



Tutorial App Inventor #2 - messaggio automatico

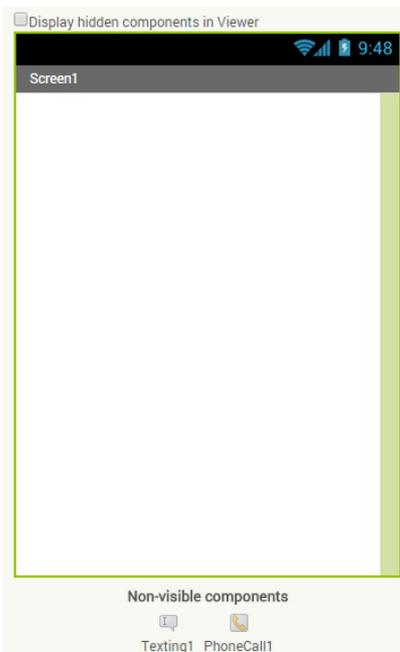
Leggi tutti i tutorial di App Inventor - [clicca qui](#)

In questo tutorial utilizzeremo due componenti che sono gli elementi base di uno smartphone, la gestione delle chiamate e dei messaggi. Realizziamo un'App in grado di inviare in automatico un sms se riceviamo una chiamata mentre siamo alla guida della nostra auto o in palestra.

I due componenti utilizzati sono **PhoneCall** e **Texting** (presenti nel pannello di controllo di App Inventor sotto la voce SOCIAL).

Collegatevi ad App Inventor - Start New Project - inserire il nome "lez_2_smsautomatico".

Trascinate i due componenti PhoneCall e Texting all'interno del display.



I due elementi appaiono in fondo perchè sono due componenti invisibili. Per il momento la parte grafica termina qui. Passiamo alla programmazione cliccando sul bottone **Blocks** (in alto a destra).

La logica di programmazione di questo esercizio è questa qui:

Quando ricevo una telefonata (e la telefonata termina perchè non posso rispondere) invio un sms al numero di telefono che mi ha chiamato con un testo impostato da noi (ad esempio: Scusa non posso risponderti ti chiamo appena possibile...).

Cerchiamo di ricostruire queste azioni con le proprietà dei due blocchi. Partiamo da: "**Quando ricevo una telefonata e questa termina...**" questo genere di istruzione è presente nel blocco di gestione delle telefonate (PhoneCall).



Click su **PhoneCall** per visualizzare le sue proprietà di programmazione.

Il componente che dobbiamo selezionare è **When PhoneCallEnded**.

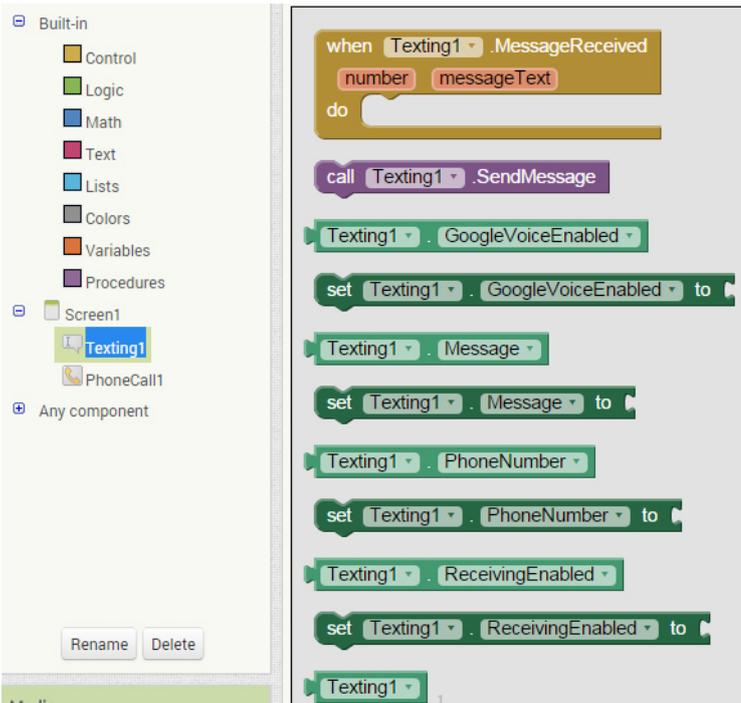


```
when PhoneCall1 .PhoneCallEnded
  status  phoneNumber
do
```

Questo blocco ha al suo interno una variabile che si chiama **phoneNumber** dove è presente il numero di telefono della persona che ci ha chiamato.

All'interno del blocco marrone dobbiamo incastrare i componenti che ricostruiscono l'azione: **"invio un sms al numero di telefono che mi ha chiamato con un testo impostato da noi..."**

Click sul componente **Texting** per scoprire le sue proprietà.



Tra le numerose opzioni scartiamo quelle che non entrano all'interno del componente PhoneCallEnded. Selezioniamo il pezzo **call Texting1.SendMessage** (di colore viola). Questo blocco (quando è chiamato) invia un sms.

Tra i blocchi verdi (le proprietà di un comando) dobbiamo selezionare **set Texting Message to** e **set Texting PhoneNumber to**.

Con il primo blocco inseriamo un testo nel messaggio da inviare, con il secondo invece impostiamo il numero di telefono a cui dobbiamo inviarlo.

```
when PhoneCall1 .PhoneCallEnded
  status  phoneNumber
do
  set Texting1 . PhoneNumber to
  set Texting1 . Message to
  call Texting1 .SendMessage
```

I due blocchi verdi servono a settare il blocco viola che poi compierà l'azione (invio dell'sms).

I due blocchi verdi però devono essere completati con delle informazioni.

Nel primo dobbiamo aggiungere il numero di telefono a cui inviare l'sms.

Passate con il mouse sul blocco marrone sopra la scritta **phoneNumber**. Appare una finestra con due componenti.

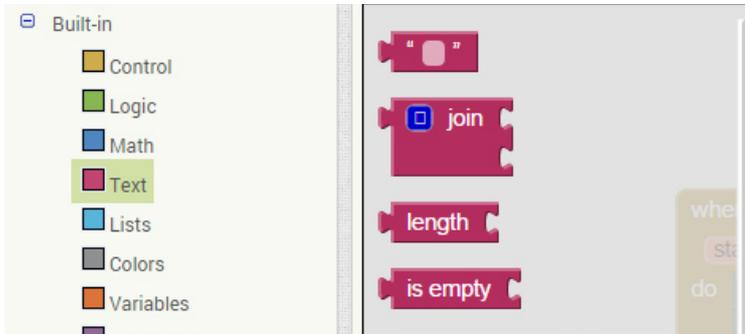
```
when PhoneCall1 .PhoneCallEnded
  status  phoneNumber
do
  set Texting1 . PhoneNumber to get phoneNumber
  set Texting1 . Message to
  call Texting1 .SendMessage
```

Selezioniamo **getPhoneNumber** e incastriamolo con il blocco verde **set Texting PhoneNumber to**.

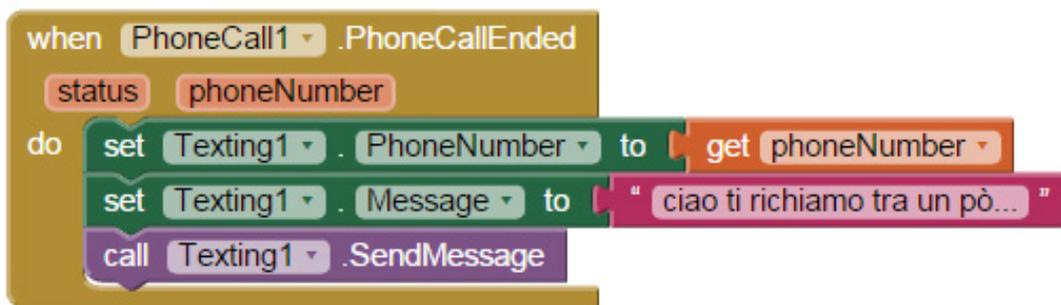
In questo modo abbiamo detto alla nostra App setta il numero di telefono con il numero di telefono della persona che ci ha chiamato.



Per completare l'App dobbiamo impostare il testo del messaggio da inviare. Click su **Text** (a sinistra all'interno dei componenti Built-in). Selezioniamo il primo blocco rosso (nella figura qui sotto) e incastriamolo con il blocco verde **set TextingMessage to**.



Click sul blocco di testo e scriviamo il testo del messaggio, ad esempio "ciao ti richiamo tra un pò sto guidando..."



La nostra App è terminata, esportate il file .apk ([clicca qui](#) per vedere il tutorial precedente) e installate l'App.

Fate una prova con un vostro amico. Attivate l'App e fatevi chiamare senza rispondere. Una volta terminata la telefonata il vostro smartphone invierà in maniera automatica il messaggio predefinito.

Nella prossima pagina vediamo un pò di grafica e l'elemento **checkBox** per migliorare la nostra App.

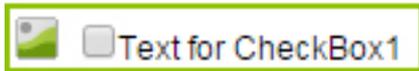


Click su **Designer** per tornare alla modalità grafica.

CLICCA QUI per scaricare le immagini da utilizzare per completare l'App.

Oltre alla grafica modifichiamo leggermente l'App inserendo tre checkBox per selezionare tre diversi messaggi a seconda se stiamo guidando, siamo in piscina o in palestra.

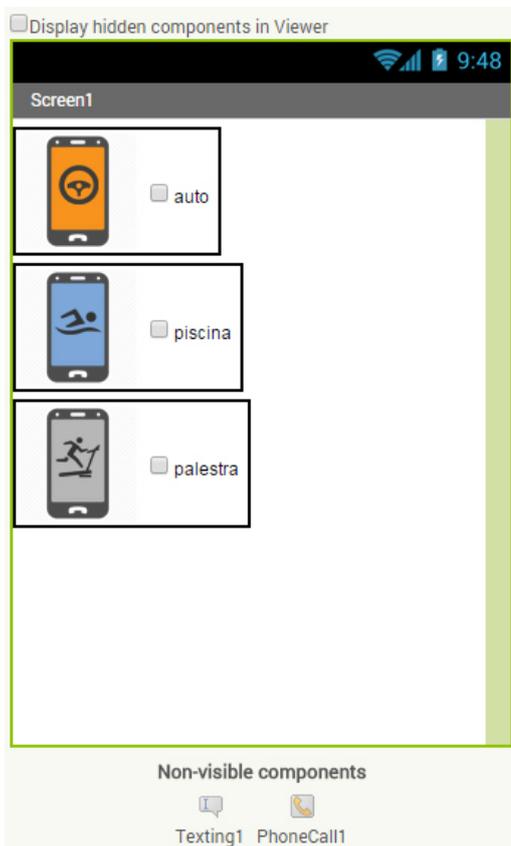
Trasciniamo nel Viewer l'elemento HorizontalArrangement presente all'interno della sezione Layout. Questo componente permette di inserire nel display diversi elementi in orizzontale.



Al suo interno inseriamo una **image** e un **checkBox**.

Click sull'icona dell'immagine e nelle sue proprietà click su **Picture**. Qui possiamo caricare l'immagine precedentemente scaricata **bt_auto.png**.

Click sul checkBox e nelle sue proprietà, in Text, scriviamo "auto".



Aggiungiamo altri due HorizontalArrangement e inseriamo altre due immagini e due checkbox.

Nelle immagini caricate i file **bt_piscina.png** e **bt_palestra.png**.

Nei checkbox inserite le scritte "piscina" e "palestra".

Completiamo questa parte di layout cliccando sul primo checkBox. Nella finestra **Components** click su **Rename** e scriviamo "ck_auto". stessa cosa per gli altri due checkBox, "ck_piscina" e "ck_palestra".

Abbiamo cambiato nome ai componenti per individuarli più facilmente durante la programmazione.

Click su **Blocks** per passare alla programmazione.

La logica dell'App è identica a quella precedente. In questo caso quello che cambia è vedere quale dei tre checkBox è stato attivato per inviare un messaggio specifico.



```
when PhoneCall1 .PhoneCallEnded
  status phoneNumber
do
  if
  then
```

Subito dopo il termine della chiamata verifichiamo quale checkBox è stato selezionato con il comando **IF**. Click su **Control**, selezionate il primo blocco e trascinatelo dentro When.PhoneCallEnded.

Il blocco IF funziona in questo modo: il controllo che viene inserito nel suo primo pin è quello di verifica. Se questa condizione è vera allora esegue l'azione che incastriamo sotto.

```
when PhoneCall1 .PhoneCallEnded
  status phoneNumber
do
  if ck_auto . Checked
  then
    set Texting1 . PhoneNumber to get phoneNumber
    set Texting1 . Message to " ciao ti richiamo tra un pò sto guidando... "
    call Texting1 .SendMessage
```

Nell'immagine sopra la struttura del controllo IF completa.

Se (**IF**) il checkBox auto è stato selezionato (**Checked**) allora esegui l'azione che abbiamo visto nella prima versione di questa App.

La proprietà **ck_auto.Checked** da inserire nella IF la trovate tra le azioni del componente ck_auto.

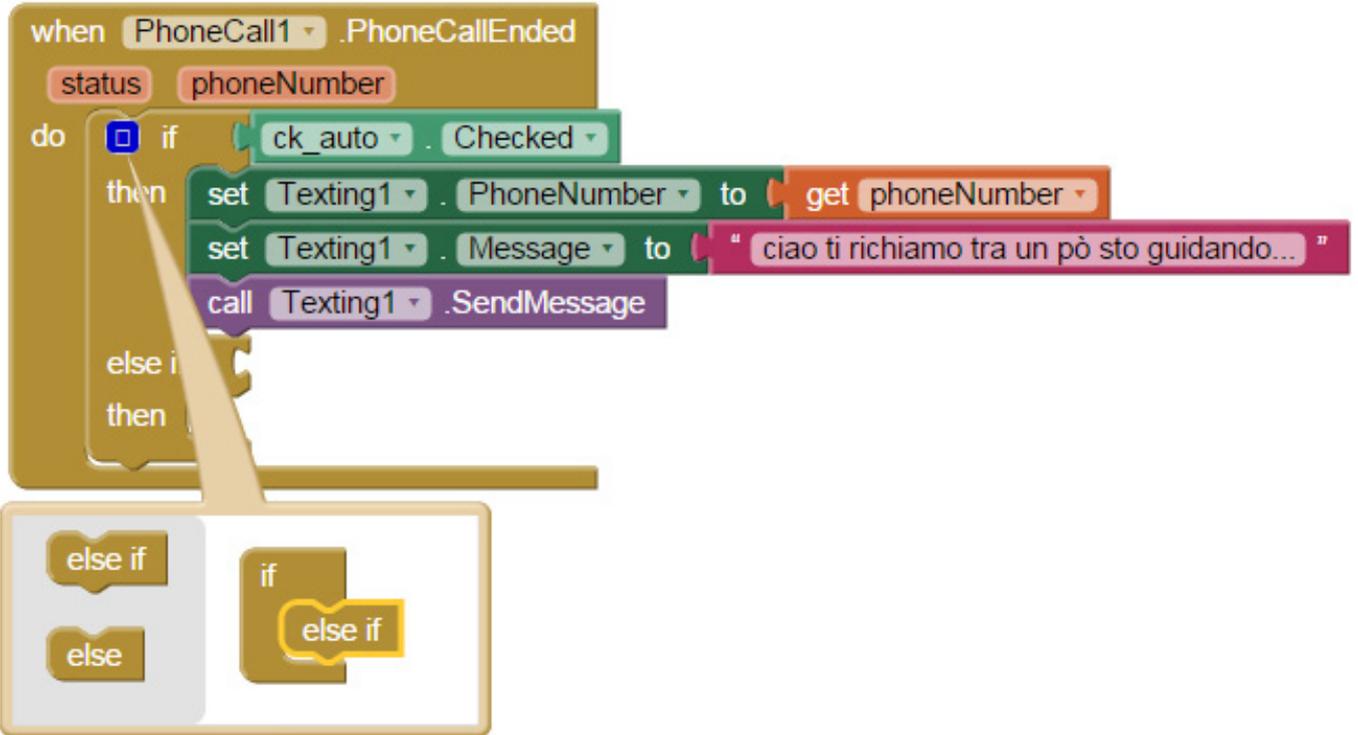
Per eseguire le altre due verifiche (se è stato premuto il checkBox piscina o palestra) dobbiamo aggiungere altri due controlli IF.

Click sull'icona blu vicino alla scritta IF.

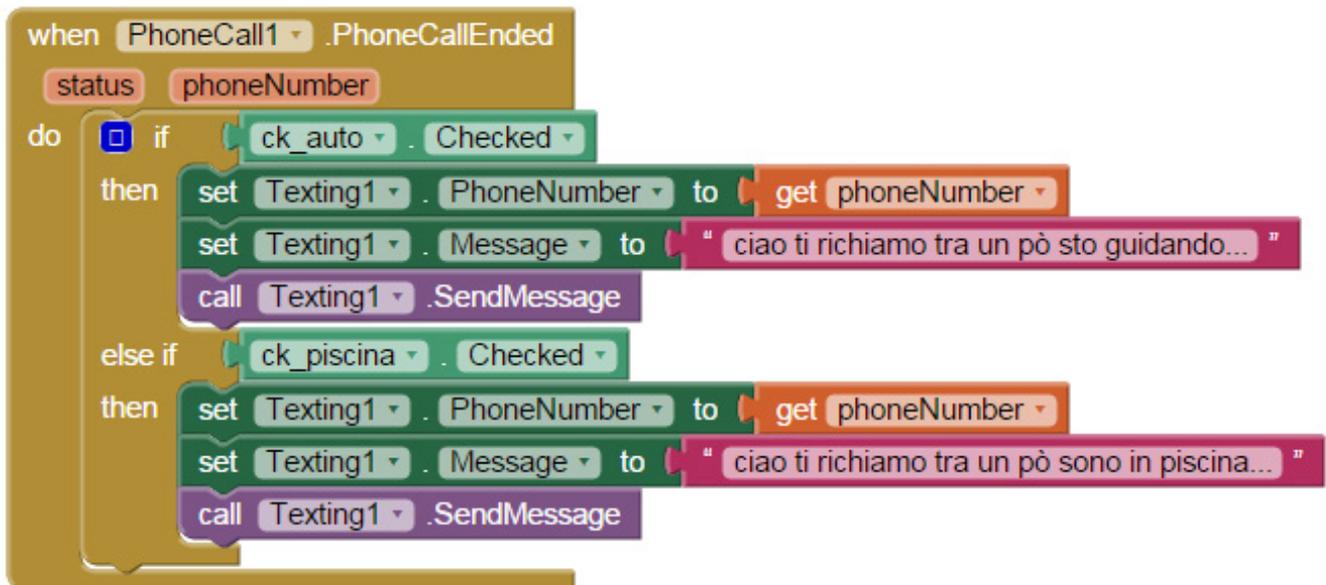
```
when PhoneCall1 .PhoneCallEnded
  status phoneNumber
do
  if ck_auto . Checked
  then
    set Texting1 . PhoneNumber to get phoneNumber
    set Texting1 . Message to " ciao ti richiamo tra un pò sto guidando... "
    call Texting1 .SendMessage
```



Trascinate il blocco **Else If** dentro il blocco IF (come nella figura qui sotto).
In automatico apparirà la seconda IF.



All'interno di questo secondo controllo IF inseriamo la verifica sul checkBox piscina.



Click sull'icona blu per inserire il terzo controllo IF sul checkBox palestra.



Qui sotto tutti i blocchi della nostra App.

```
when PhoneCall1 . PhoneCallEnded
  status phoneNumber
do
  if ck_auto . Checked
  then
    set Texting1 . PhoneNumber to get phoneNumber
    set Texting1 . Message to " ciao ti richiamo tra un pò sto guidando... "
    call Texting1 . SendMessage
  else if ck_piscina . Checked
  then
    set Texting1 . PhoneNumber to get phoneNumber
    set Texting1 . Message to " ciao ti richiamo tra un pò sono in piscina... "
    call Texting1 . SendMessage
  else if ck_palestra . Checked
  then
    set Texting1 . PhoneNumber to get phoneNumber
    set Texting1 . Message to " ciao ti richiamo tra un pò sono in palestra... "
    call Texting1 . SendMessage
```

Completiamo l'App aggiungendo l'icona che verrà visualizzata sul nostro smartphone una volta installata. Click su **Designer** - nella finestra **Components** click su **Screen1** e nelle sue **proprietà** click su **Icon**. Caricate il file **icon.png**. Potete modificare a vostro piacere tutte le immagini che abbiamo caricato. Per l'icona serve un file delle dimensioni di 48x48 pixel.

Click su **Build** e poi su **Save .apk to my computer**.