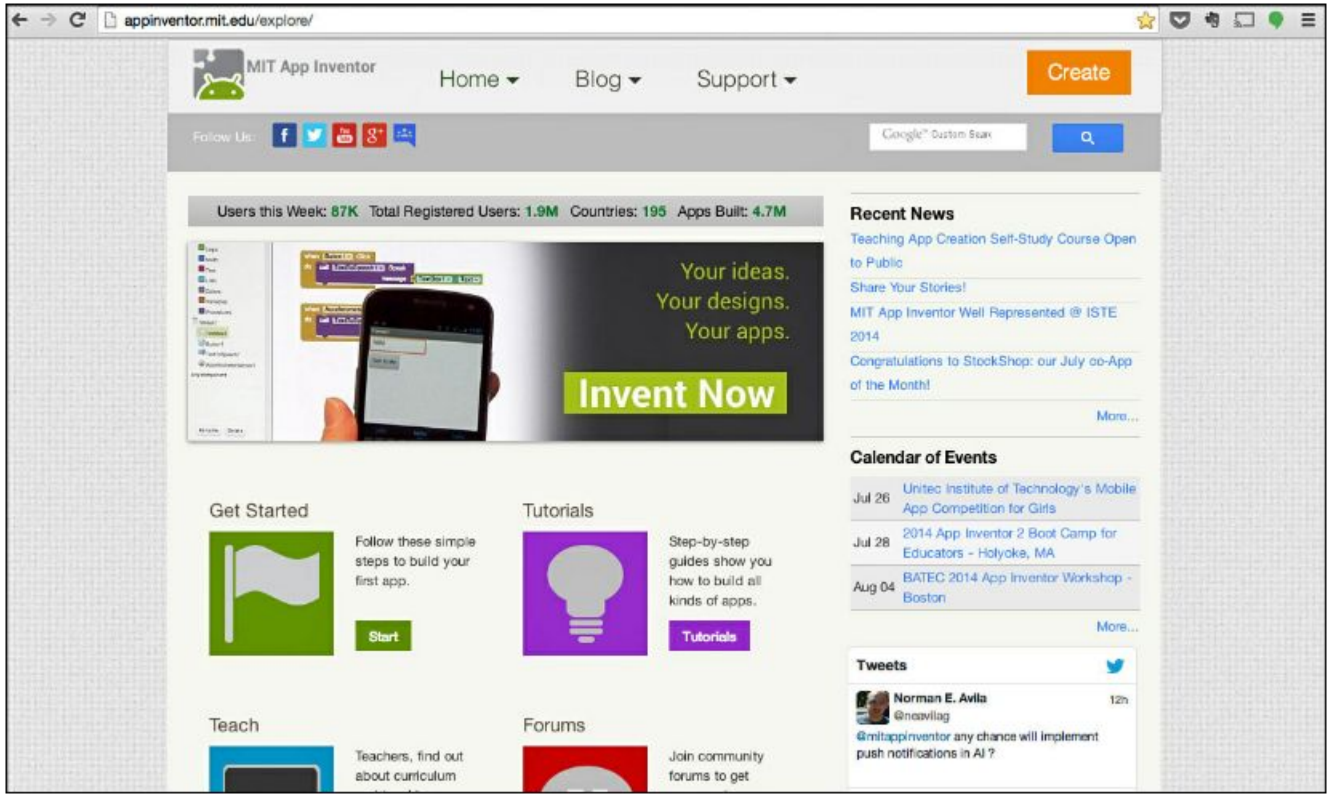
PROJECT-1
TRANSLATOR



Users this Week: 87K Total Registered Users: 1.9M Countries: 195 Apps Built: 4.7M

Recent News

Teaching App Creation Self-Study Course Open to Public

Share Your Stories!

MIT App Inventor Well Represented @ ISTE 2014

Congratulations to StockShop: our July co-App of the Month!

More...

Calendar of Events

Jul 26	Unitec Institute of Technology's Mobile App Competition for Girls
Jul 28	2014 App Inventor 2 Boot Camp for Educators - Holyoke, MA
Aug 04	BATEC 2014 App Inventor Workshop - Boston

More...

Tweets

Norman E. Avila @neavilag 12h

@mitappinventor any chance will implement push notifications in AI ?

Get Started

Follow these simple steps to build your first app.

Tutorials

Step-by-step guides show you how to build all kinds of apps.

Teach

Teachers, find out about curriculum

Forums

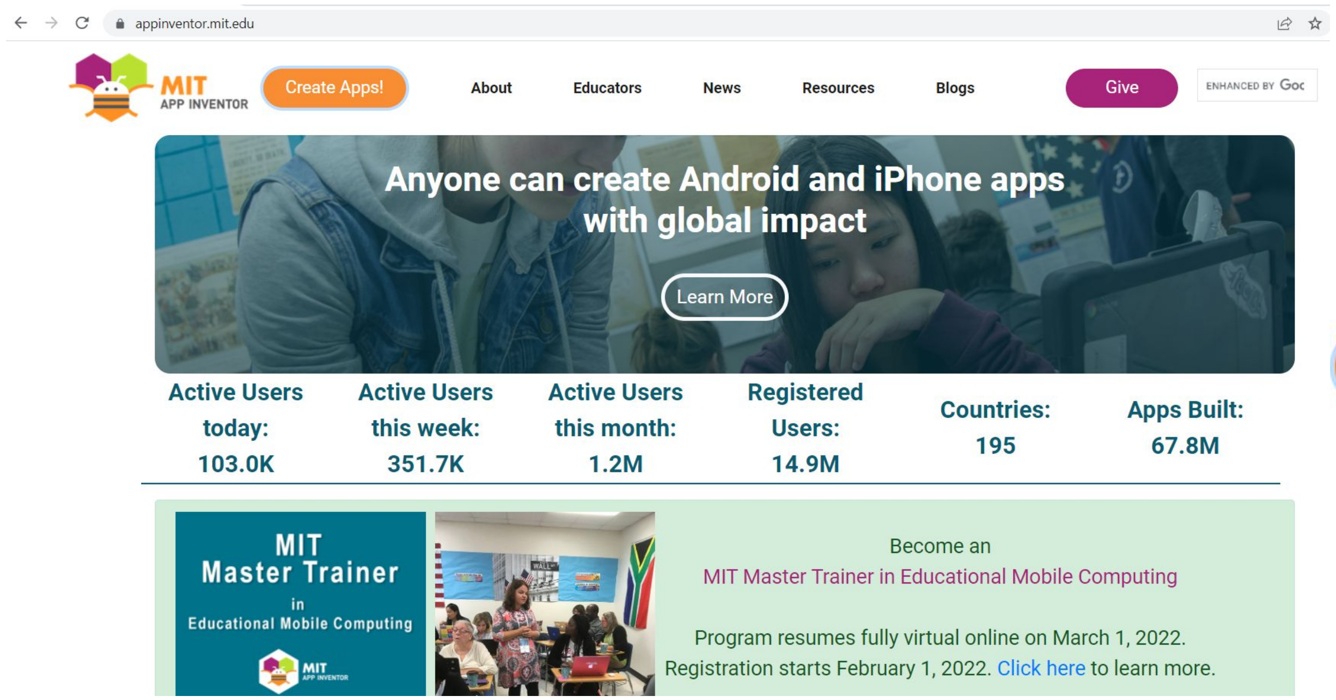
Join community forums to get

APP INVENTOR

BUILDING APPLICATIONS WITH APP INVENTOR

- App Inventor works over the internet.
- You do not need to download any other program.
- The web browser you use is important because it works over the Internet.
- Google Chrome or firefox is the most suitable web browser.
- You can login at appinventor.mit.edu.

BUILDING APPLICATIONS WITH APP INVENTOR


appinventor.mit.edu

MIT APP INVENTOR | Create Apps! | About | Educators | News | Resources | Blogs | Give

Anyone can create Android and iPhone apps with global impact

Learn More

Active Users today:	Active Users this week:	Active Users this month:	Registered Users:	Countries:	Apps Built:
103.0K	351.7K	1.2M	14.9M	195	67.8M

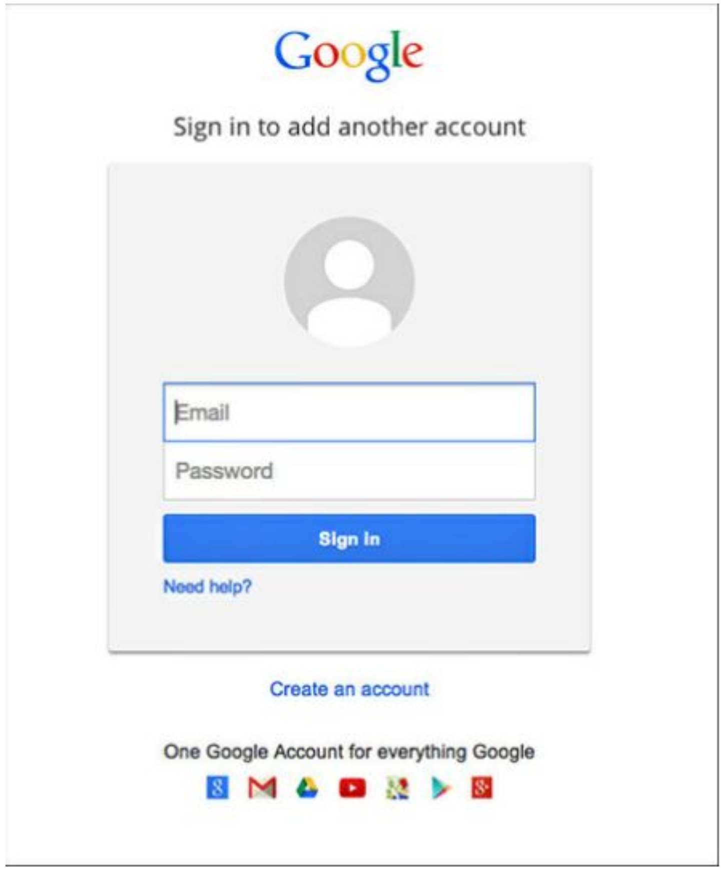
MIT Master Trainer in Educational Mobile Computing

Become an MIT Master Trainer in Educational Mobile Computing

Program resumes fully virtual online on March 1, 2022. Registration starts February 1, 2022. Click here to learn more.

After logging in to AppInventor, we press Create Apps! the menu to program our first application.

BUILDING APPLICATIONS WITH APP INVENTOR


Google

Sign in to add another account

Email

Password

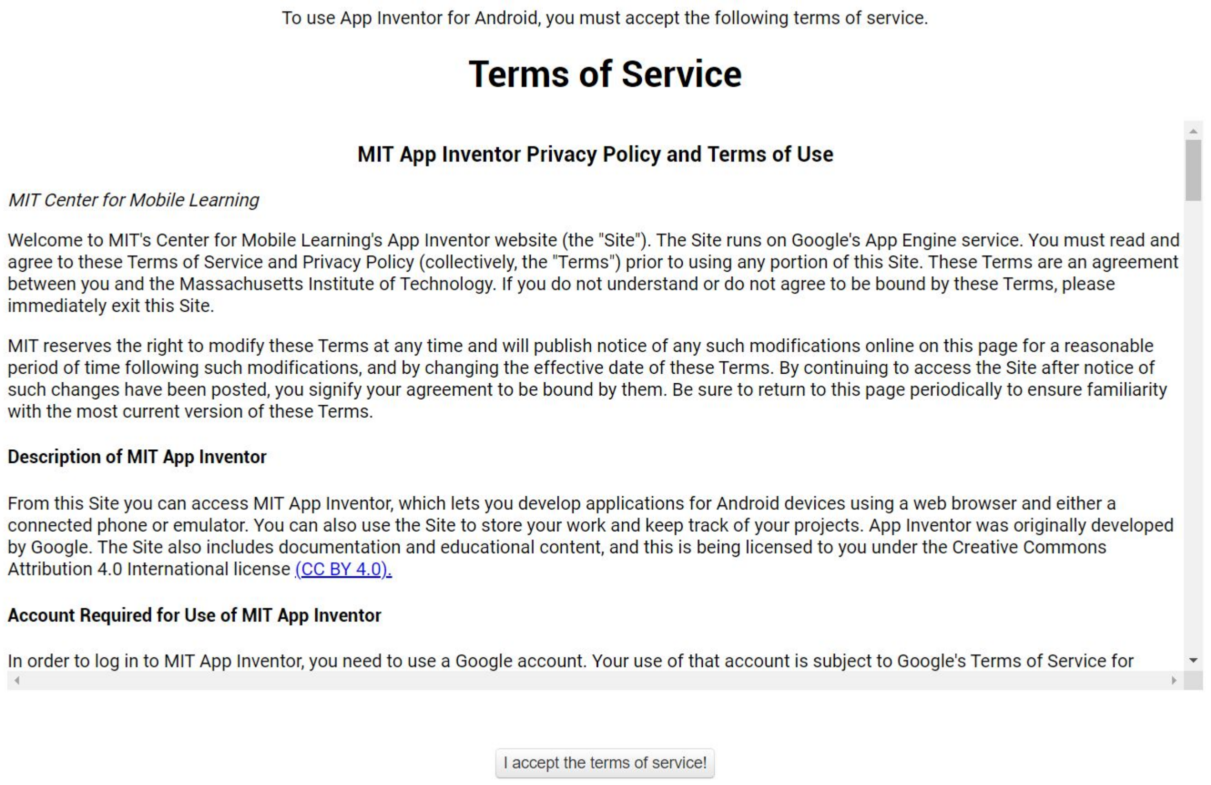
Sign in

Need help?

Create an account

One Google Account for everything Google

- Since Appinventor works on Android, you need to have a Google account.
- If you do not have a Google account, you must create one from the menu. Create an account
- If you have a Google account, you must enter your account and password and log in.


To use App Inventor for Android, you must accept the following terms of service.

Terms of Service

MIT App Inventor Privacy Policy and Terms of Use

MIT Center for Mobile Learning

Welcome to MIT's Center for Mobile Learning's App Inventor website (the "Site"). The Site runs on Google's App Engine service. You must read and agree to these Terms of Service and Privacy Policy (collectively, the "Terms") prior to using any portion of this Site. These Terms are an agreement between you and the Massachusetts Institute of Technology. If you do not understand or do not agree to be bound by these Terms, please immediately exit this Site.

MIT reserves the right to modify these Terms at any time and will publish notice of any such modifications online on this page for a reasonable period of time following such modifications, and by changing the effective date of these Terms. By continuing to access the Site after notice of such changes have been posted, you signify your agreement to be bound by them. Be sure to return to this page periodically to ensure familiarity with the most current version of these Terms.

Description of MIT App Inventor

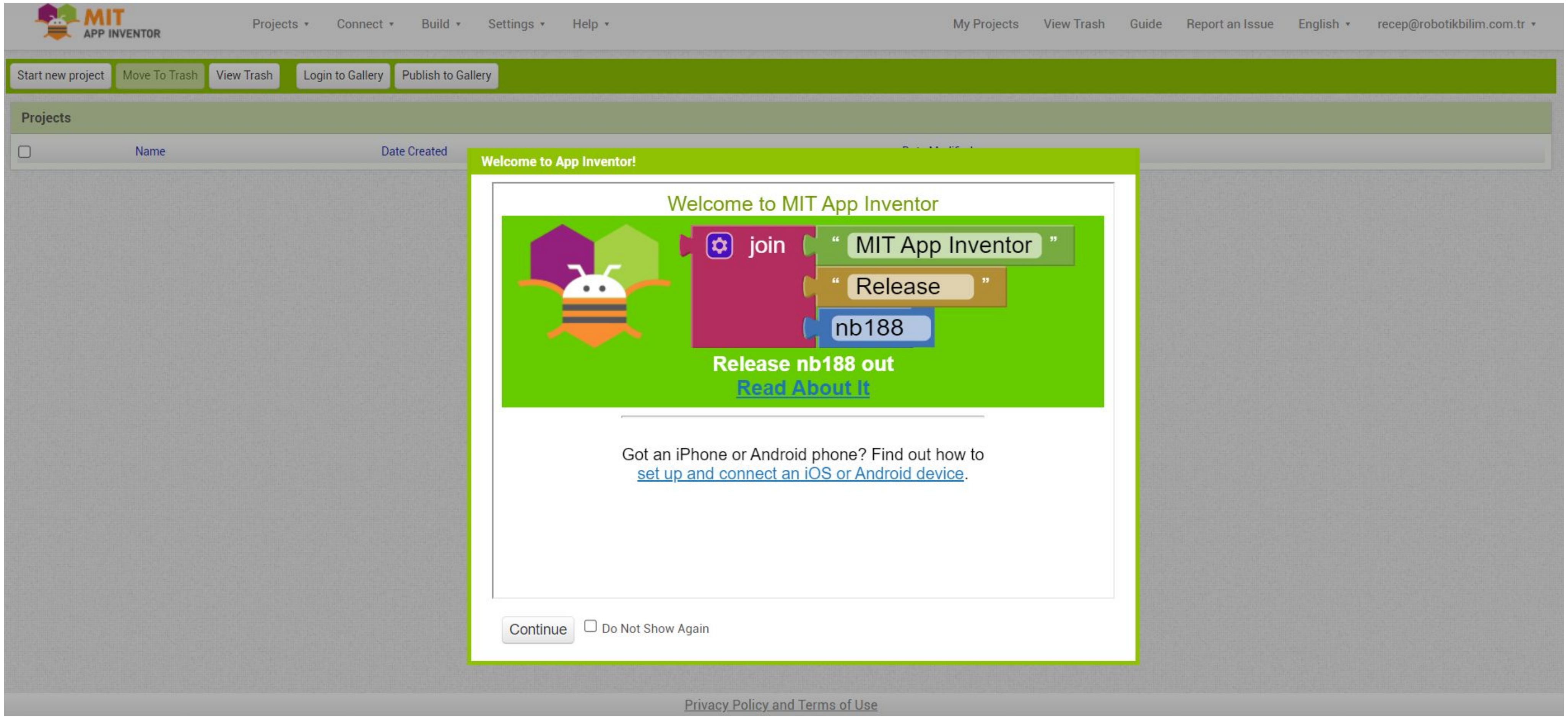
From this Site you can access MIT App Inventor, which lets you develop applications for Android devices using a web browser and either a connected phone or emulator. You can also use the Site to store your work and keep track of your projects. App Inventor was originally developed by Google. The Site also includes documentation and educational content, and this is being licensed to you under the Creative Commons Attribution 4.0 International license (CC BY 4.0).

Account Required for Use of MIT App Inventor

In order to log in to MIT App Inventor, you need to use a Google account. Your use of that account is subject to Google's Terms of Service for

BUILDING APPLICATIONS WITH APP INVENTOR

- After logging in to your Google account, a screen will appear about confirming the terms of use.
- After confirming all the conditions, you can start the first project.



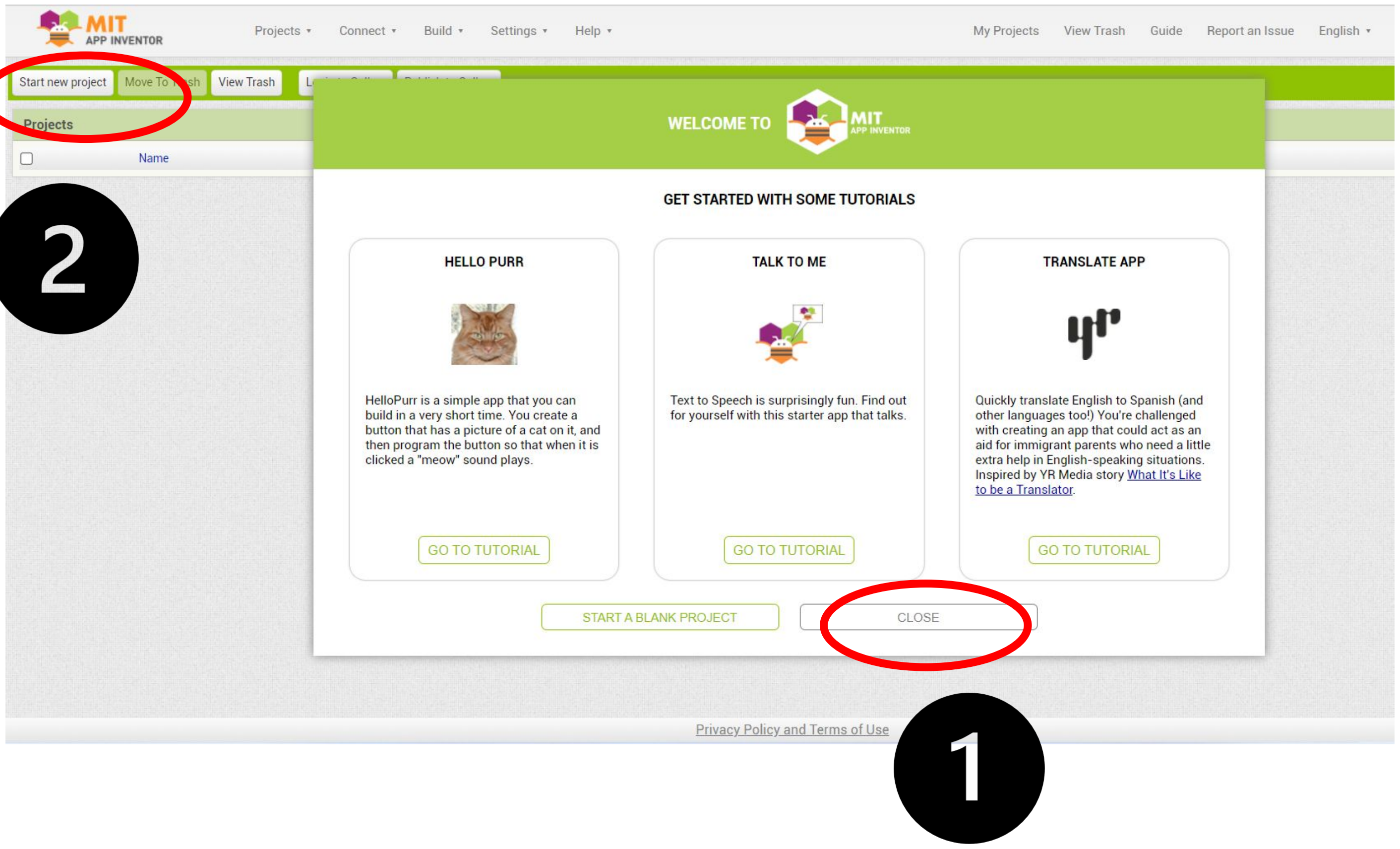
Welcome to MIT App Inventor

Release nb188 out

Read About It

Got an iPhone or Android phone? Find out how to set up and connect an iOS or Android device.

BUILDING APPLICATIONS WITH APP INVENTOR



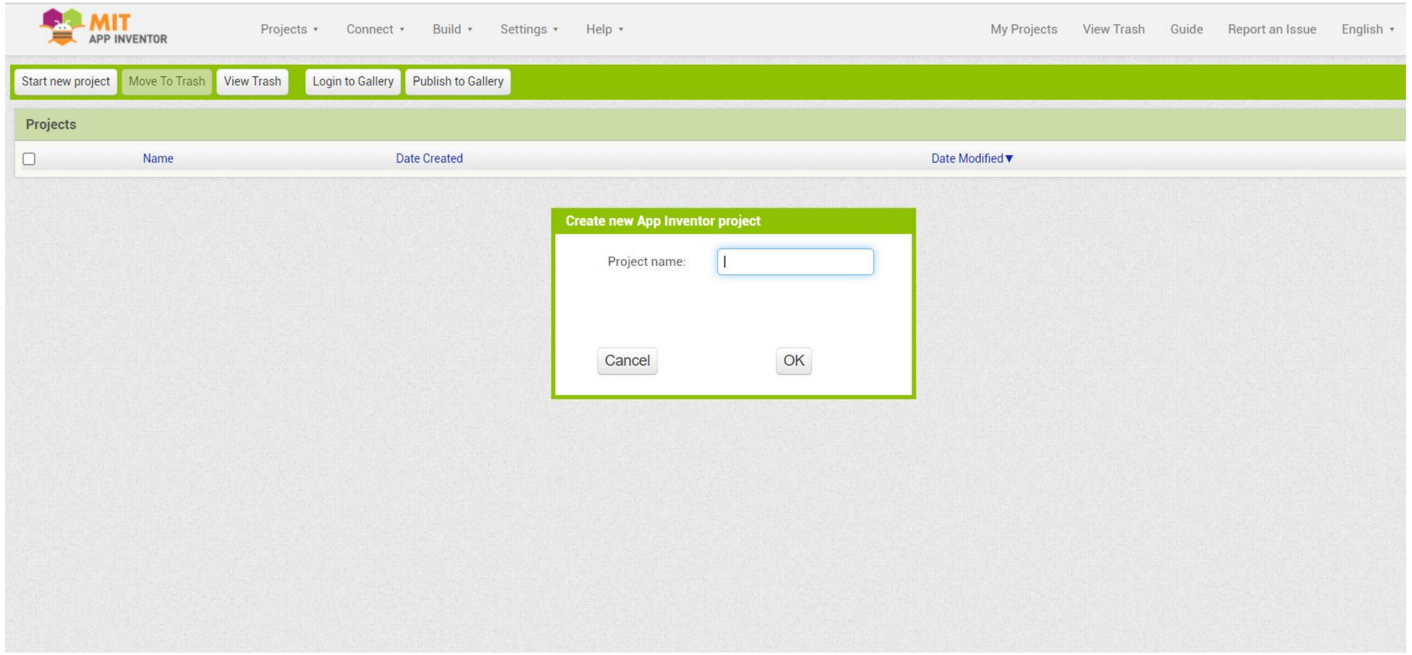
HelloPurr is a simple app that you can build in a very short time. You create a button that has a picture of a cat on it, and then program the button so that when it is clicked a "meow" sound plays.

Text to Speech is surprisingly fun. Find out for yourself with this starter app that talks.

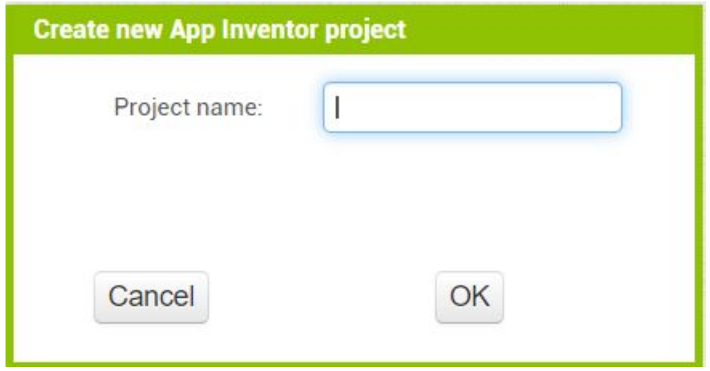
Quickly translate English to Spanish (and other languages too!) You're challenged with creating an app that could act as an aid for immigrant parents who need a little extra help in English-speaking situations. Inspired by YR Media story What It's Like to be a Translator.

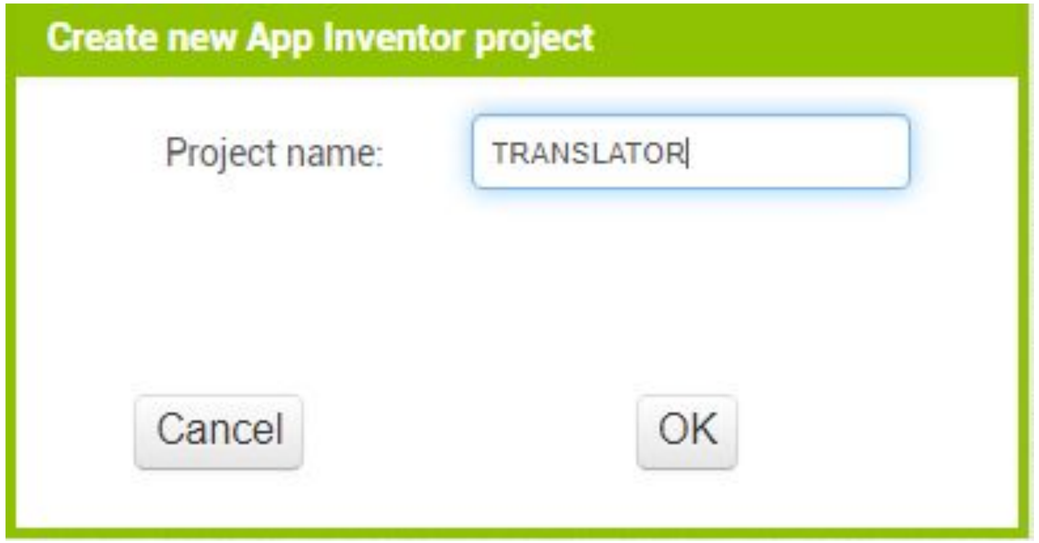
- After logging in to the main screen, 3 tutorials appear. You can find detailed information about appinventor with these tutorials.
- Let's start creating our first project by pressing the Start a Blank Project button.

Application 1: TRANSLATOR



- Before we start programming the Mobile App, we must first give our project a suitable name.
- The name we will give will also be the name that will appear on our phone.
- **App Inventor does not allow spaces and punctuation in the project name.**




Create new App Inventor project

Project name: TRANSLATOR

Application 1: TRANSLATOR

- Let the name of our first project be TRANSLATOR.
- Let's write the name as seen on the screen and click the ok button.

Application 1: TRANSLATOR

Home Page: Designer

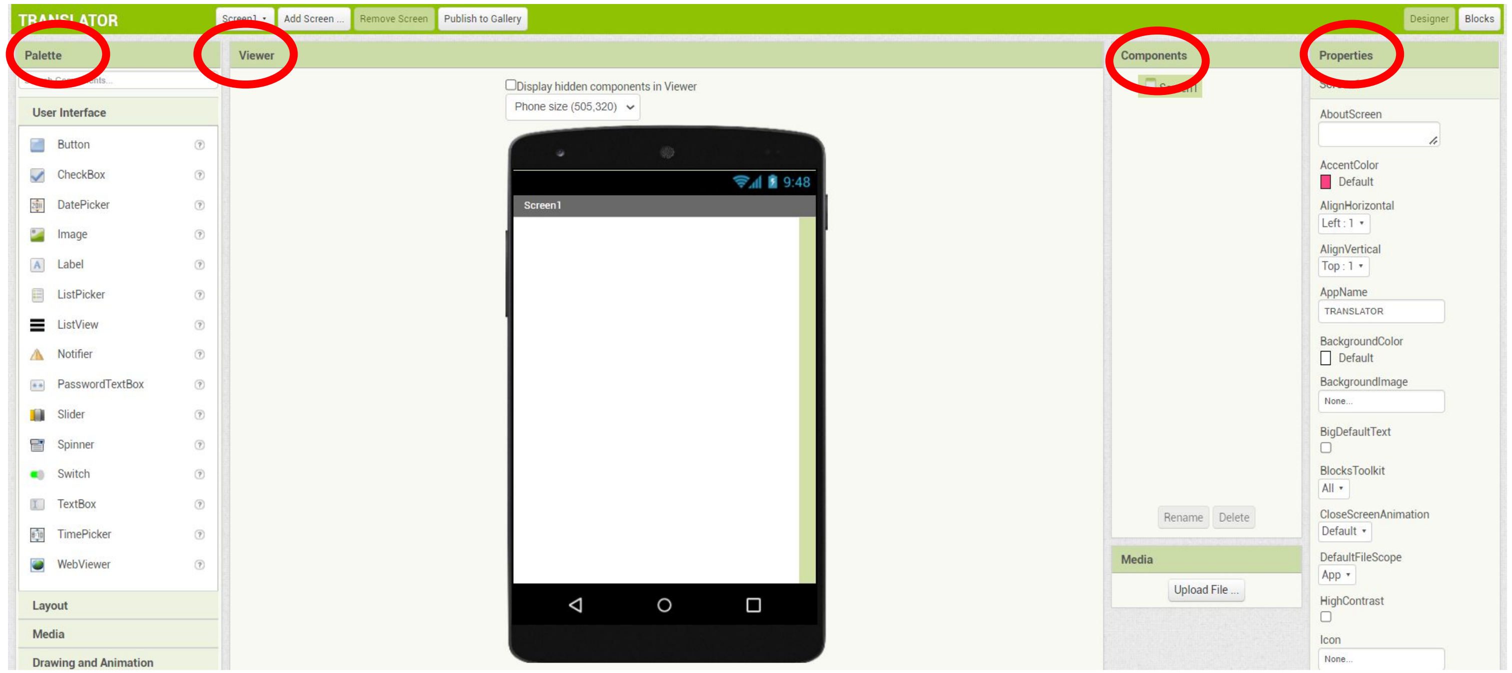
- The main page consists of 4 sections.

Palette

Viewer

Components

Properties



AboutScreen

AccentColor: Default

AlignHorizontal: Left : 1

AlignVertical: Top : 1

AppName: TRANSLATOR

BackgroundColor: Default

BackgroundImage: None...

BigDefaultText

BlocksToolkit: All

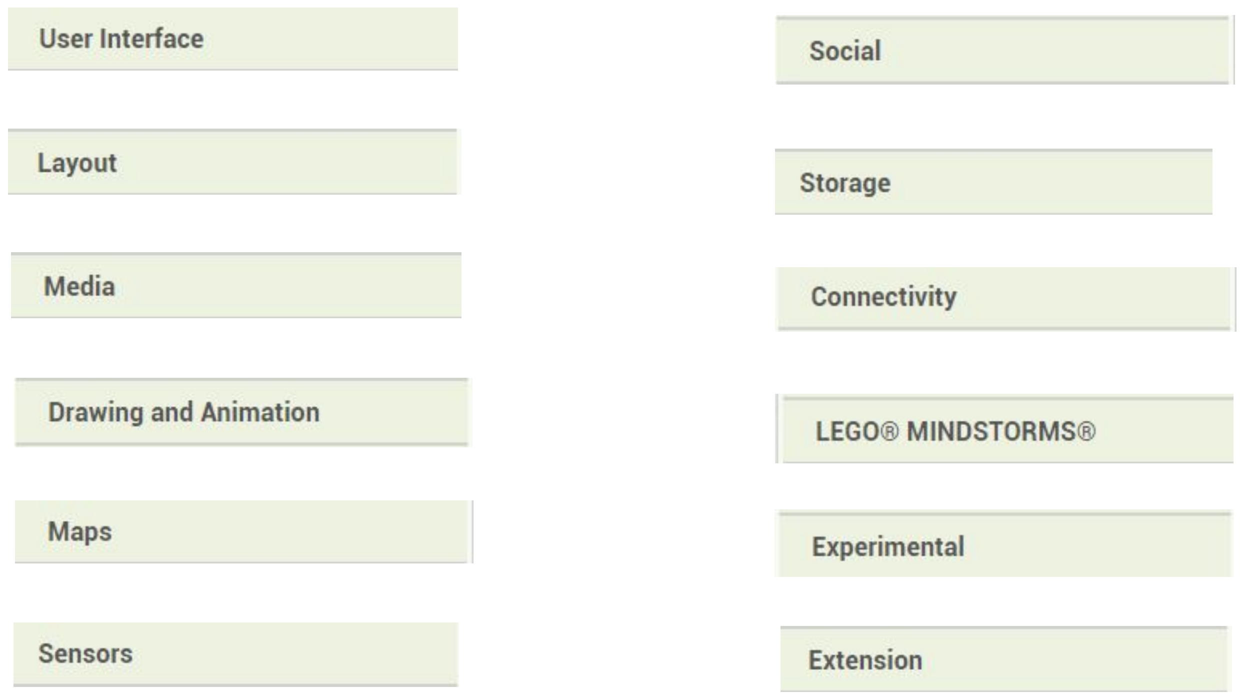
CloseScreenAnimation: Default

DefaultFileScope: App

HighContrast

Icon: None...

PALETS



User Interface

Layout

Media

Drawing and Animation

Maps

Sensors

Social

Storage

Connectivity

LEGO® MINDSTORMS®

Experimental

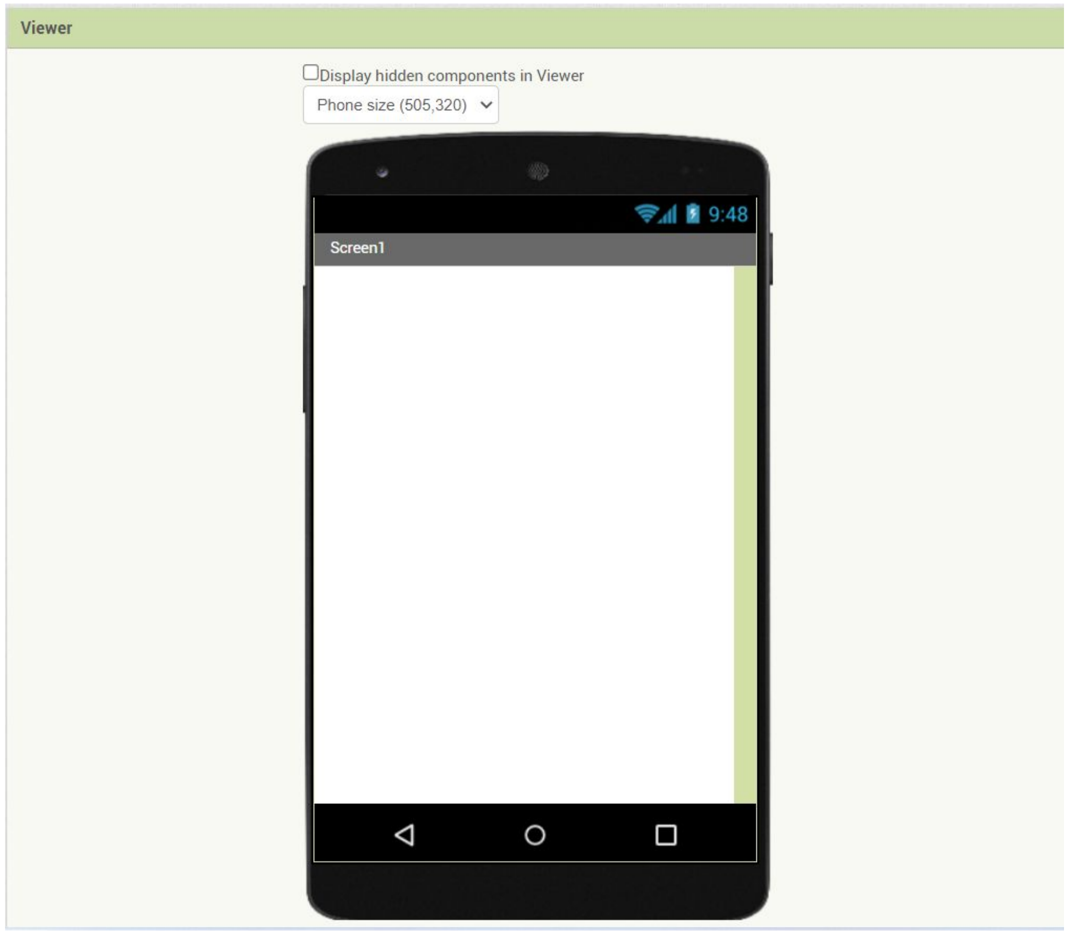
Extension

Application 1: TRANSLATOR

Palets:

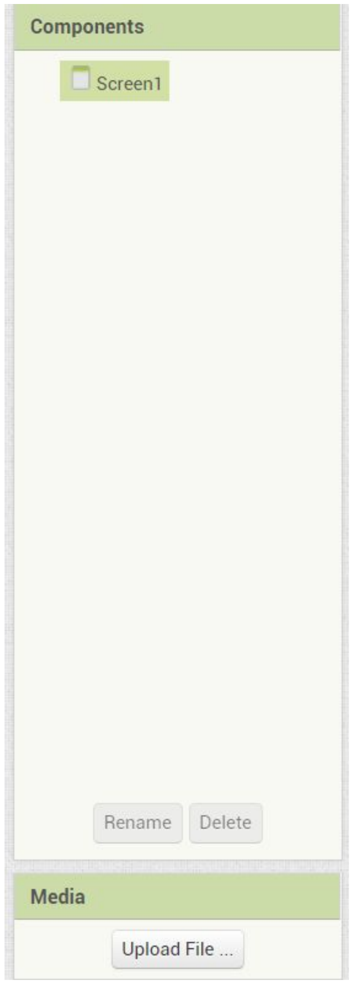
- The left side of the screen features the Palette , which, as the name implies, is the space for all the creation tools.

Application 1: TRANSLATOR



Viewer:

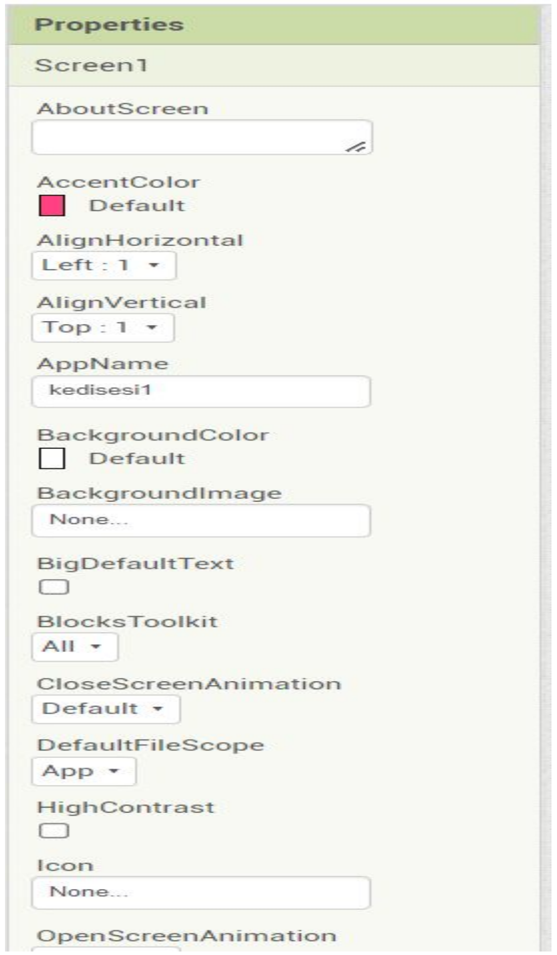
- App Inventor is the part that shows how the Mobile Application will appear on the user's screen.
- In the upper part, the size of the smartphone or tablet on which we will install the application is determined..



Application 1: TRANSLATOR

Components:

- In this section, we see the components that we will use in App Inventor.
- We can upload files from the media section.
- In the picture on the right you can see that the screen widget has been added.



Properties

Screen1

AboutScreen

AccentColor
🟥 Default

AlignHorizontal
Left : 1

AlignVertical
Top : 1

AppName
kedisesi1

BackgroundColor
☐ Default

BackgroundImage
None...

BigDefaultText
☐

BlocksToolkit
All

CloseScreenAnimation
Default

DefaultFileScope
App

HighContrast
☐

Icon
None...

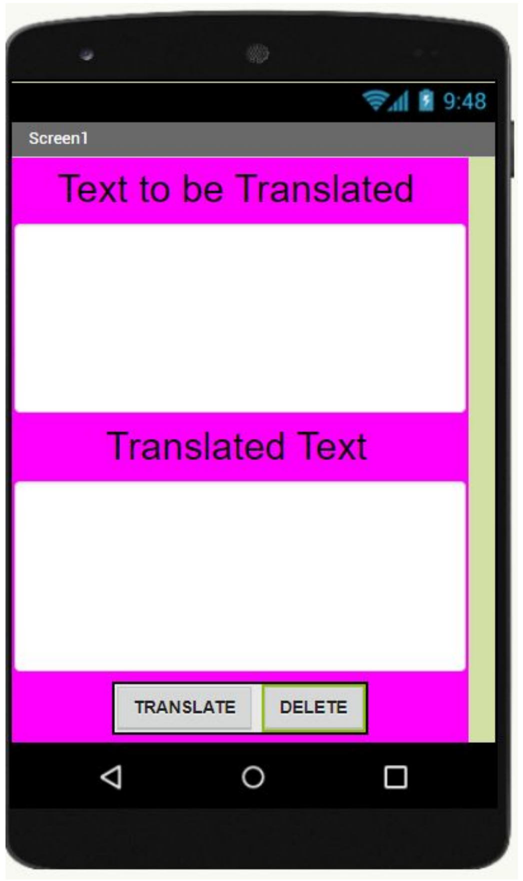
OpenScreenAnimation

Application 1: TRANSLATOR

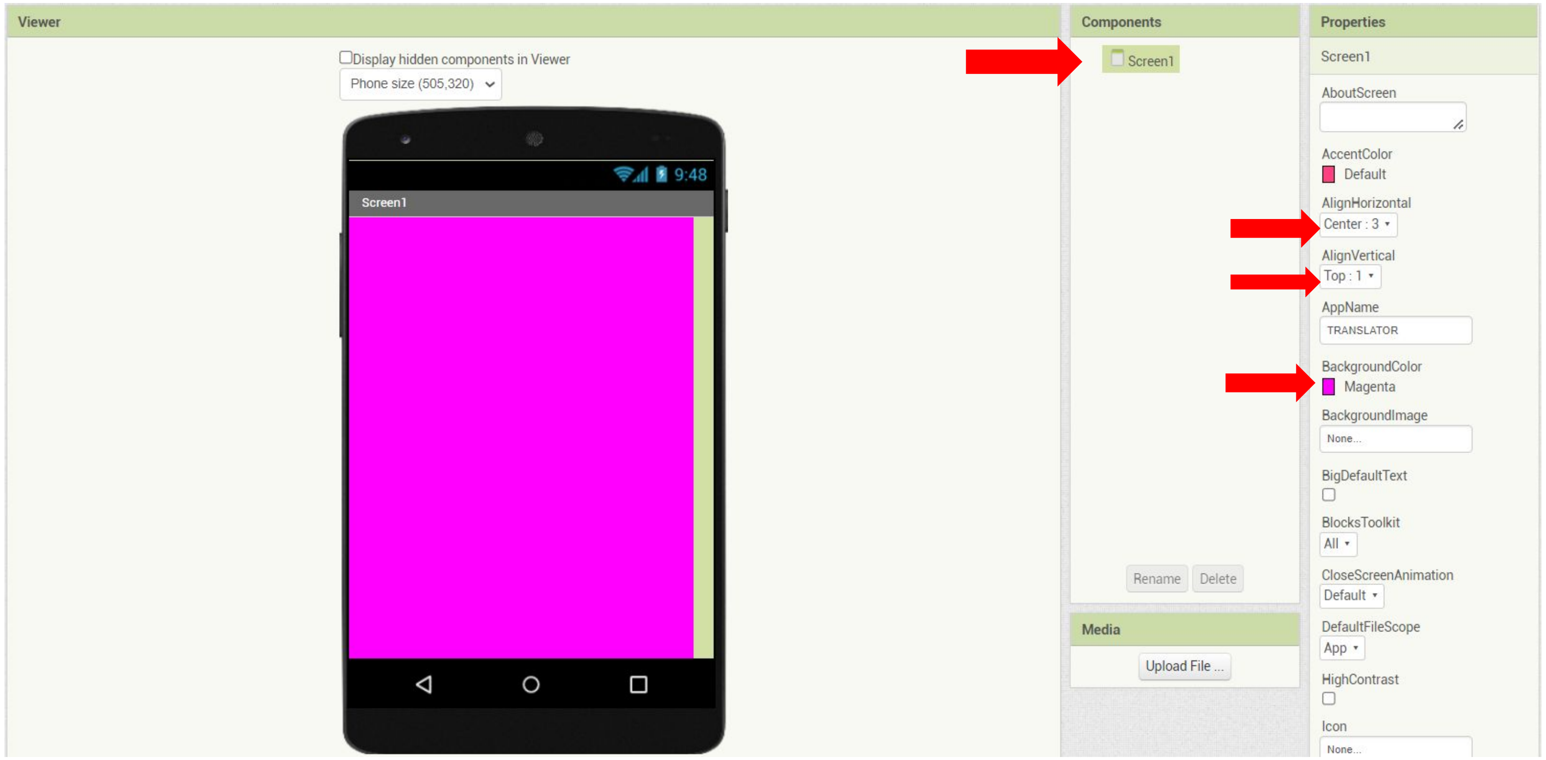
Properties:

- We can set the properties of the components we use in the properties section in App Inventor.
- In this section, we can assign the media files that we have uploaded to the component.

Application 1: TRANSLATOR



1-) We will make a translation application using App Inventor. We will make an application that translates the English text entered in the application into Turkish.


Viewer

Display hidden components in Viewer

Phone size (505,320)

Components

Screen1

Rename Delete

Media

Upload File ...

Properties

Screen1

AboutScreen

AccentColor
Default

AlignHorizontal
Center : 3

AlignVertical
Top : 1

AppName
TRANSLATOR

BackgroundColor
Magenta

BackgroundImage
None...

BigDefaultText

BlocksToolkit
All

CloseScreenAnimation
Default

DefaultFileScope
App

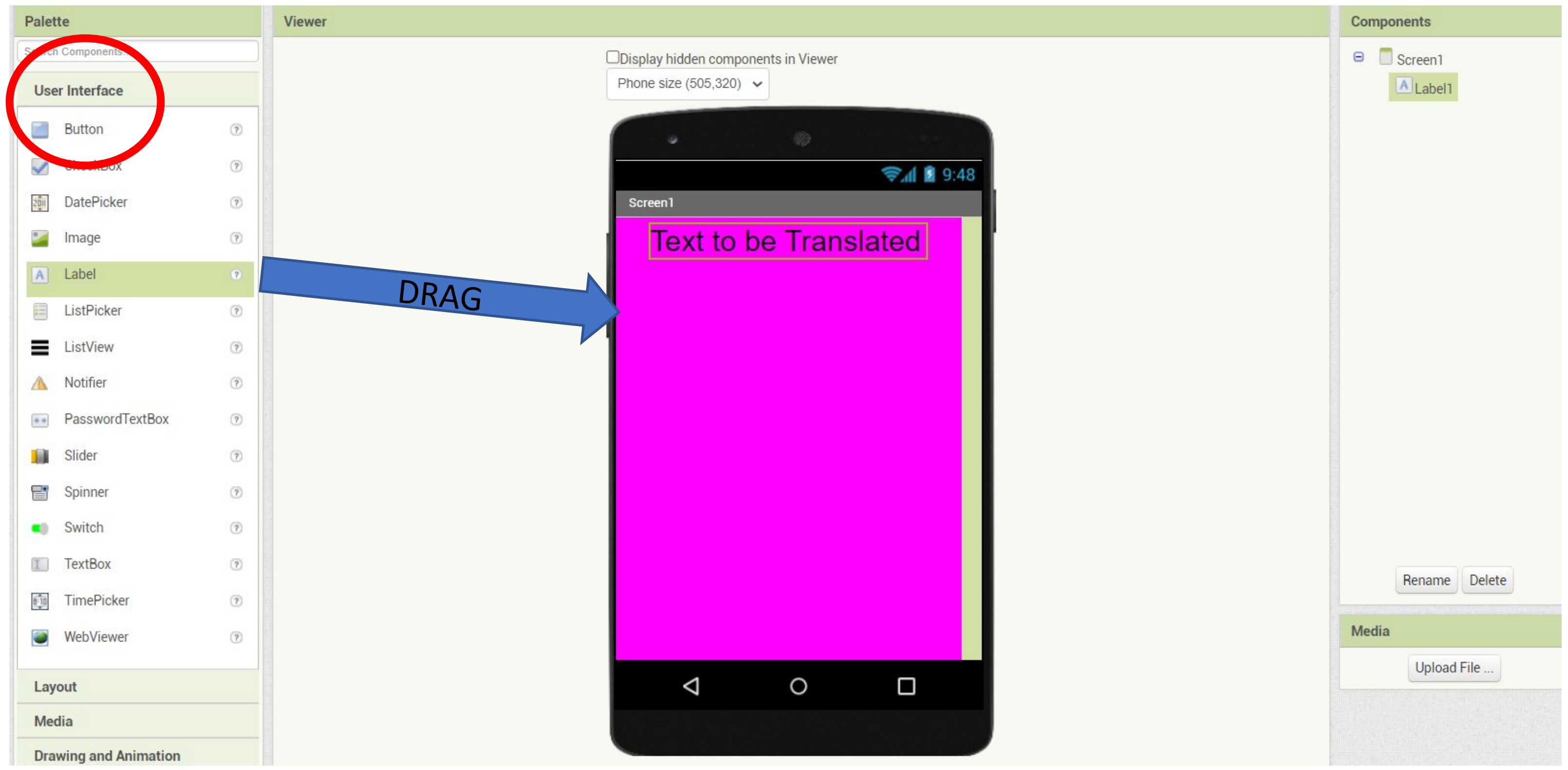
HighContrast

Icon
None...

Application 1: TRANSLATOR

1-)Screen1 Properties:

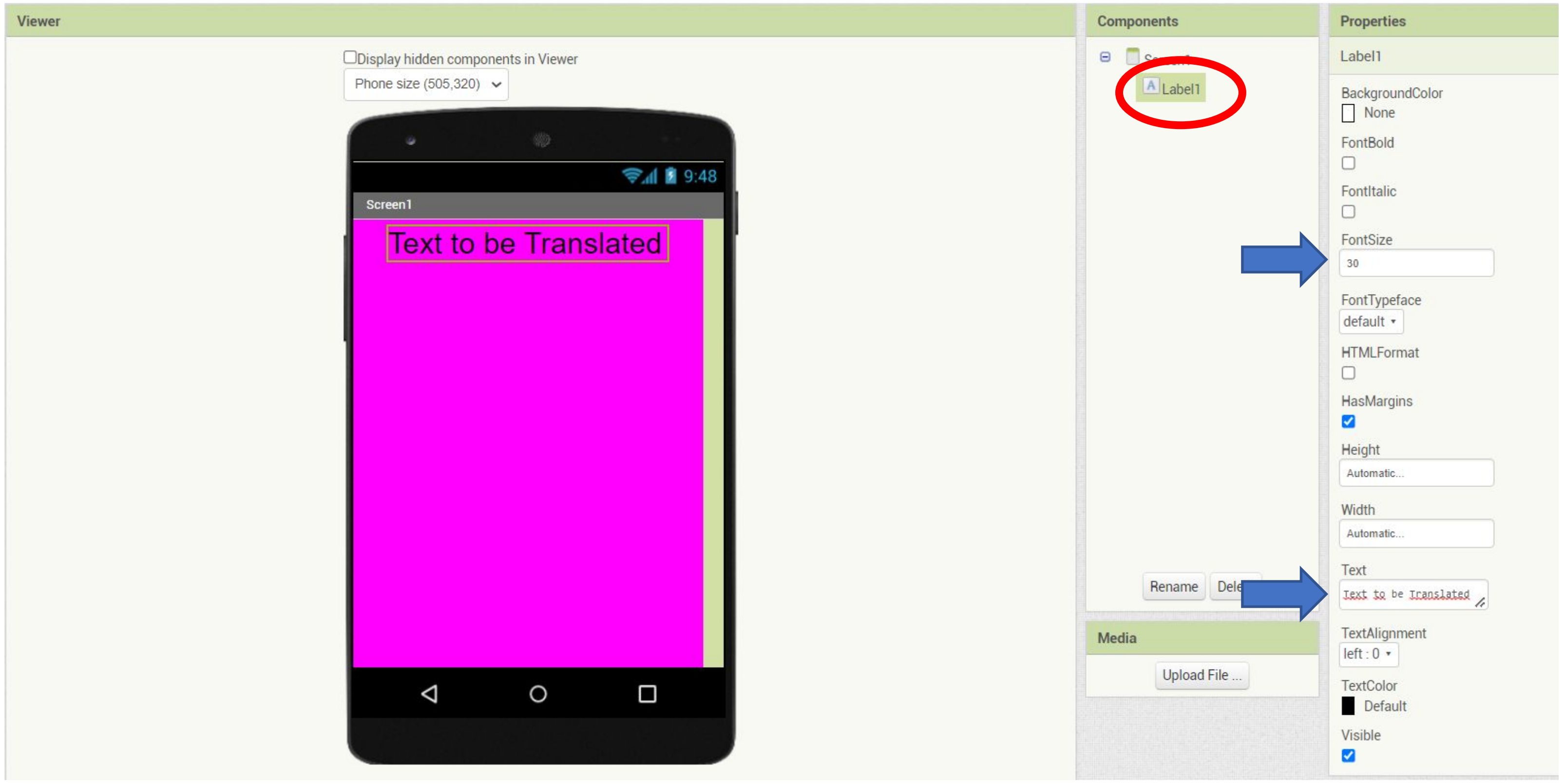
- **Let's choose "center" for the AlignHorizontal property.**
- **Let's select the AlignVertical property as "Top 1".**
- **Let's choose the backgroundColor color magneta.**



Application 1: TRANSLATOR

1-)DRAG LABEL:

Let's drag and drop the label component to the screen.



Viewer

Display hidden components in Viewer

Phone size (505,320)

Screen1

Text to be Translated

Components

Label1

Rename

Media

Upload File ...

Properties

Label1

BackgroundColor: None

FontBold: ☐

FontItalic: ☐

FontSize: 30

FontTypeface: default

HTMLFormat: ☐

HasMargins: ✓

Height: Automatic...

Width: Automatic...

Text: Text to be Translated

TextAlignment: left : 0

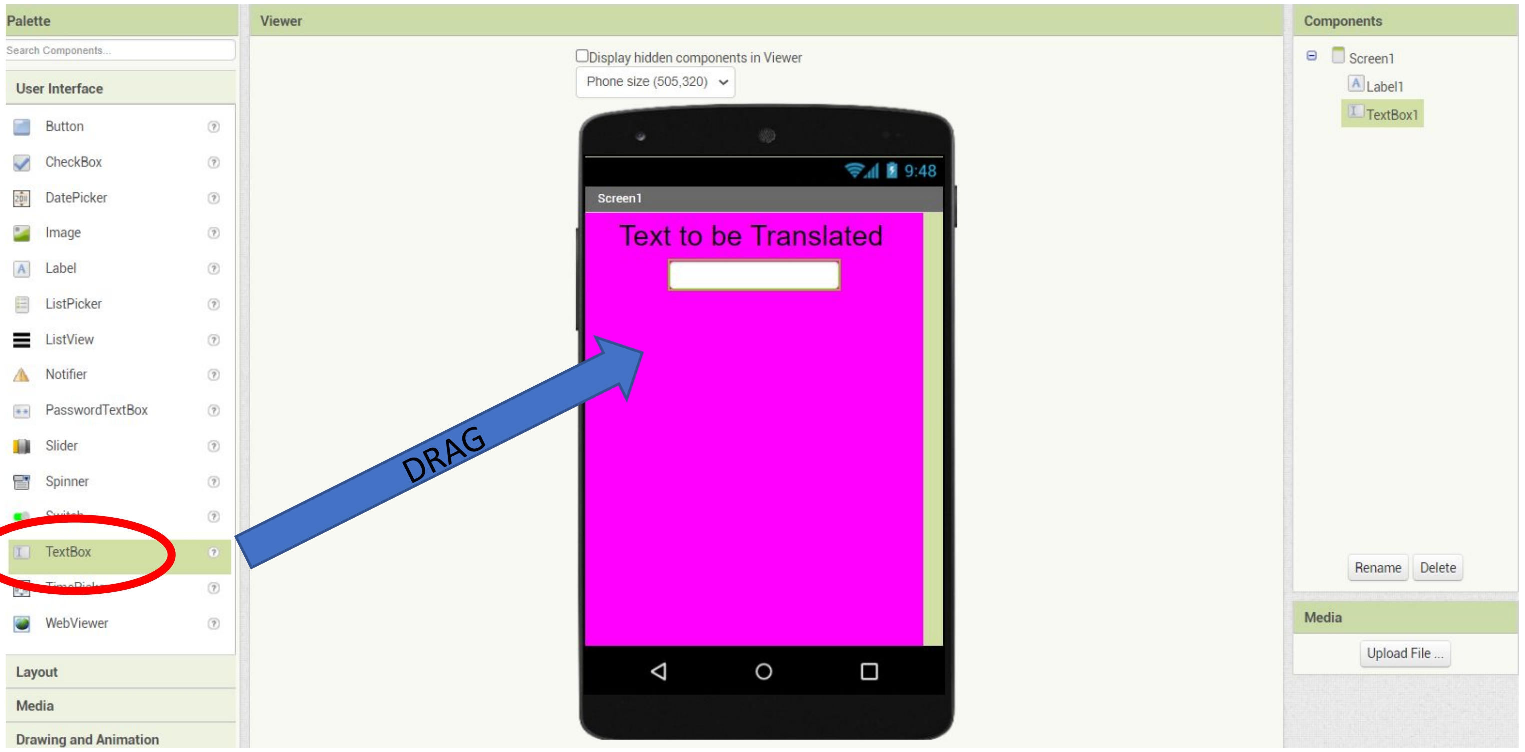
TextColor: Default

Visible: ✓

Application 1: TRANSLATOR

1-)PROPERTIES LABEL:

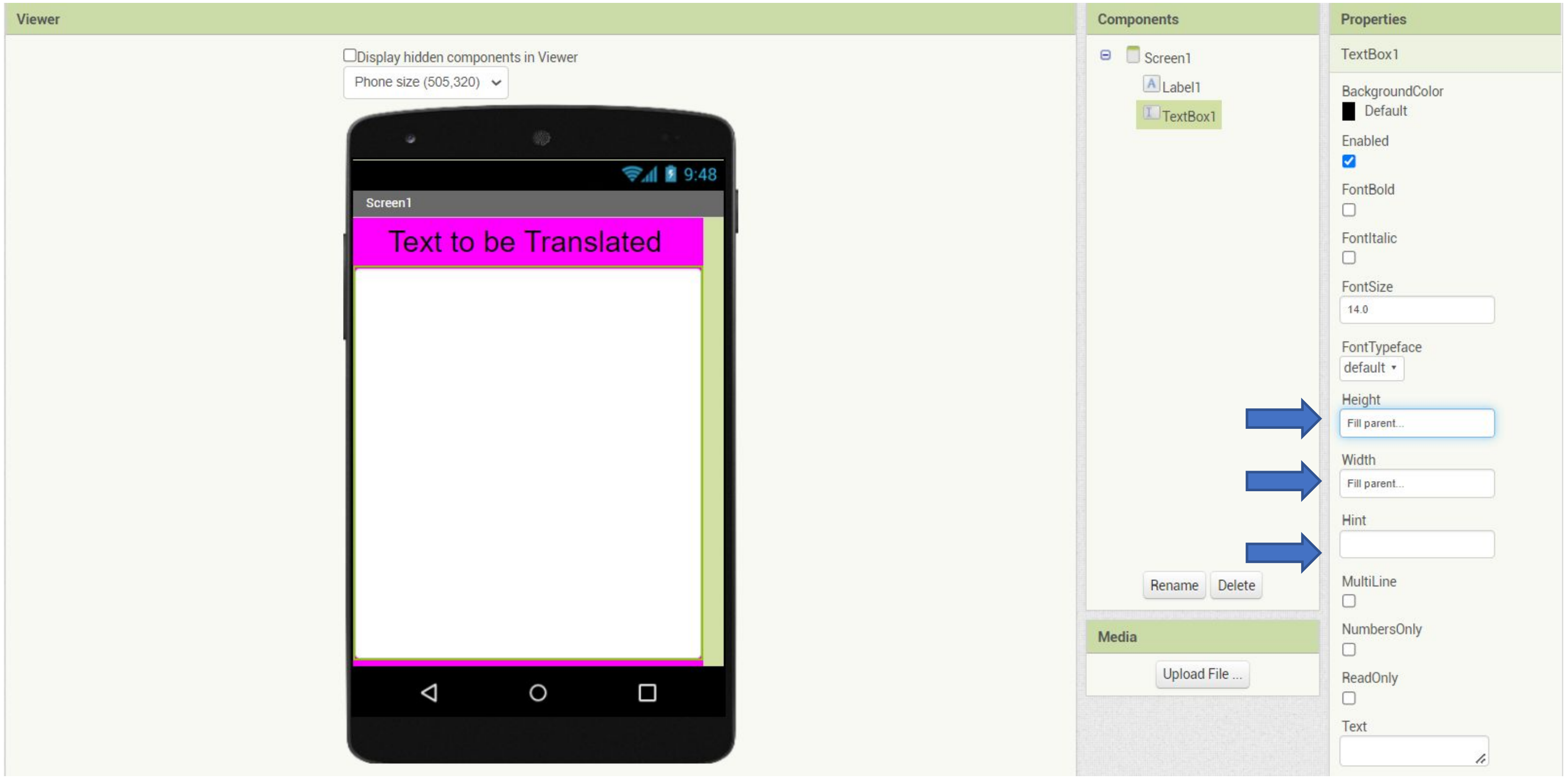
- **Change FontSize to ”30”**
- **Let's write the ”Text to be Translated” in the Text section.**



Application 1: TRANSLATOR

1-)DRAG TEXTBOX:

Let's drag and drop the TextBox component to the screen.


Viewer

Display hidden components in Viewer

Phone size (505,320)

Screen1

Text to be Translated

Components

Screen1

Label1

TextBox1

Rename Delete

Media

Upload File ...

Properties

TextBox1

BackgroundColor: Default

Enabled: ✓

FontBold: ☐

FontItalic: ☐

FontSize: 14.0

FontTypeface: default

Height: Fill parent...

Width: Fill parent...

Hint:

MultiLine: ☐

NumbersOnly: ☐

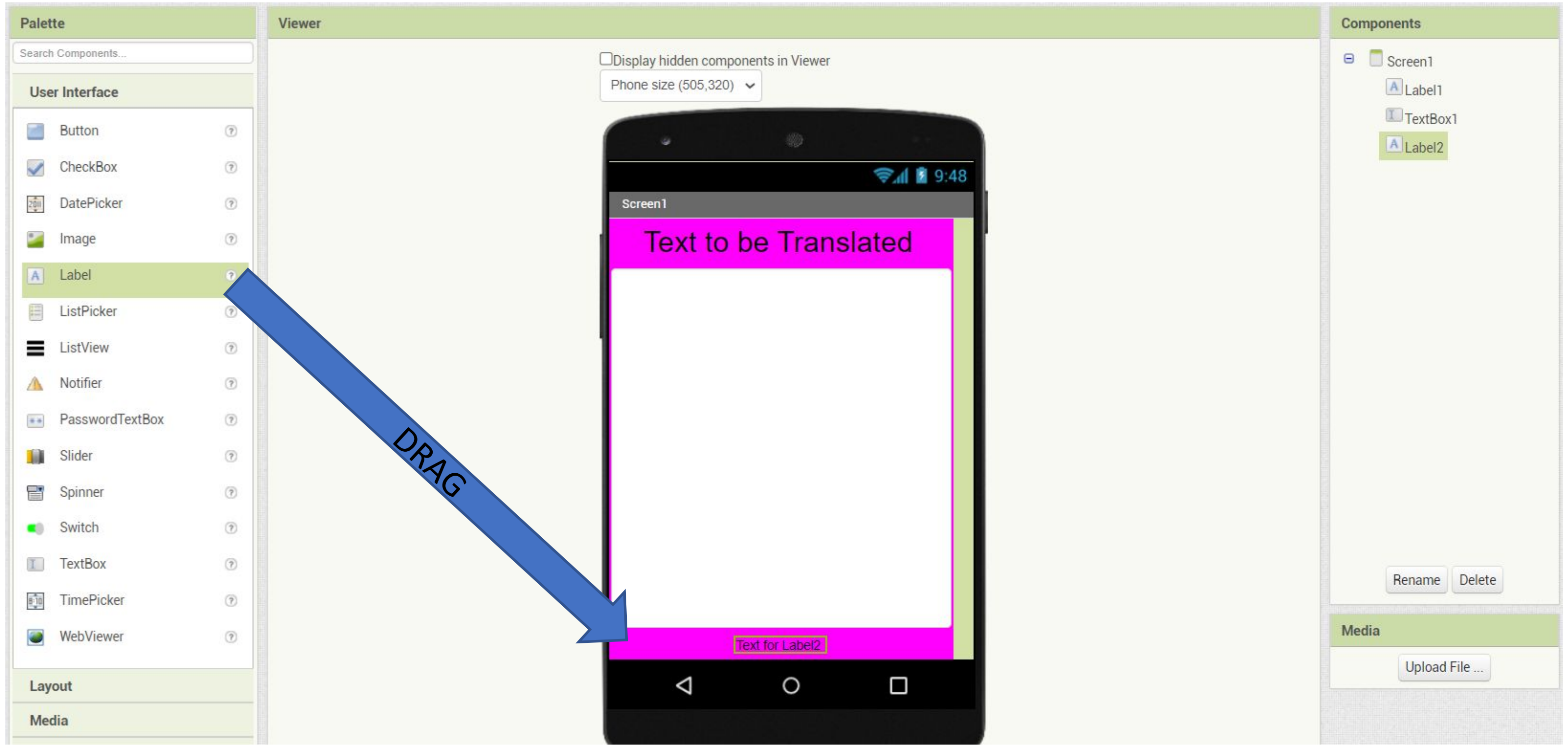
ReadOnly: ☐

Text:

Application 1: TRANSLATOR

1-)PROPERTIES TEXTBOX:

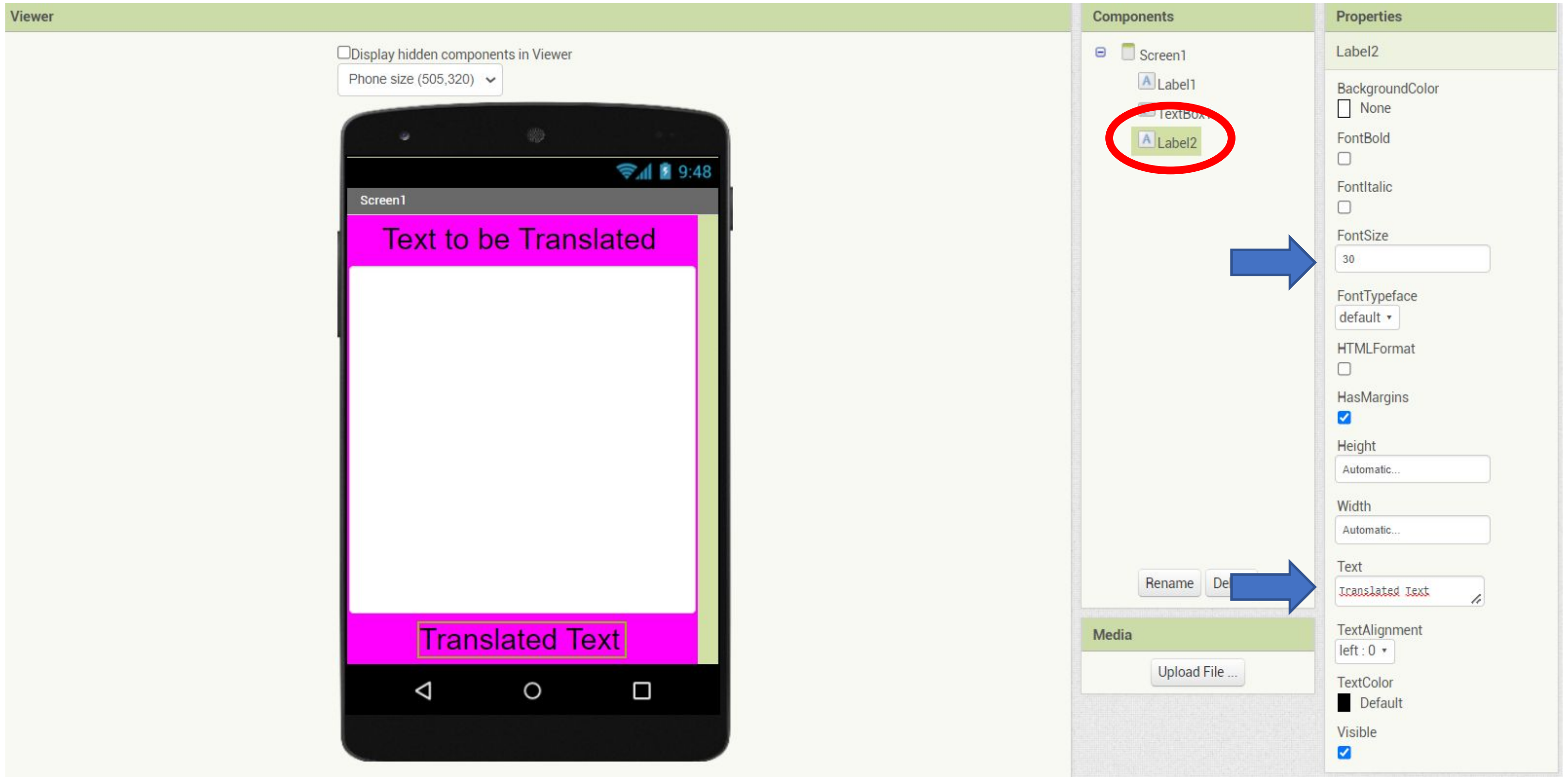
- **Let's choose height to fill parent.**
- **Select the width to fill parent.**
- **Let's delete the text in the Hint menu.**



Application 1: TRANSLATOR

1-)DRAG LABEL:

Let's drag and drop the label 2 component to the screen.


Viewer

Display hidden components in Viewer

Phone size (505,320)

Screen1

Text to be Translated

Translated Text

Components

Screen1

Label1

TextBox1

Label2

Rename Delete

Media

Upload File ...

Properties

Label2

BackgroundColor: None

FontBold: ☐

FontItalic: ☐

FontSize: 30

FontTypeface: default

HTMLFormat: ☐

HasMargins: ✓

Height: Automatic...

Width: Automatic...

Text: Translated Text

TextAlignment: left : 0

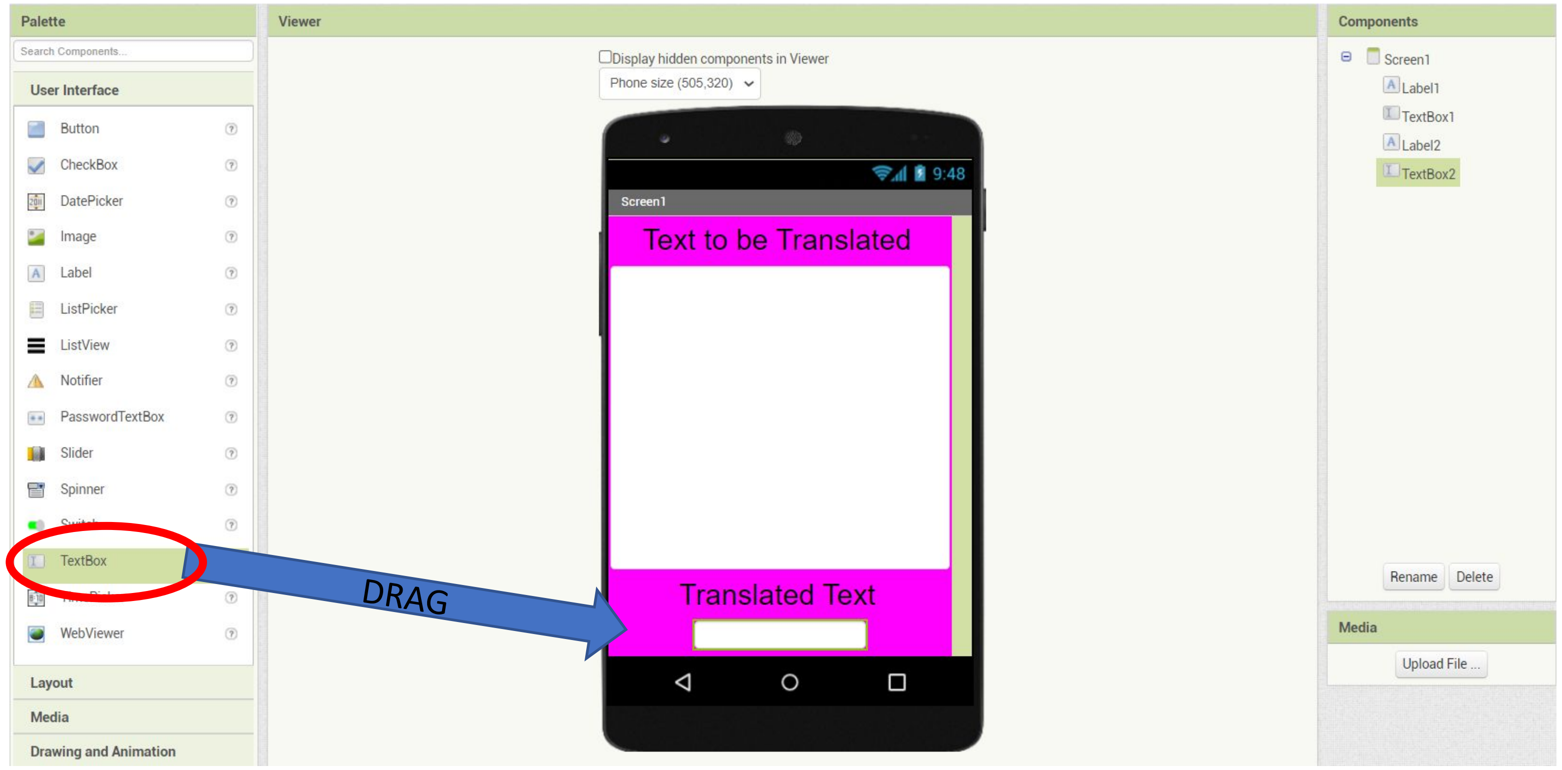
TextColor: Default

Visible: ✓

Application 1: TRANSLATOR

1-)PROPERTIES LABEL:

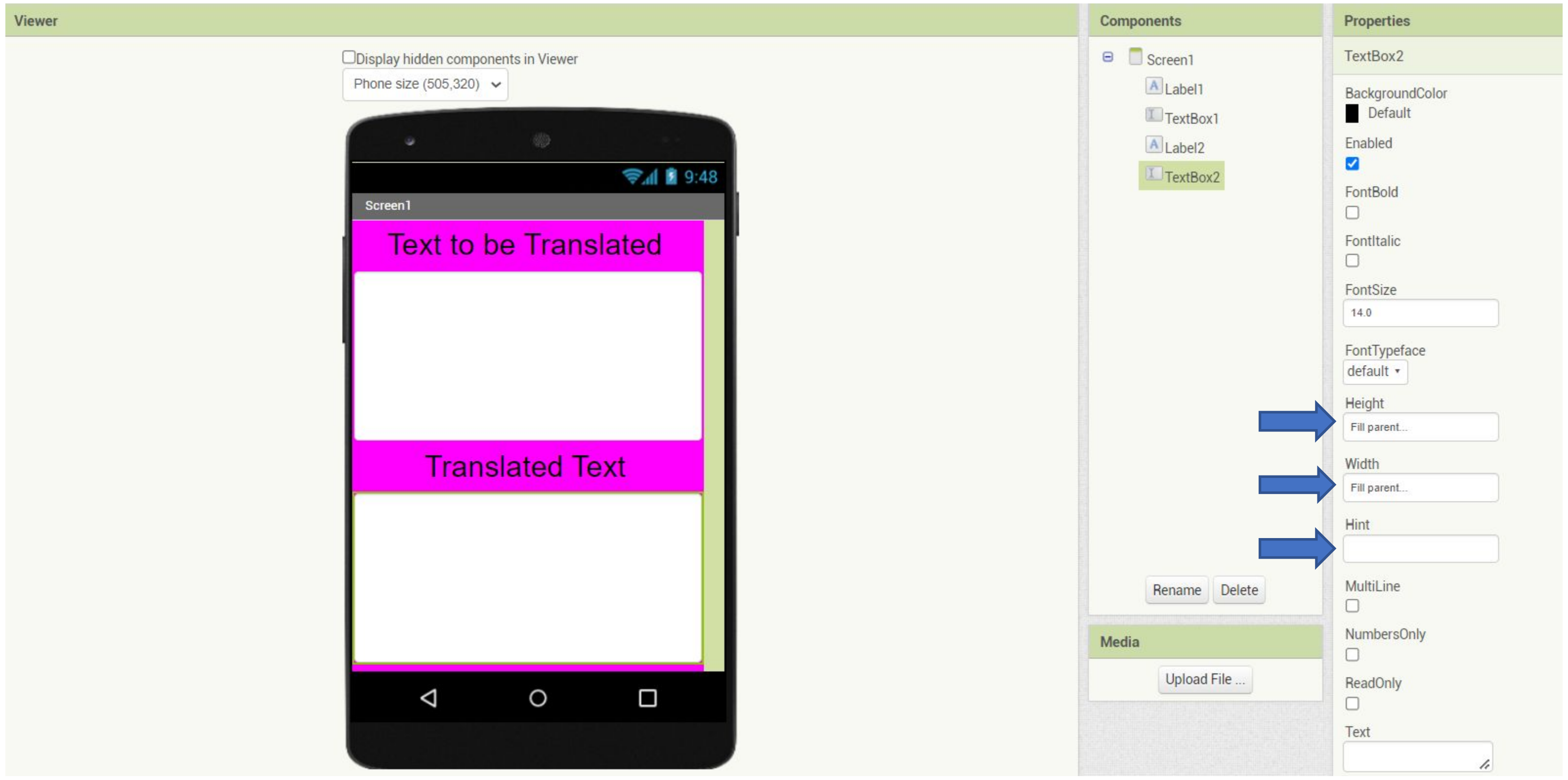
- **Change FontSize to ''30''**
- **Let's write the ''Translated Text'' in the Text section.**



Application 1: TRANSLATOR

1-)DRAG TEXTBOX 2:

Let's drag and drop the TextBox component to the screen.


Viewer

Display hidden components in Viewer

Phone size (505,320)

Screen1

Text to be Translated

Translated Text

Components

Screen1
Label1
TextBox1
Label2
TextBox2

Rename Delete

Media

Upload File ...

Properties

TextBox2

BackgroundColor
Default

Enabled ✓

FontBold

FontItalic

FontSize
14.0

FontTypeface
default

Height
Fill parent...

Width
Fill parent...

Hint

MultiLine

NumbersOnly

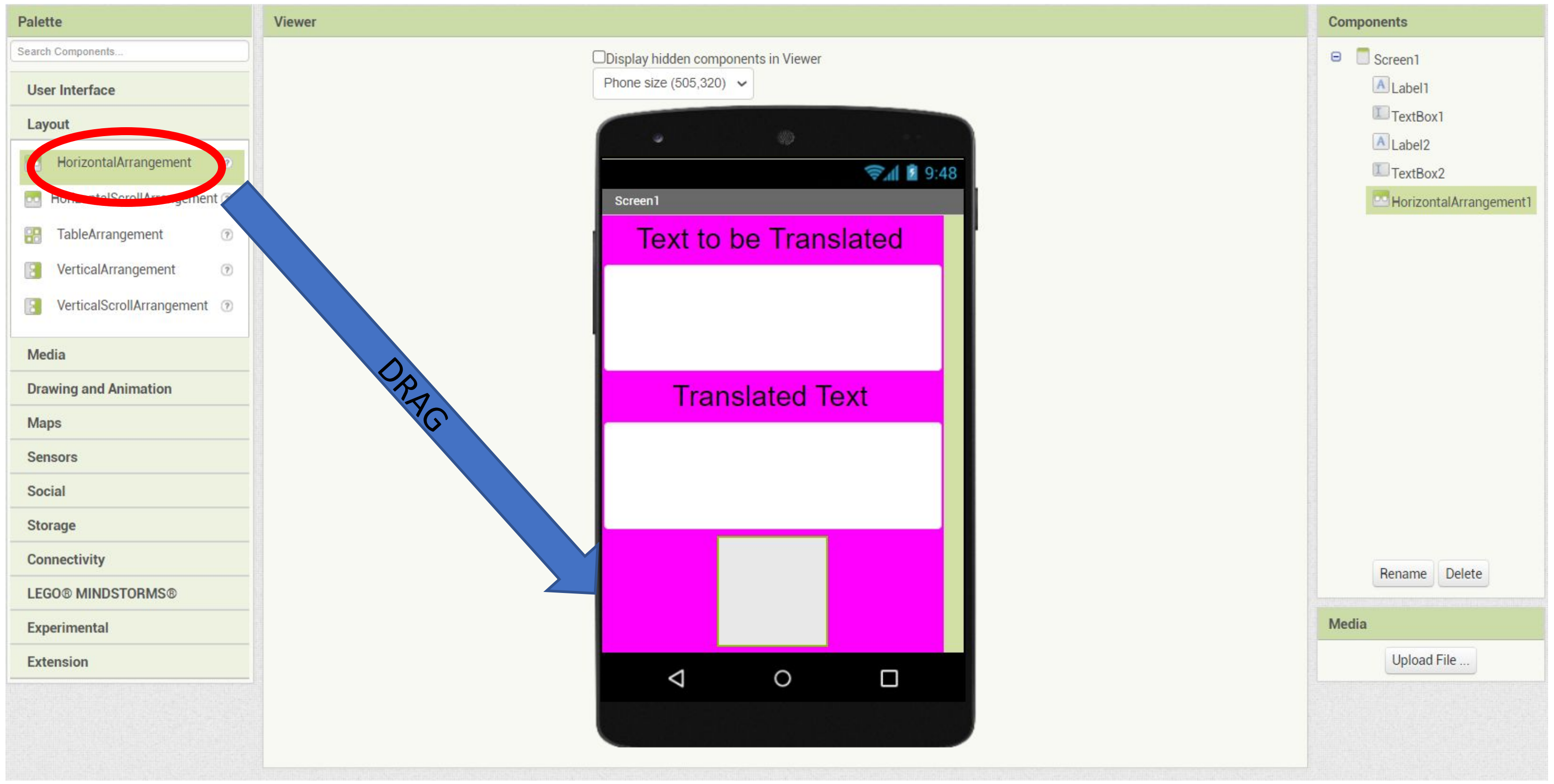
ReadOnly

Text

Application 1: TRANSLATOR

1-)PROPERTIES TEXTBOX 2:

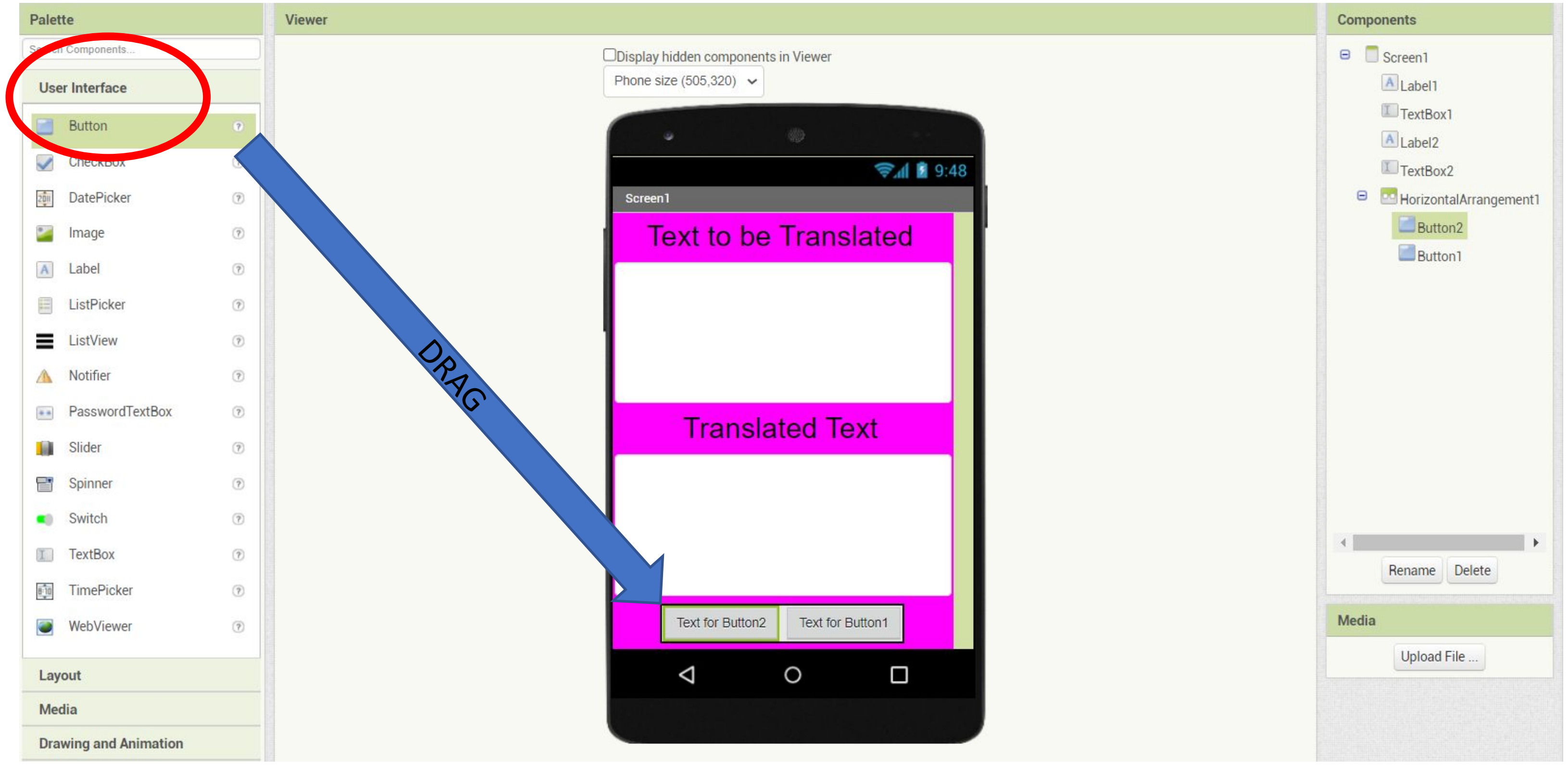
- **Let's choose height to fill parent.**
- **Select the width to fill parent.**
- **Let's delete the text in the Hint menu.**



Application 1: TRANSLATOR

1-)DRAG HORIZONTAL ARRANGEMENT:

Let's drag and drop the HorizontalArrangement component to the screen.

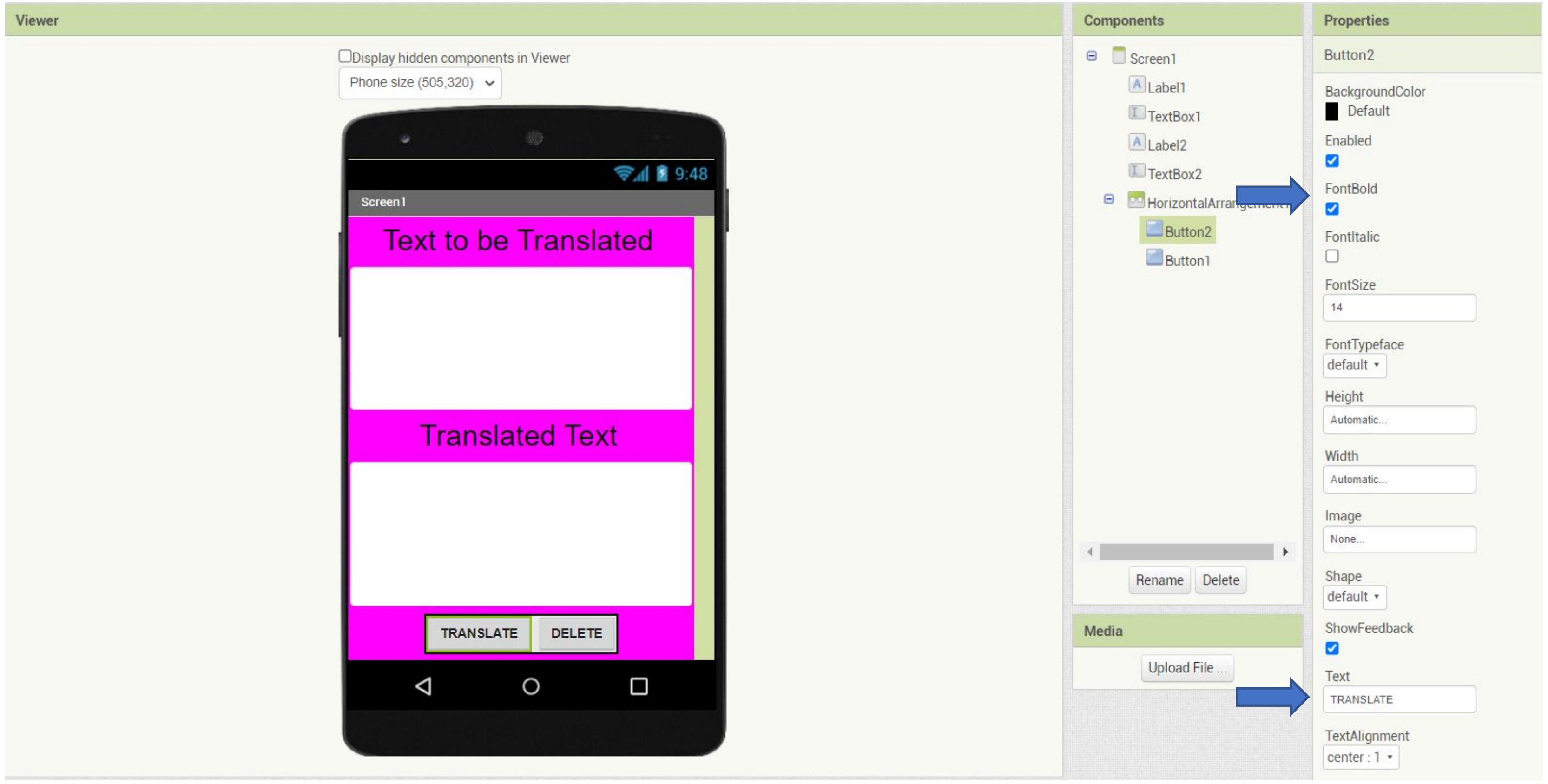


Application 1: TRANSLATOR

1-)DRAG BUTTON COMPONENTS:

Let's drag and drop the TWO BUTTON component into the HorrizontalArrangement component.

Application 1: TRANSLATOR


Viewer

Display hidden components in Viewer

Phone size (505,320)

Components

Screen1
Label1
TextBox1
Label2
TextBox2
HorizontalArrangement1
Button2
Button1

Rename Delete

Media

Upload File ...

Properties

Button2

BackgroundColor: Default

Enabled: ✓

FontBold: ✓

FontItalic: ☐

FontSize: 14

FontTypeface: default

Height: Automatic...

Width: Automatic...

Image: None...

Shape: default

ShowFeedback: ✓

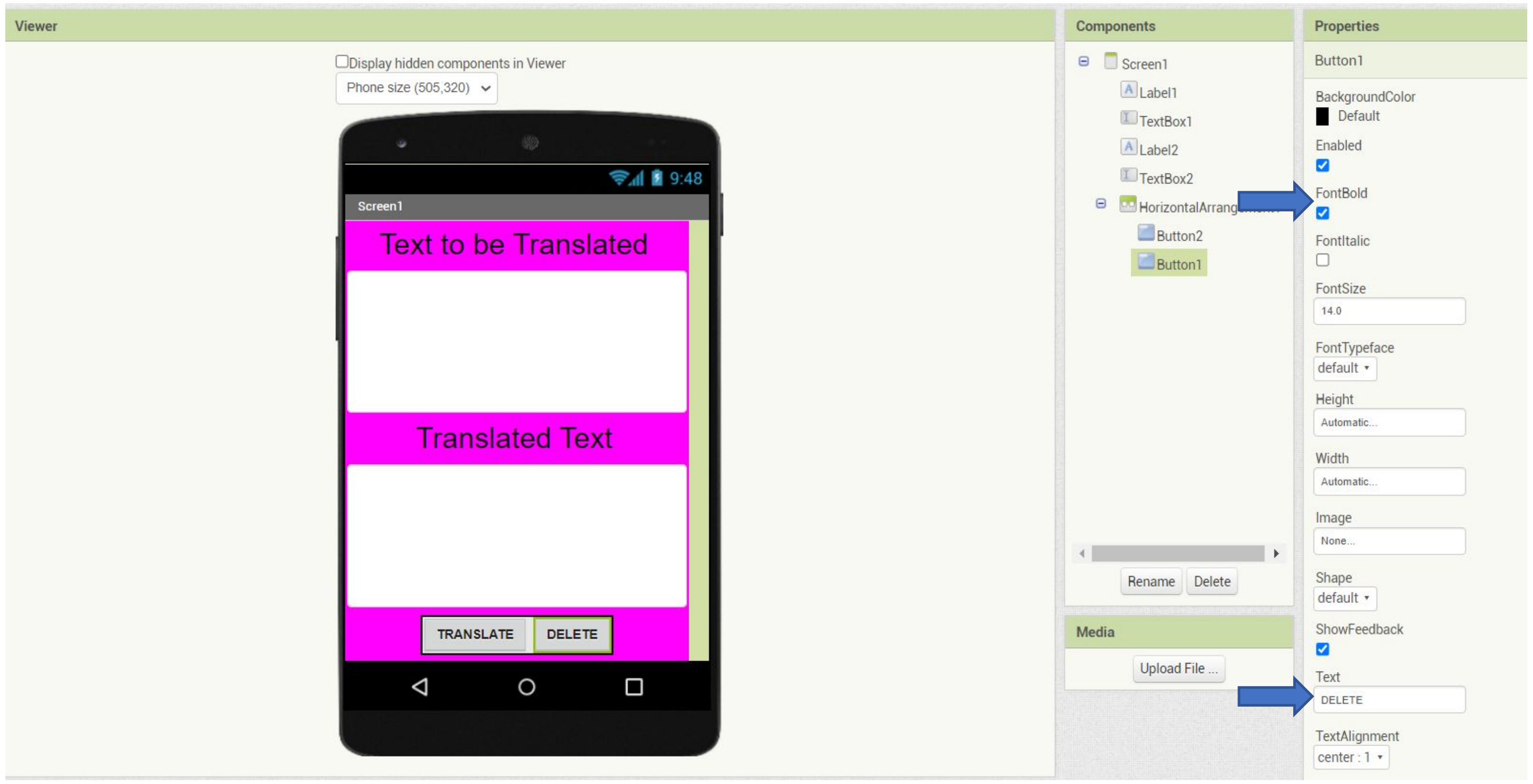
Text: TRANSLATE

TextAlignment: center : 1

1-)PROPERTIES BUTTON COMPONENT:

- Select the FONT to FontBold.
- Let's write the ”Translate” in the Text section.

Application 1: TRANSLATOR

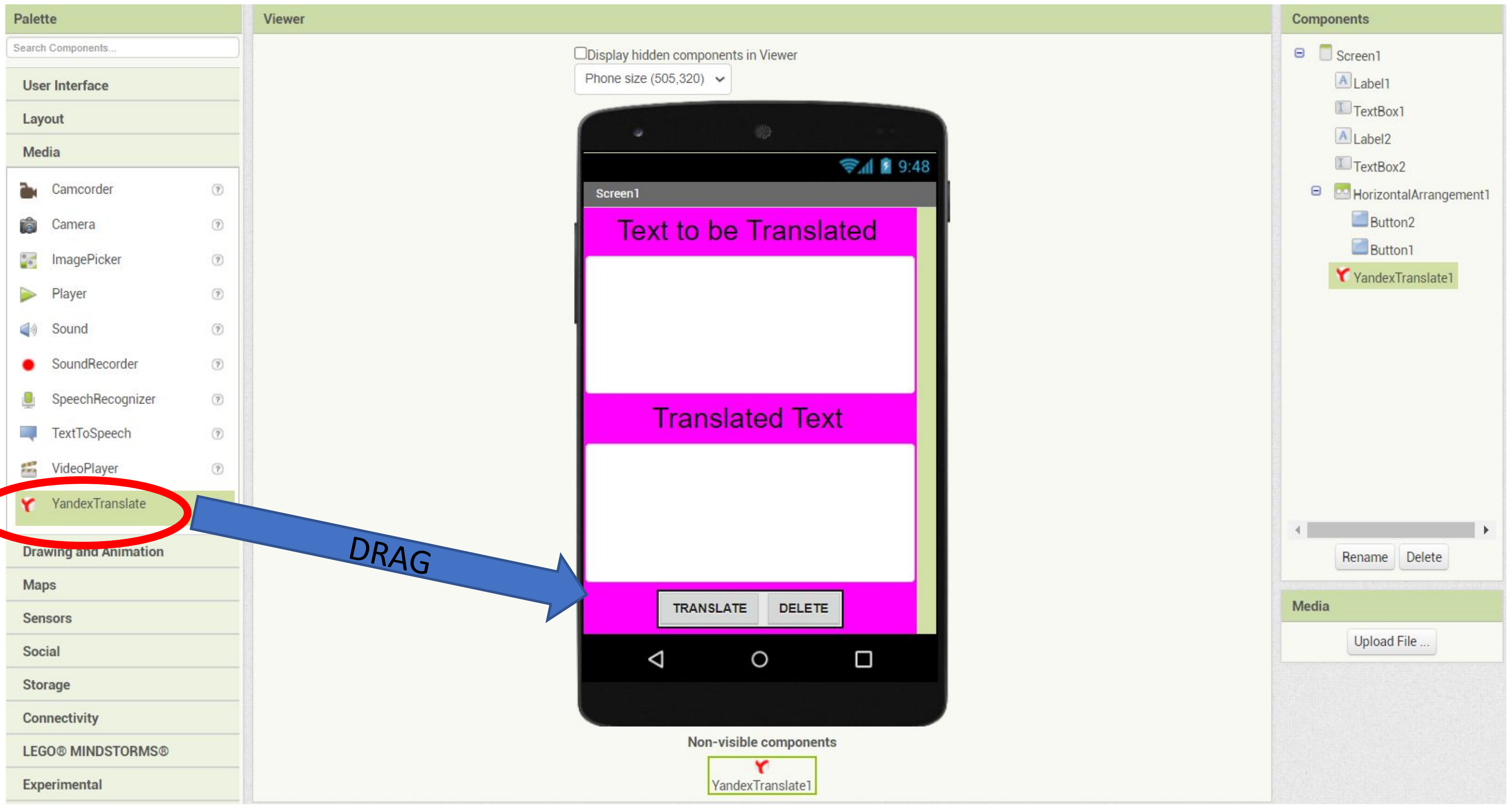


Button1

BackgroundColor: Default
Enabled: ✓
FontBold: ✓
FontItalic: ☐
FontSize: 14.0
FontTypeface: default
Height: Automatic...
Width: Automatic...
Image: None...
Shape: default
ShowFeedback: ✓
Text: DELETE
TextAlignment: center : 1

1-)PROPERTIES BUTTON COMPONENT:

- **Select the FONT to FontBold.**
- **Let's write the "Delete" in the Text section.**

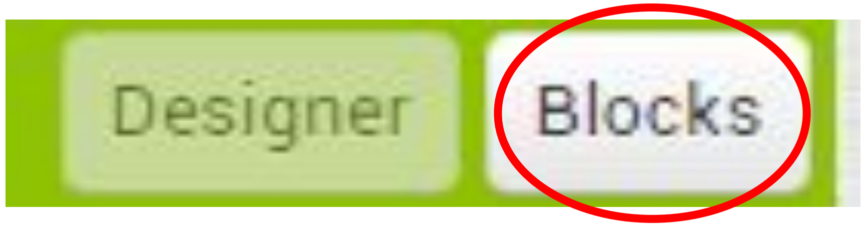


Application 1: TRANSLATOR

1-)DRAG YANDEX TRANSLATE COMPONENT:

Let's drag the Yandex translate component from the media menu and drop it on the screen.

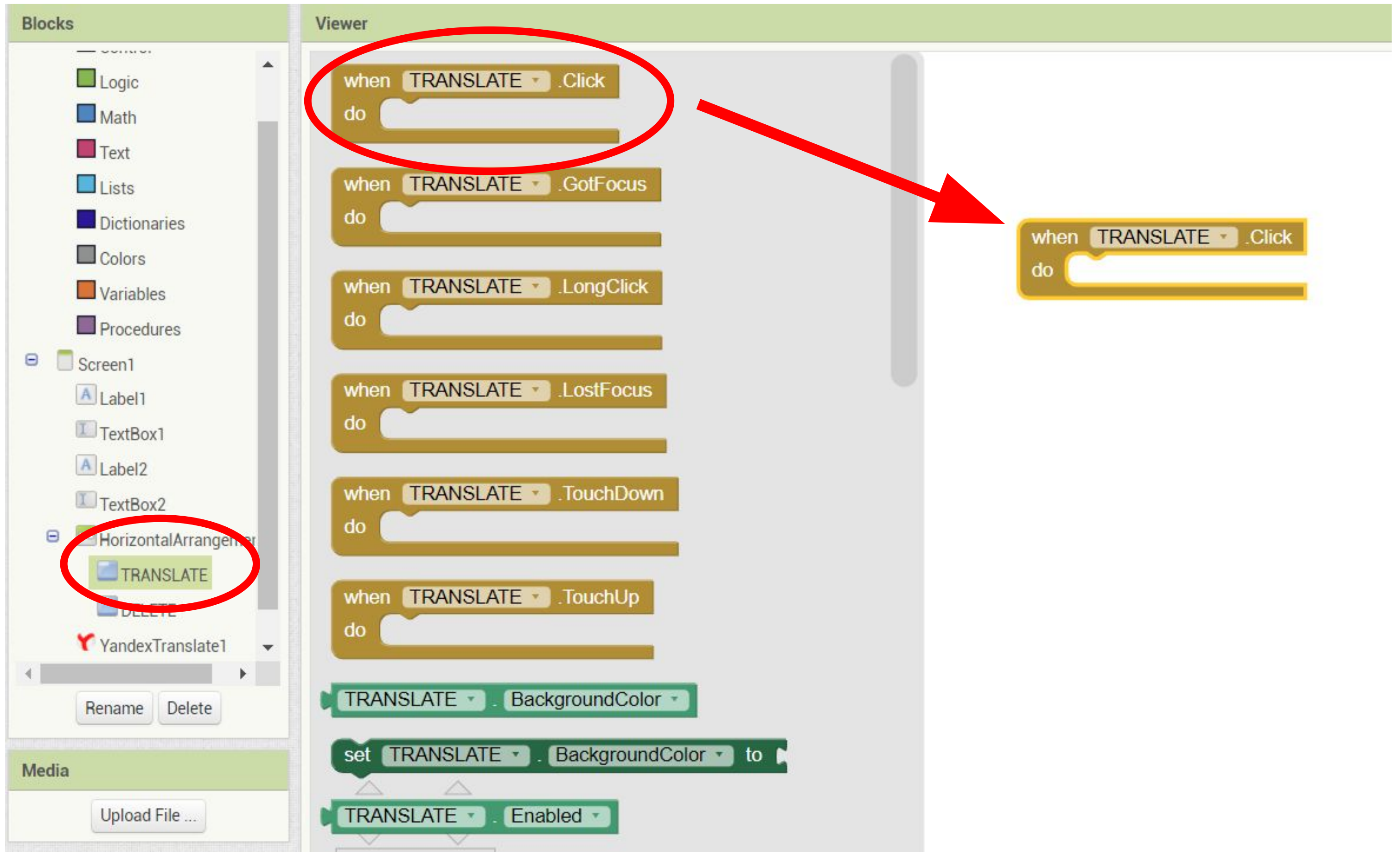
AI WITH APP INVENTOR



Application 1: TRANSLATOR

1-)PROGRAMMING TRANSLATOR:

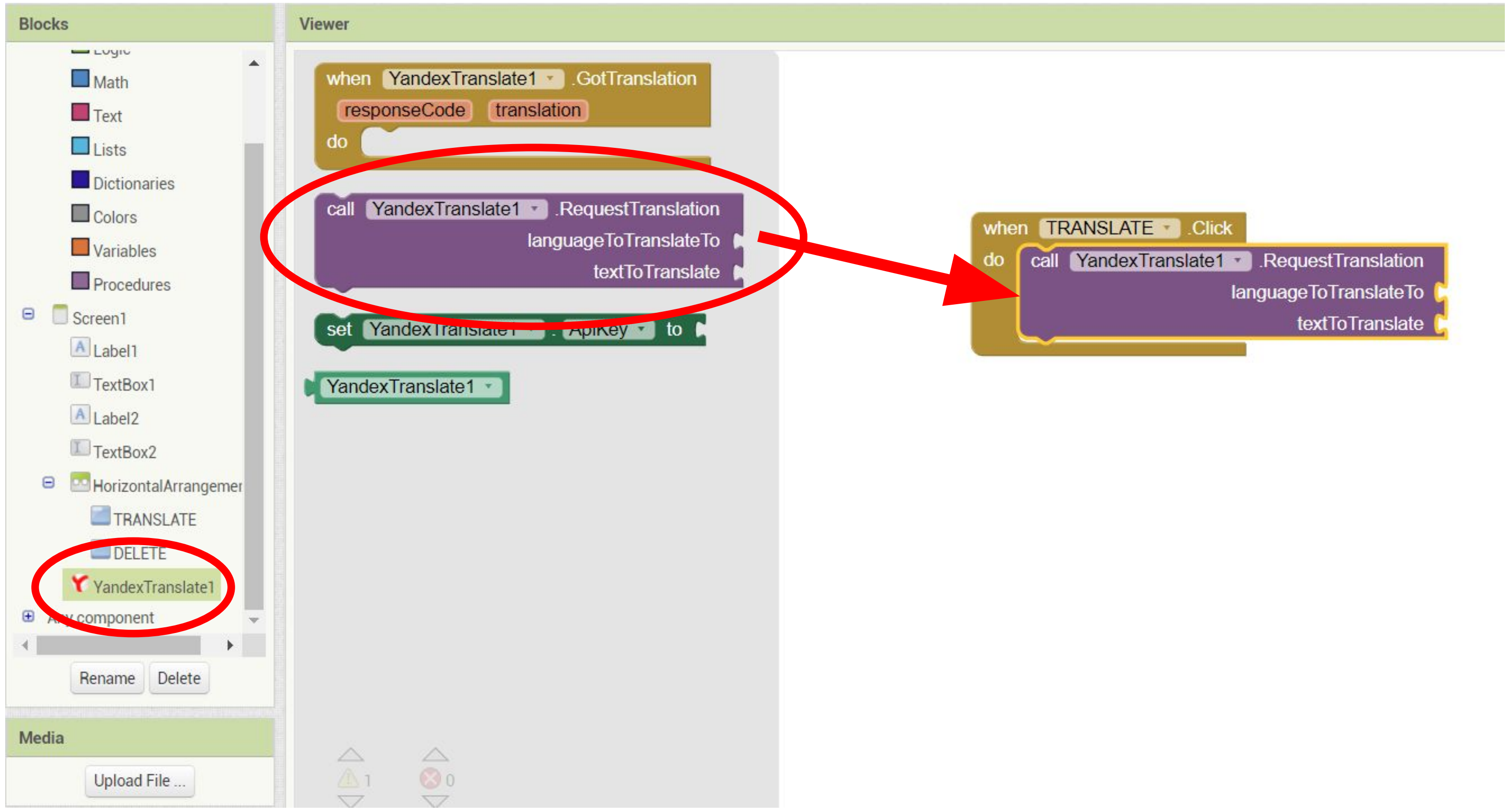
Select the BLOCKS menu at the top right of the screen.



Application 1: TRANSLATOR

1-)PROGRAMMING TRANSLATOR:

- **First let's code the translate button.**
- **When you click on Translate button, the text in the text to be translated into the TextBox is translated into Turkish with the YandexTranslate tool.**



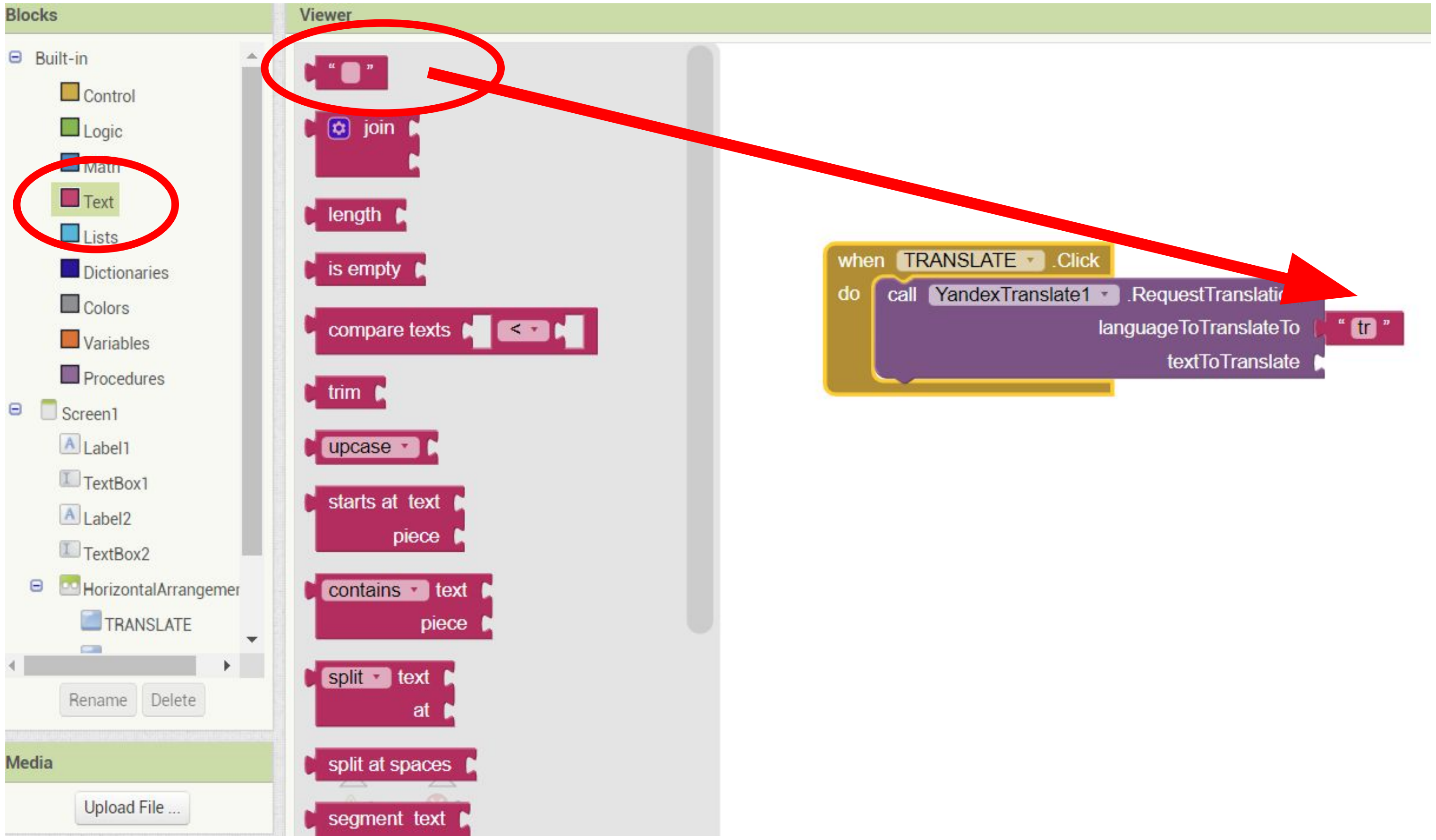
```
when YandexTranslate1 .GotTranslation
  responseCode  translation
do

call YandexTranslate1 .RequestTranslation
    languageToTranslateTo
    textToTranslate

set YandexTranslate1 . ApiKey to

YandexTranslate1

when TRANSLATE .Click
do  call YandexTranslate1 .RequestTranslation
        languageToTranslateTo
        textToTranslate
```

Application 1: TRANSLATOR

1-)PROGRAMMING TRANSLATOR:

- **First let's code the translate button.**

- **When you click on Translate button, the text in the text to be translated into the TextBox is translated into Turkish with the YandexTranslate tool.**

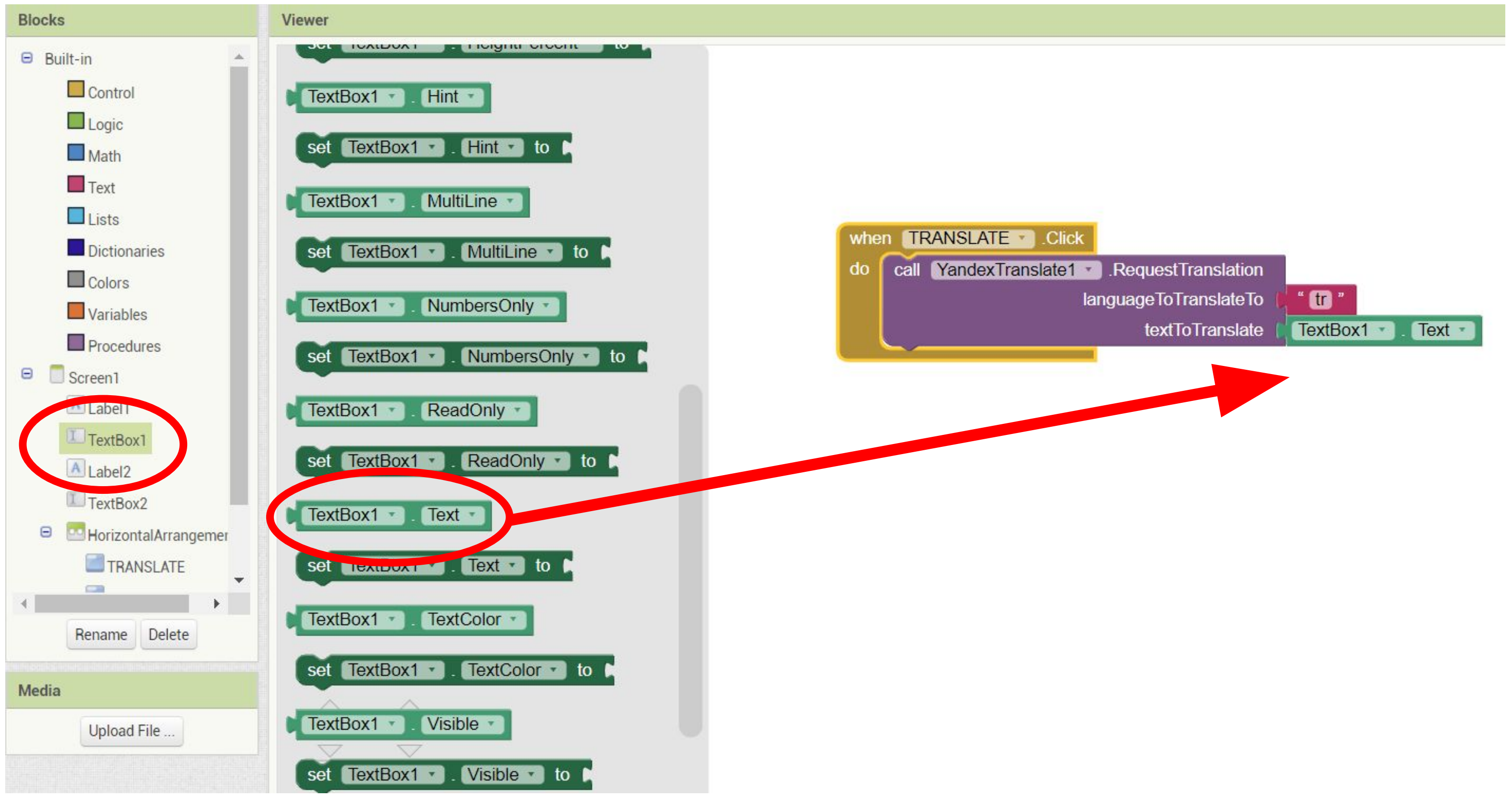


```
when TRANSLATE .Click
do  call YandexTranslate1 .RequestTranslati
        languageToTranslateTo  " tr "
        textToTranslate
```

Application 1: TRANSLATOR

1-)PROGRAMMING TRANSLATOR:

- Let's write to "tr" into the string block.

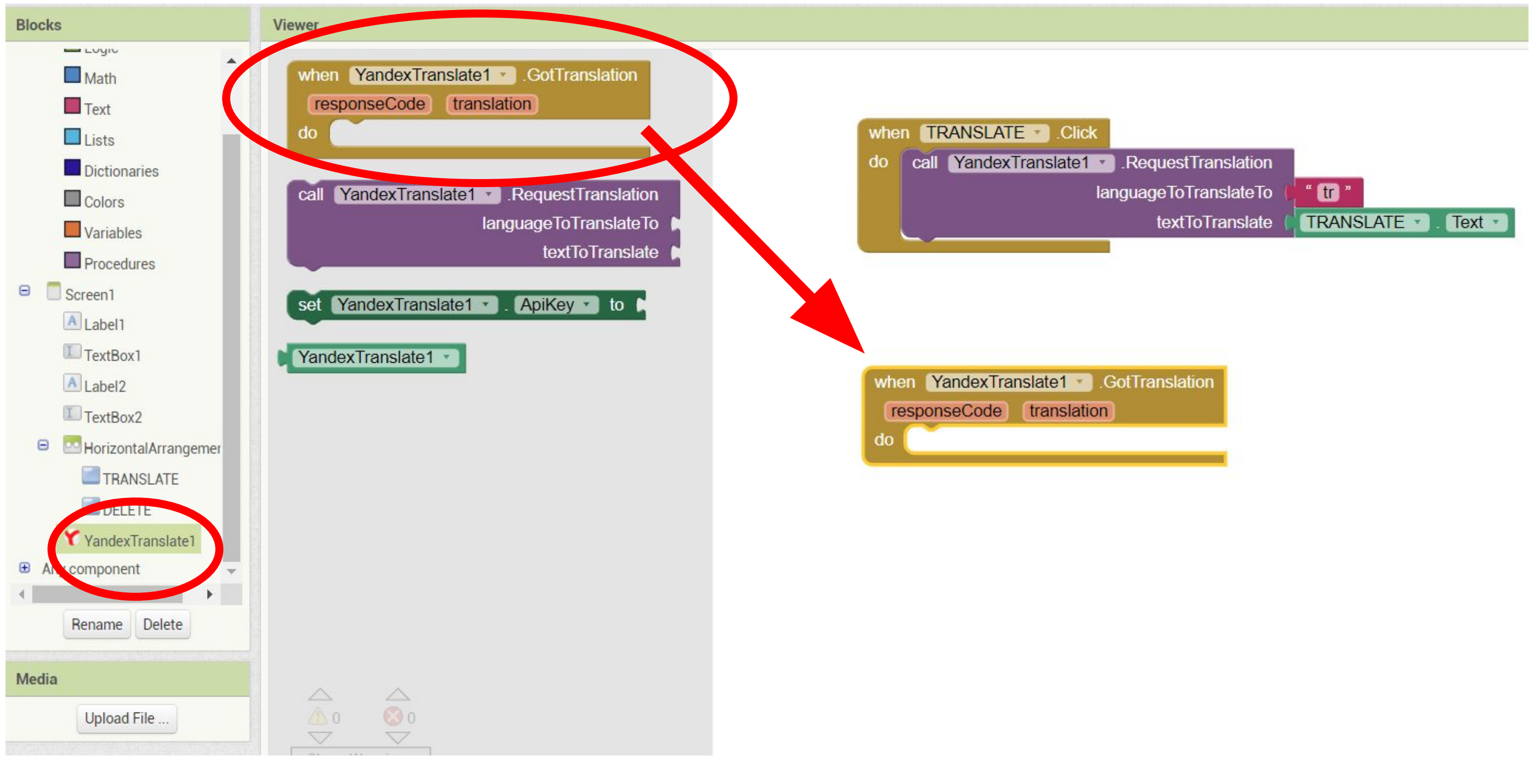


```
TextBox1 . Hint
set TextBox1 . Hint to
TextBox1 . MultiLine
set TextBox1 . MultiLine to
TextBox1 . NumbersOnly
set TextBox1 . NumbersOnly to
TextBox1 . ReadOnly
set TextBox1 . ReadOnly to
TextBox1 . Text
set TextBox1 . Text to
TextBox1 . TextColor
set TextBox1 . TextColor to
TextBox1 . Visible
set TextBox1 . Visible to

when TRANSLATE .Click
do  call YandexTranslate1 .RequestTranslation
                languageToTranslateTo  " tr "
                textToTranslate  TextBox1 . Text
```

Application 1: TRANSLATOR

1-)PROGRAMMING TRANSLATOR:

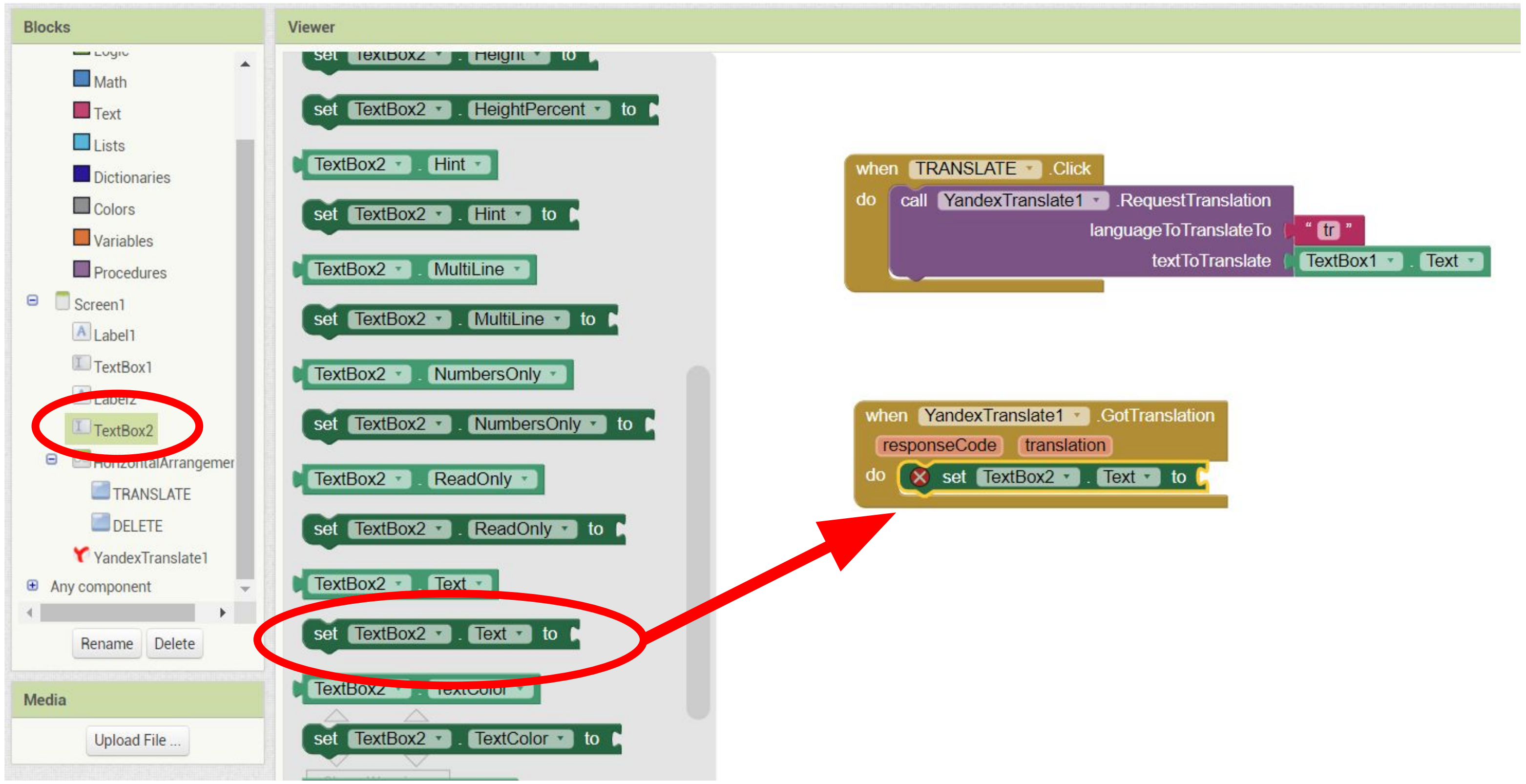
- **Let's drag the Translate Text code from the translate buton and drop it end of yandex translate code.**



Application 1: TRANSLATOR

1-)PROGRAMMING TRANSLATOR:

- **After the YandexTranslate tool finishes the translation process, the result is thrown into the TextBox translation. Thus, this text is displayed in the TextBox.**
- **Let's drag the GetTranslation code from the YandexTranslate1 and drop it on the code screen.**



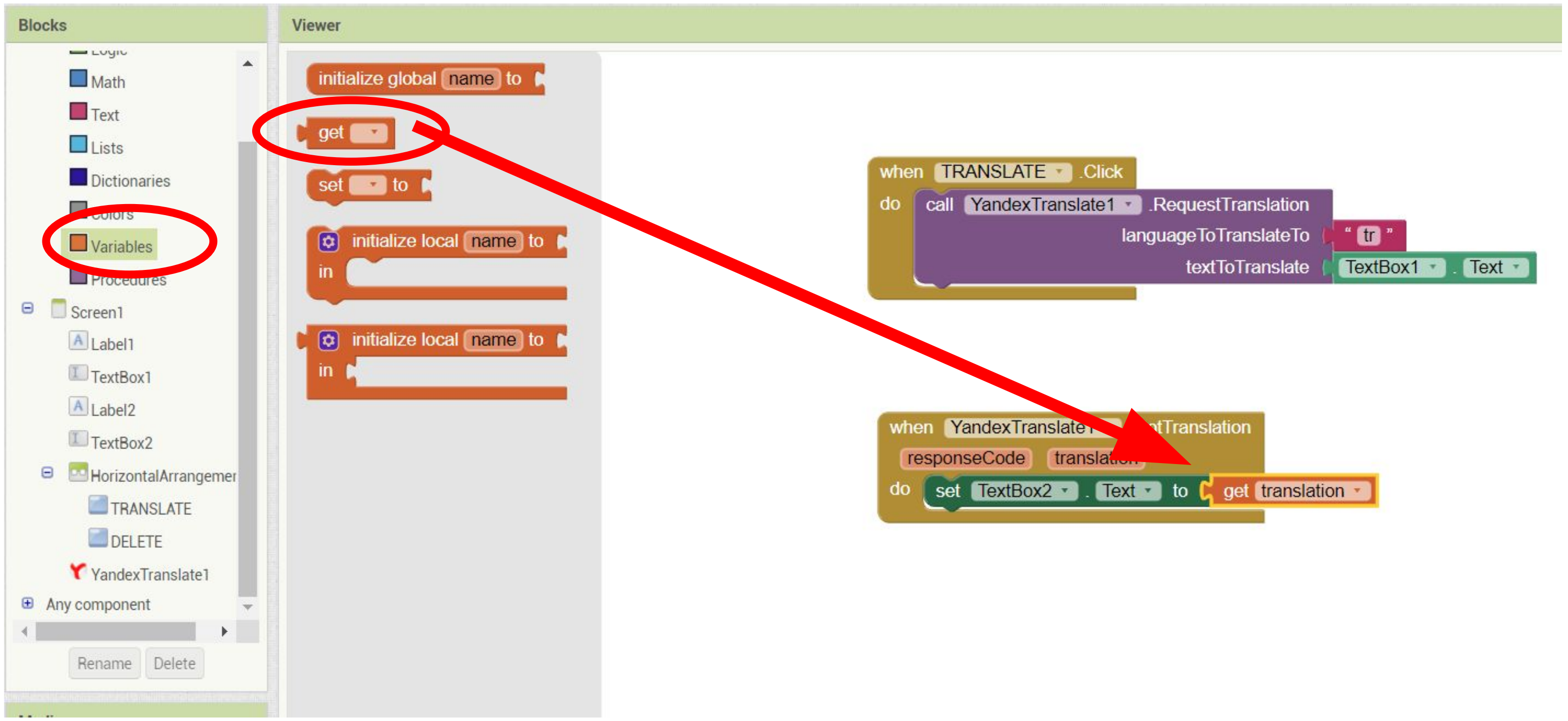
```
when TRANSLATE .Click
do  call YandexTranslate1 .RequestTranslation
              languageToTranslateTo  " tr "
                    textToTranslate  TextBox1 . Text
```

```
when YandexTranslate1 .GotTranslation
 responseCode  translation
do  set TextBox2 . Text to
```

Application 1: TRANSLATOR

1-)PROGRAMMING TRANSLATOR:

- **After the YandexTranslate tool finishes the translation process, the result is thrown into the TextBox translation. Thus, this text is displayed in the TextBox.**
- **Let's drag the GetTranslation code from the YandexTranslate1 and drop it on the code screen.**

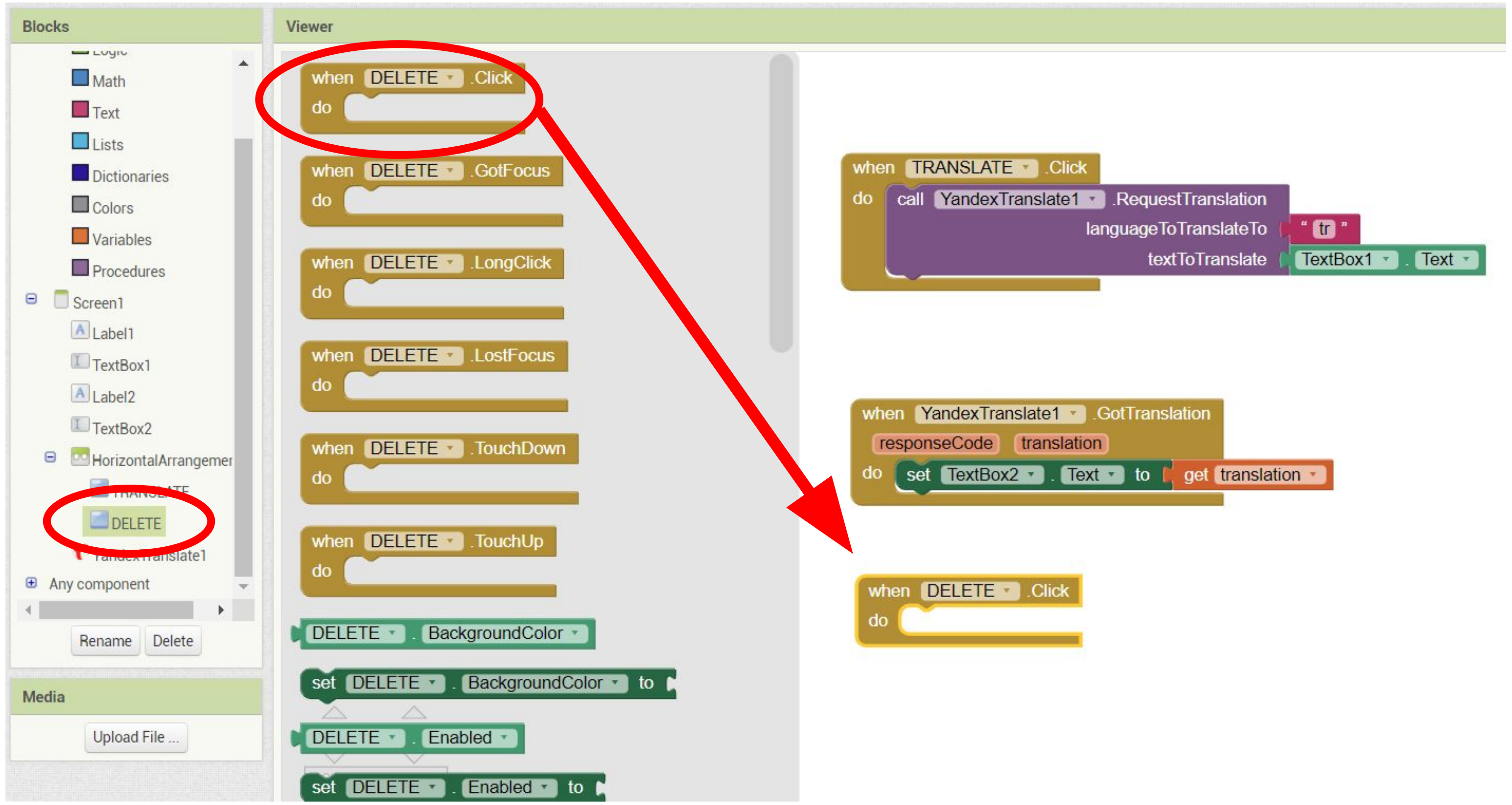


```
initialize global name to
get
set to
initialize local name to
in
initialize local name to
in

when TRANSLATE .Click
do call YandexTranslate1 .RequestTranslation
        languageToTranslateTo "tr"
        textToTranslate TextBox1 . Text

when YandexTranslate1 .GotTranslation
  responseCode  translation
do set TextBox2 . Text to get translation
```

Application 1: TRANSLATOR

1-)PROGRAMMING TRANSLATOR:

- Let's drag the "get" code from the variables and drop it end of the text code.



```
when DELETE .Click
do

when DELETE .GotFocus
do

when DELETE .LongClick
do

when DELETE .LostFocus
do

when DELETE .TouchDown
do

when DELETE .TouchUp
do

DELETE . BackgroundColor
set DELETE . BackgroundColor to
DELETE . Enabled
set DELETE . Enabled to
```

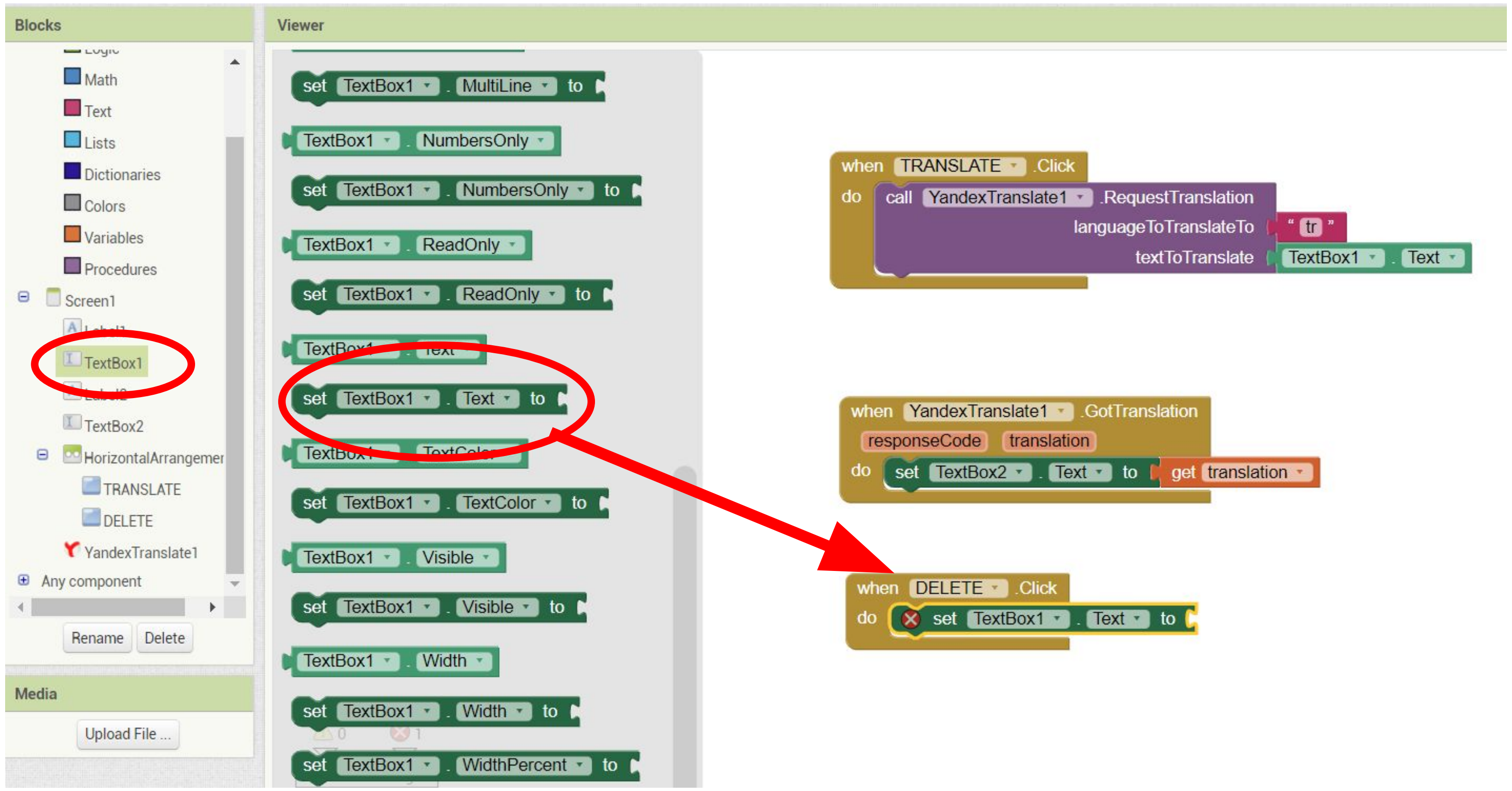
```
when TRANSLATE .Click
do  call YandexTranslate1 .RequestTranslation
                 languageToTranslateTo  " tr "
                 textToTranslate  TextBox1 . Text

when YandexTranslate1 .GotTranslation
  responseCode  translation
do  set TextBox2 . Text to  get translation

when DELETE .Click
do
```

Application 1: TRANSLATOR

1-)PROGRAMMING TRANSLATOR:

- **When you click Delete Button, the text to be translated in the TextBox and the text in the Translation of the TextBox are emptied and cleared.**

AI WITH APP INVENTOR



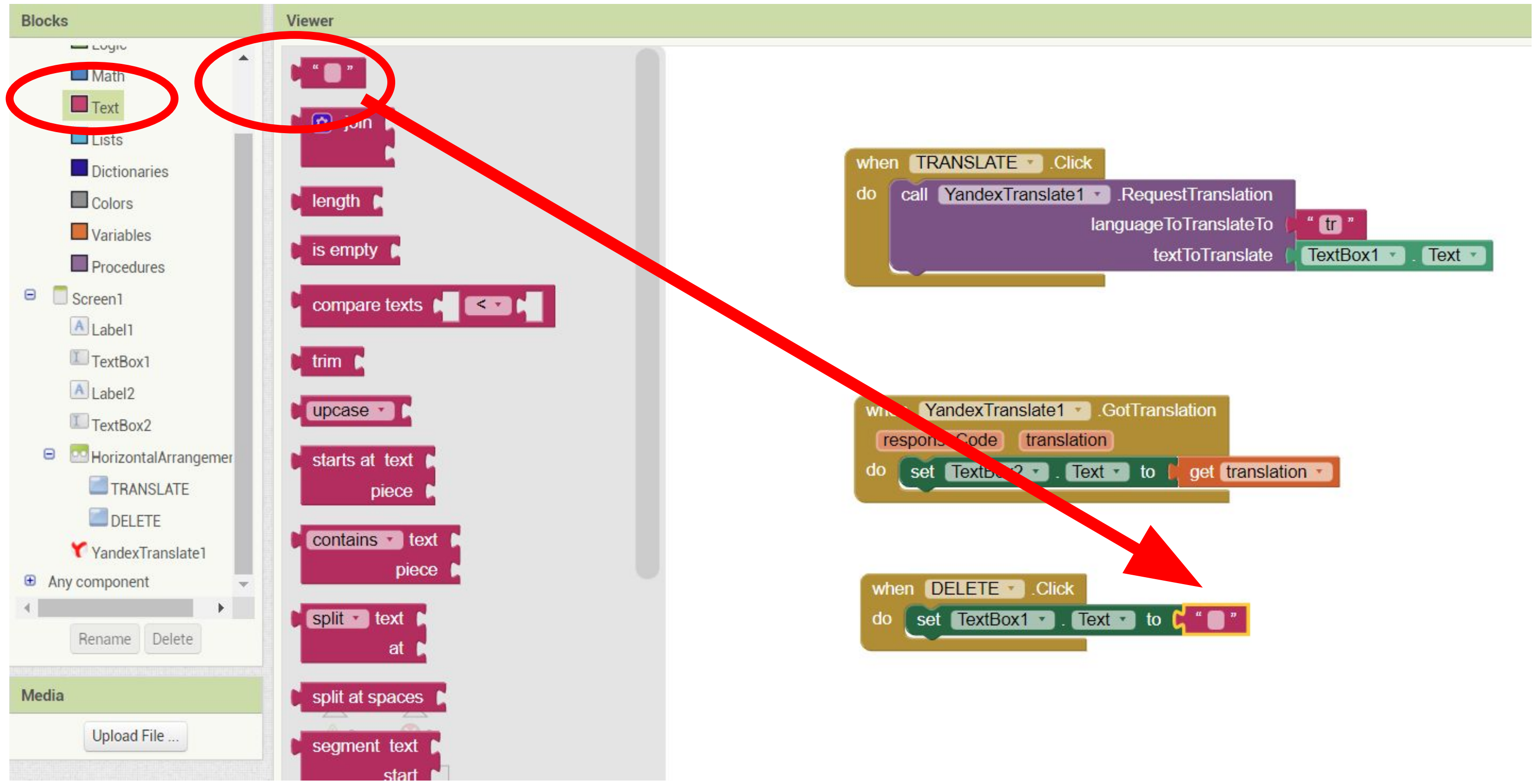
```
set TextBox1 . MultiLine to
TextBox1 . NumbersOnly
set TextBox1 . NumbersOnly to
TextBox1 . ReadOnly
set TextBox1 . ReadOnly to
TextBox1 . Text
set TextBox1 . Text to
TextBox1 . TextColor
set TextBox1 . TextColor to
TextBox1 . Visible
set TextBox1 . Visible to
TextBox1 . Width
set TextBox1 . Width to
set TextBox1 . WidthPercent to
```

```
when TRANSLATE .Click
do  call YandexTranslate1 .RequestTranslation
                languageToTranslateTo  " tr "
                textToTranslate  TextBox1 . Text

when YandexTranslate1 .GotTranslation
  responseCode  translation
do  set TextBox2 . Text to  get translation

when DELETE .Click
do  set TextBox1 . Text to
```

Application 1: TRANSLATOR

1-)PROGRAMMING TRANSLATOR:

- **When you click Delete Button, the text to be translated in the TextBox and the text in the Translation of the TextBox are emptied and cleared.**



```
when TRANSLATE .Click
do  call YandexTranslate1 .RequestTranslation
        languageToTranslateTo  " tr "
        textToTranslate  TextBox1 . Text

when YandexTranslate1 .GotTranslation
  responseCode  translation
do  set TextBox2 . Text to  get translation

when DELETE .Click
do  set TextBox1 . Text to  " "
```

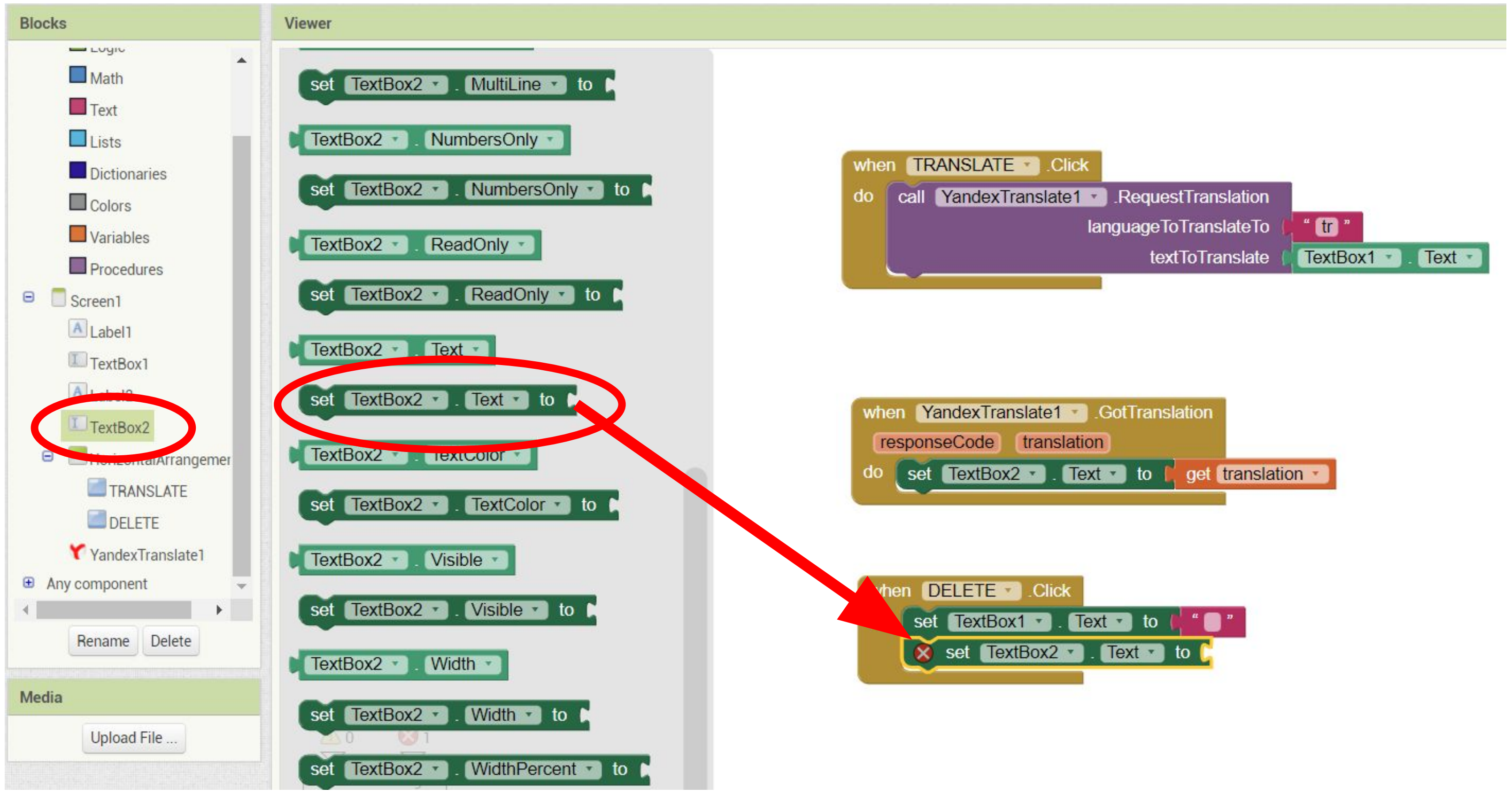
Application 1: TRANSLATOR

1-)PROGRAMMING TRANSLATOR:

- **When you click Delete Button, the text to be translated in the TextBox and the text in the Translation of the TextBox are emptied and cleared.**

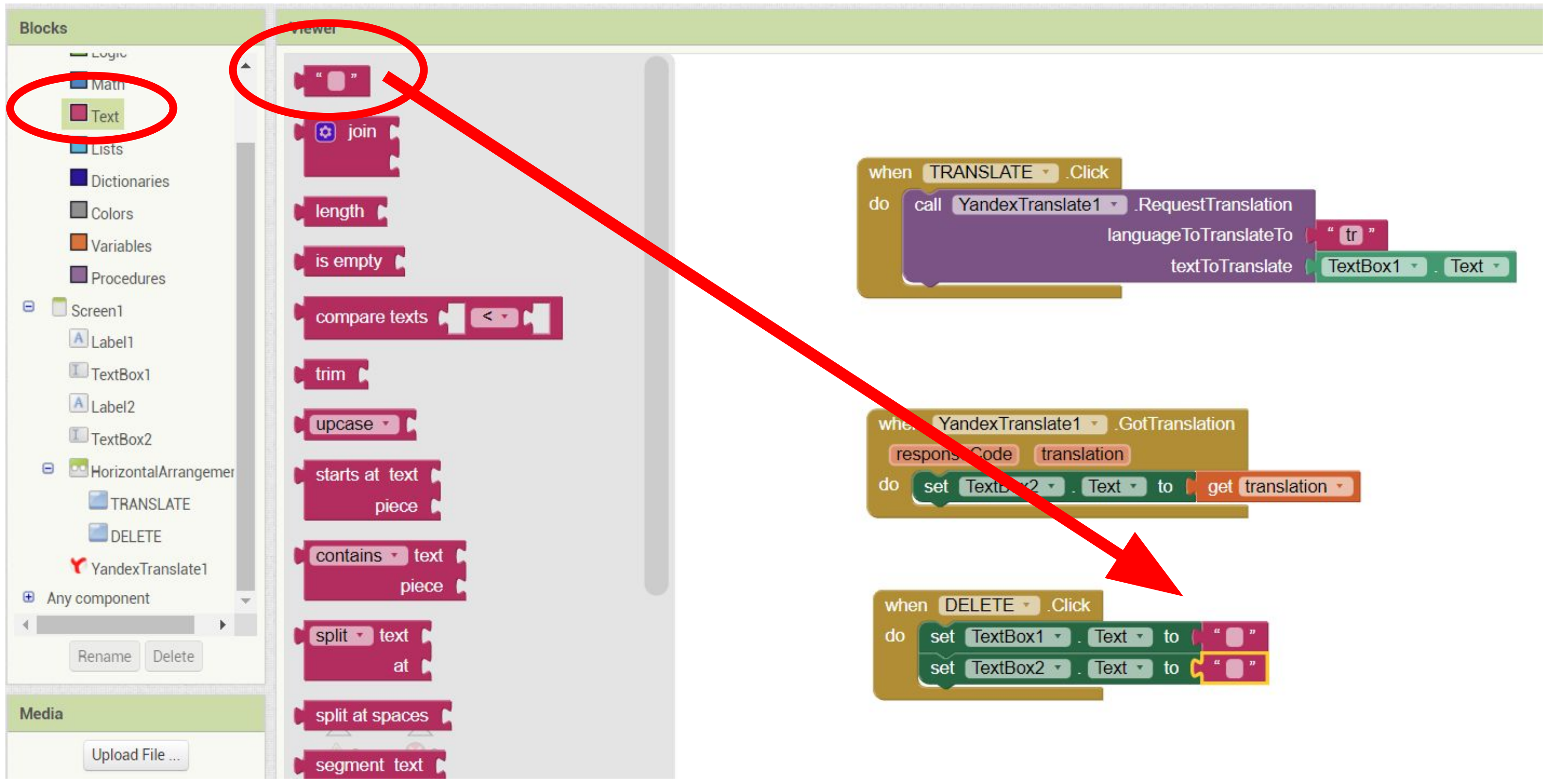
Application 1: TRANSLATOR

1-)PROGRAMMING TRANSLATOR:



```
when TRANSLATE .Click
do  call YandexTranslate1 .RequestTranslation
              languageToTranslateTo  " tr "
              textToTranslate  TextBox1 . Text

when YandexTranslate1 .GotTranslation
 responseCode  translation
do  set TextBox2 . Text to  get translation

when DELETE .Click
    set TextBox1 . Text to  " "
    set TextBox2 . Text to
```

- When you click Delete Button, the text to be translated in the TextBox and the text in the Translation of the TextBox are emptied and cleared.

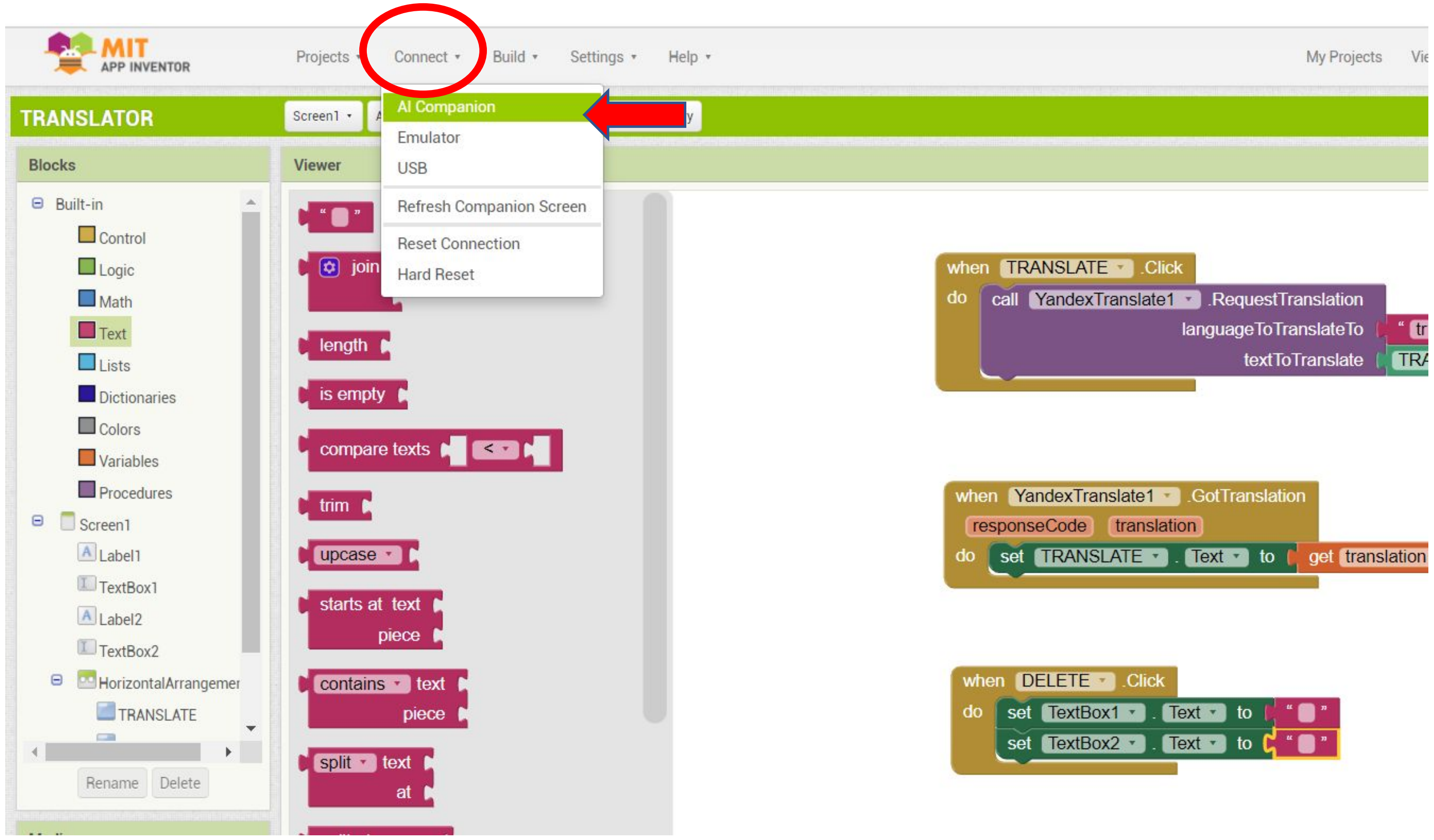


```
when TRANSLATE .Click
do  call YandexTranslate1 .RequestTranslation
                languageToTranslateTo  " tr "
                textToTranslate        TextBox1 . Text

when YandexTranslate1 .GotTranslation
  responseCode  translation
do  set TextBox2 . Text to  get translation

when DELETE .Click
do  set TextBox1 . Text to  " "
    set TextBox2 . Text to  " "
```

Application 1: TRANSLATOR

1-)PROGRAMMING TRANSLATOR:

- When you click Delete Button, the text to be translated in the TextBox and the text in the Translation of the TextBox are emptied and cleared.



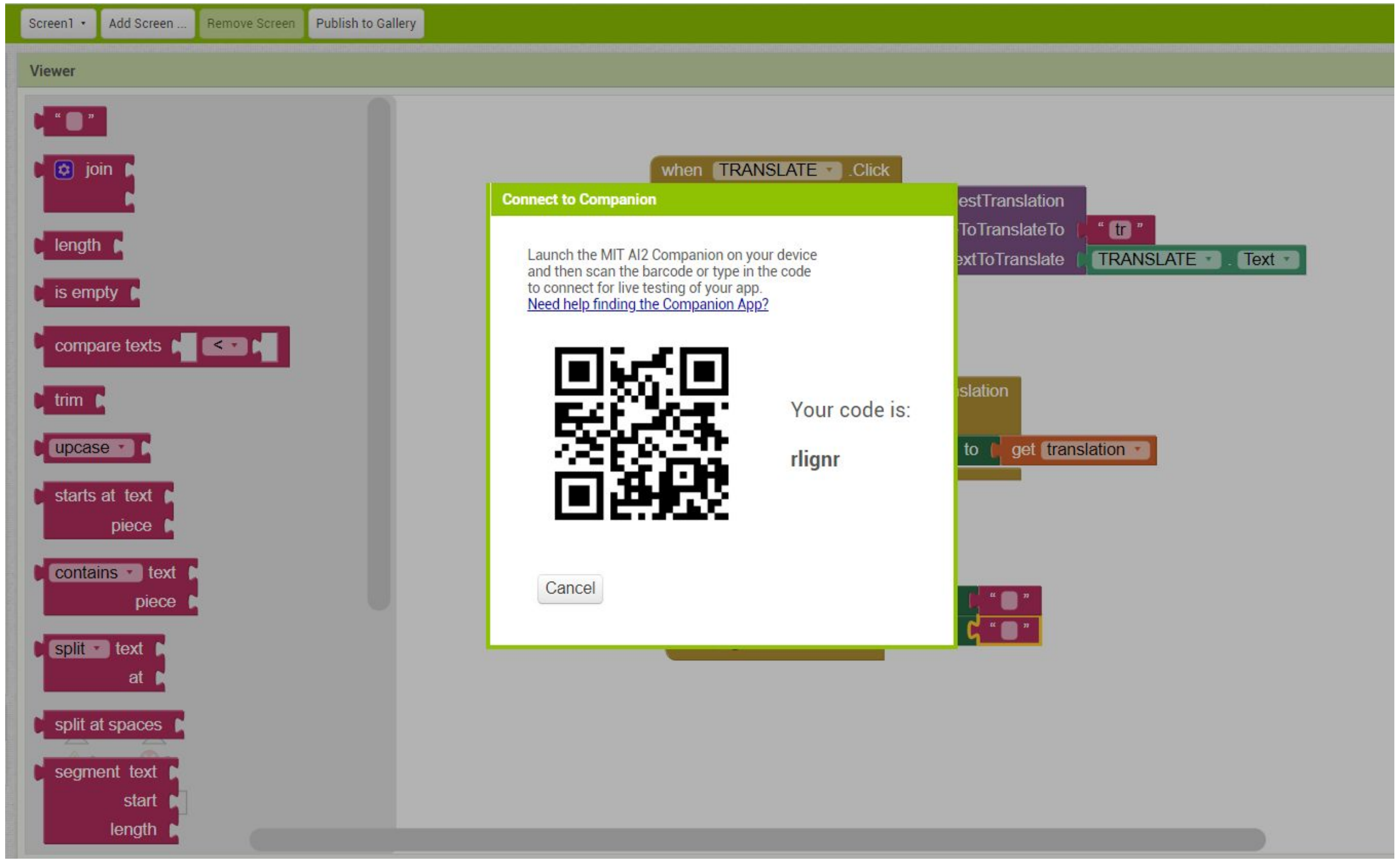
```
when TRANSLATE .Click
do  call YandexTranslate1 .RequestTranslation
                 languageToTranslateTo
                 textToTranslate

when YandexTranslate1 .GotTranslation
 responseCode  translation
do  set TRANSLATE . Text to  get translation

when DELETE .Click
do  set TextBox1 . Text to " "
    set TextBox2 . Text to " "
```

Application 1: TRANSLATOR

1-)LOADING TRANSLATOR:

Let's choose the connect menu from the top menu of the screen.

Then let's select the AI companion menu.



Connect to Companion

Launch the MIT AI2 Companion on your device and then scan the barcode or type in the code to connect for live testing of your app.
Need help finding the Companion App?

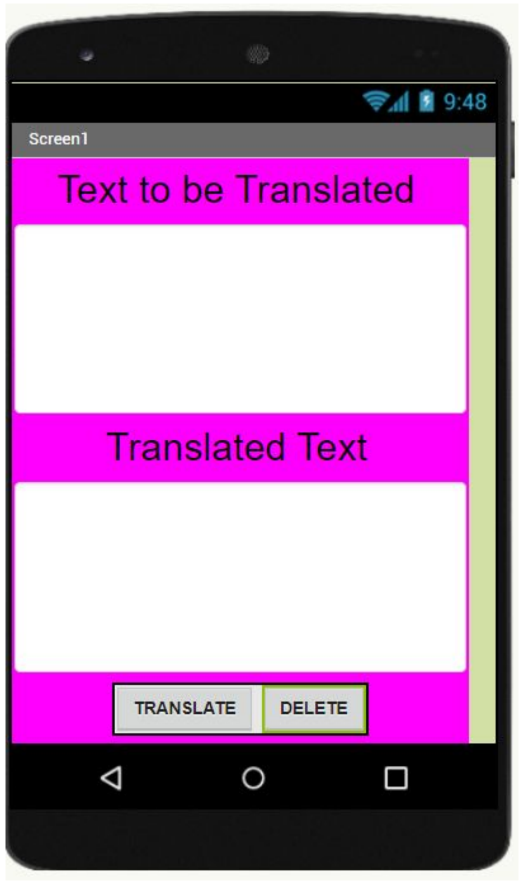
Your code is:

rlignr

Application 1: TRANSLATOR

1-)LOADING TRANSLATOR:

Let's scan the QR code that appears on the screen to the application we have downloaded on our phone.



Application 1: TRANSLATOR

1-)LOADING TRANSLATOR:

Translator is ready…

ROBOTIKBILIM

AI WITH APP INVENTOR

JUNIOR MAKER LAND

THANK YOU