xml`
אשר נמצא בתיקיה `res/menu`
3. לקבוע פעולות בקשר לסעיפים



לצור משתנים וערכים ב-strings.xml

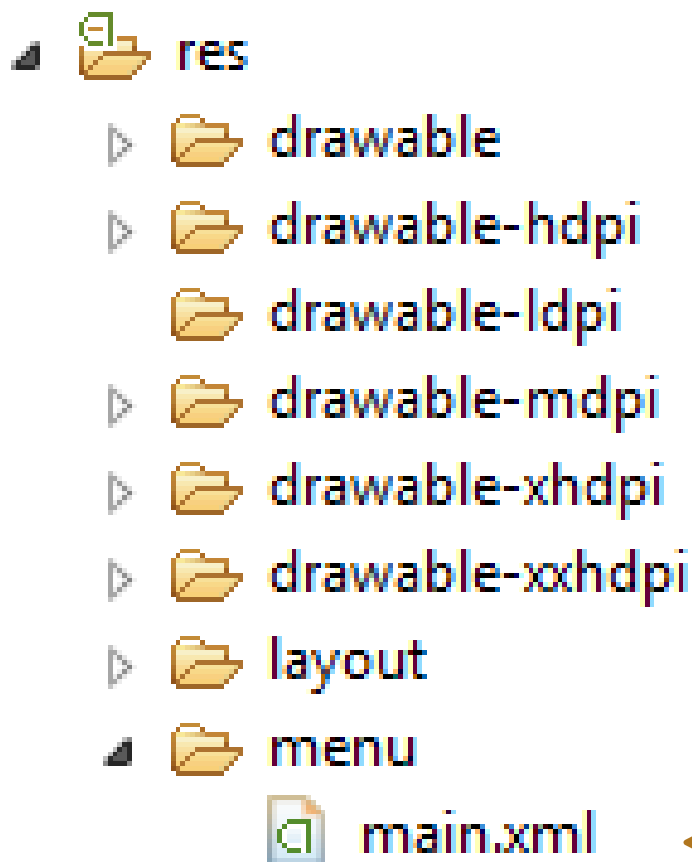
```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">amit_menu01</string>
    <string name="hello_world">Hello world!</string>
    <string name="action_settings">Settings</string>
    <string name="sum">Summa</string>
    <string name="mult">Multiplication</string>
    <string name="diffr">Different</string>
</resources>
```




לצור "סעיפים" ב"קובץ לתפריט"

main.xml אשר נמצא בתיקיה menu





לצור "סעיפים" ב"קובץ לתפריט"

main.xml אשר נמצא בתיקיה menu

```
<item
    android:id="@+id/action_difr"
    android:orderInCategory="100"
    android:title="@string/difr"
    app:showAsAction="never"/>
```

```
<item
    android:id="@+id/action_sum"
    android:orderInCategory="100"
    android:title="@string/sum"
    app:showAsAction="never"/>
```

```
<item
    android:id="@+id/action_mult"
    android:orderInCategory="100"
    android:title="@string/mult"
    app:showAsAction="never"/>
```



זהו המאפיין הכי חשוב="מזהה"

```
<item  
    android:id="@+id/action_difr"  
    android:orderInCategory="100"  
    android:title="@string/difr"  
    app:showAsAction="never"/>
```



```
<item
```

```
    android:id="@+id/action_difr"  
    android:orderInCategory="100"  
    android:title="@string/difr"  
    app:showAsAction="never"/>
```

קביעת מזהה (id)

שימוש בפרמטר

```
public boolean onOptionsItemSelected(MenuItem item) {  
  
    int id = item.getItemId();  
  
    if (id == R.id.action_difr) {  
        tv2.setText(" - ");  
        tv4.setText(""+(a-b));  
        return true;  
    }  
}
```

שימוש במזהה (id)



אפשר ללמוד פרטים נוספים באתר מפתחים

`android:showAsAction`

Keyword. When and how this item should appear as an action item in the Action Bar. A menu item can appear as an action item only when the activity includes an `ActionBar` (introduced in API Level 11). Valid values:

Value	Description
<code>ifRoom</code>	Only place this item in the Action Bar if there is room for it.
<code>withText</code>	Also include the title text (defined by <code>android:title</code>) with the action item. You can include this value along with one of the others as a flag set, by separating them with a pipe .
<code>never</code>	Never place this item in the Action Bar.
<code>always</code>	Always place this item in the Action Bar. Avoid using this unless it's critical that the item always appear in the action bar. Setting multiple items to always appear as action items can result in them overlapping with other UI in the action bar.
<code>collapseActionView</code>	The action view associated with this action item (as declared by <code>android:actionLayout</code> or <code>android:actionViewClass</code>) is collapsible. Introduced in API Level 14.

Site: <http://developer.android.com/guide/topics/resources/menu-resource.html>