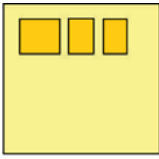
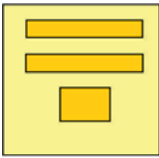
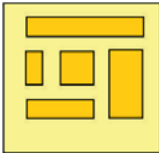


Il layout degli oggetti

Con il termine **layout** si intende la modalità con cui le componenti grafiche sono posizionate all'interno di una finestra. La descrizione del layout delle applicazioni Android è contenuta nei file XML della cartella *res/layout*.

I tre layout maggiormente utilizzati sono:

	LinearLayout <i>orizzontale</i>	Posiziona le componenti al suo interno su un'unica linea. Le componenti figlie vengono aggiunte in ordine da sinistra a destra.
	LinearLayout <i>verticale</i>	Posiziona le componenti al suo interno su un'unica colonna. Le componenti figlie vengono aggiunte in ordine dall'alto in basso.
	RelativeLayout	Permette di specificare la posizione delle componenti in modo relativo alle altre componenti presenti. Le componenti possono essere posizionate a sinistra/destra, o in alto/basso, di un'altra componente, oppure possono essere allineate al contenitore.

Il *LinearLayout* orizzontale è descritto con il seguente tag XML:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="horizontal" >

</LinearLayout>
```

Il *LinearLayout* verticale utilizza lo stesso tag ma con un valore diverso per l'attributo **orientation**.

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

</LinearLayout>
```

Le componenti grafiche vengono aggiunte all'interno del tag *LinearLayout* e si dispongono automaticamente in base all'orientamento impostato.

ESEMPIO

Creare un layout per chiedere l'apertura o il salvataggio di un file.

Il layout è composto da una etichetta, su cui viene visualizzato il nome del file, e due pulsanti per eseguire le operazioni di apertura e salvataggio. Le componenti grafiche vengono posizionate in orizzontale, una di fianco all'altra.

L'etichetta è stata creata con una dimensione grande (*Large*) e prevede uno spazio bianco alla destra del nome del file, inserito tramite l'attributo **paddingRight**.

```
android:paddingRight="50dp"  
android:textAppearance="?android:attr/textAppearanceLarge"
```

Il codice completo del file di layout è riportato di seguito.

Layout (*res/layout/apertura_file.xml*)

```
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/  
android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="horizontal" >  
  
    <TextView  
        android:id="@+id/txtFile"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:paddingRight="50dp"  
        android:text="File.txt"  
        android:textAppearance="?android:attr/textAppearanceLarge" />  
  
    <Button  
        android:id="@+id/btnApri"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Apri" />  
  
    <Button  
        android:id="@+id/btnSalva"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Salva" />  
  
</LinearLayout>
```



ESEMPIO

Creare un layout per gestire la login.

Il layout è composto da due caselle di testo, in cui inserire il nome utente e la password, e un pulsante per verificare i dati inseriti. Le componenti grafiche vengono posizionate in verticale, una sotto l'altra.

Le due caselle di testo occupano tutta la larghezza del layout, come specificato dall'attributo **layout_width**.

```
android:layout_width="match_parent"
```

Il pulsante di login è posizionato nel centro del layout, come specificato dall'attributo **layout_gravity**.

```
android:layout_gravity="center_horizontal"
```

Il testo visualizzato nelle caselle di testo, quando sono vuote, si indica con l'attributo **hint**.

```
android:hint="Utente"
```

Il codice completo del file di layout è riportato di seguito.

Layout (*res/layout/login.xml*)

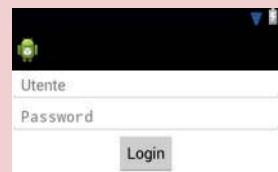
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <EditText
        android:id="@+id/editUtente"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Utente" />

    <EditText
        android:id="@+id/editPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Password"
        android:inputType="textPassword" />

    <Button
        android:id="@+id/btnLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:text="Login" />

</LinearLayout>
```



Il *RelativeLayout* è descritto con il seguente tag XML:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

</RelativeLayout>
```

Le componenti grafiche vengono aggiunte all'interno del tag *RelativeLayout* e si dispongono automaticamente in base al valore degli attributi di posizionamento.

La seguente tabella mostra gli attributi per posizionare le componenti in modo relativo rispetto al contenitore.

Attributo	Descrizione
<code>layout_alignParentTop</code>	Se ha il valore <i>true</i> , la componente è posizionata in alto nel contenitore.
<code>layout_alignParentBottom</code>	Se ha il valore <i>true</i> , la componente è posizionata in basso nel contenitore.
<code>layout_alignParentLeft</code>	Se ha il valore <i>true</i> , la componente è posizionata a sinistra nel contenitore.
<code>layout_alignParentRight</code>	Se ha il valore <i>true</i> , la componente è posizionata a destra nel contenitore.
<code>layout_centerVertical</code>	Se ha il valore <i>true</i> , la componente è centrata verticalmente rispetto al contenitore.
<code>android:layout_centerHorizontal</code>	Se ha il valore <i>true</i> , la componente è centrata orizzontalmente rispetto al contenitore.
<code>android:layout_centerInParent</code>	Se ha il valore <i>true</i> , la componente è centrata orizzontalmente e verticalmente rispetto al contenitore.

Per esempio, per allineare un pulsante alla destra del suo contenitore si inserisce l'attributo *layout_alignParentRight* con il valore *true* nel tag della componente.

```
<Button
    android:id="@+id/btnConferma"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:text="Conferma" />
```

La seguente tabella mostra gli attributi per posizionare le componenti in modo relativo rispetto alle altre componenti.

Attributo	Descrizione
layout_toLeftOf	Posiziona lo spigolo destro della componente alla sinistra della componente indicata come parametro.
layout_toRightOf	Posiziona lo spigolo sinistro della componente alla destra della componente indicata come parametro.
layout_above	Posiziona lo spigolo inferiore della componente sopra la componente indicata come parametro.
layout_below	Posiziona lo spigolo superiore della componente sotto la componente indicata come parametro.
layout_alignTop	Allinea lo spigolo superiore della componente con lo spigolo superiore della componente indicata come parametro.
layout_alignBottom	Allinea lo spigolo inferiore della componente con lo spigolo inferiore della componente indicata come parametro.
layout_alignLeft	Allinea lo spigolo sinistro componente con lo spigolo sinistro della componente indicata come parametro.
layout_alignRight	Allinea lo spigolo destro della componente con lo spigolo destro della componente indicata come parametro.

Per esempio, per allineare la casella di testo *editAnno* alla destra della casella di testo *editMese*, in modo che siano allineati anche gli spigoli inferiori, si utilizzano gli attributi *layout_toRightOf* e *layout_alignBottom*. Si noti che il nome della componente è fatto precedere dalla sequenza di caratteri **@+id/**.

```
<EditText
    android:id="@+id/editAnno"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/editMese"
    android:layout_toRightOf="@+id/editMese"
    android:hint="aaaa"
    android:inputType="number" />
```

ESEMPIO

Creare un layout per l'inserimento dei dati di pagamento con carta di credito.

Il layout è composto da diverse caselle di testo per la raccolta dei dati, un'etichetta e un pulsante di conferma.

Il codice completo del file di layout è riportato di seguito.

Layout (*res/layout/pagamento.xml*)

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```

<EditText
    android:id="@+id/editNominativo"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:hint="Nominativo" />

<EditText
    android:id="@+id/editNumero"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/editNominativo"
    android:layout_toLeftOf="@+id/editCVV"
    android:inputType="number" />

<Button
    android:id="@+id/btnConferma"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/editAnno"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/editNominativo"
    android:text="Conferma" />

<EditText
    android:id="@+id/editCVV"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/editNumero"
    android:layout_toLeftOf="@+id/btnConferma"
    android:inputType="number"
    android:width="60dp" />

<TextView
    android:id="@+id/txt"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/editMese"
    android:layout_alignParentLeft="true"
    android:text="Scadenza"
    android:textAppearance="?android:attr/textAppearanceSmall" />

<EditText
    android:id="@+id/editAnno"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/editMese"
    android:layout_alignRight="@+id/editCVV"
    android:layout_toRightOf="@+id/editMese"
    android:hint="aaaa"
    android:inputType="number" />

```

```
<EditText
    android:id="@+id/editMese"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/editNumero"
    android:layout_toRightOf="@+id/txt"
    android:hint="mm"
    android:inputType="number"
    android:width="75dp" />
```

```
</RelativeLayout>
```

