



Mobile application development

移动应用开发



Lecture 09:

APP Inventor _Example

Review the example and introduce some blocks



Dr Ata Jahangir Moshayedi

Prof Associate ,
School of information engineering Jiangxi
university of science and technology, China



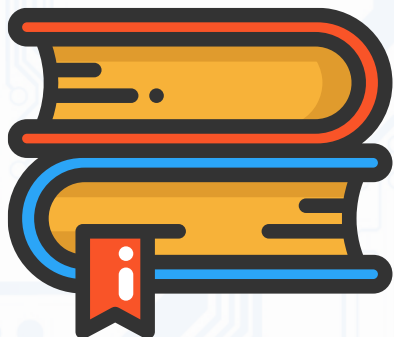
EMAIL: ajm@jxust.edu.cn

Autumn _2021



江西理工大学信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



MOBILE APPLICATION DEVELOPMENT

LECTURE 09: APP Inventor _Example

Review the example and introduce some blocks

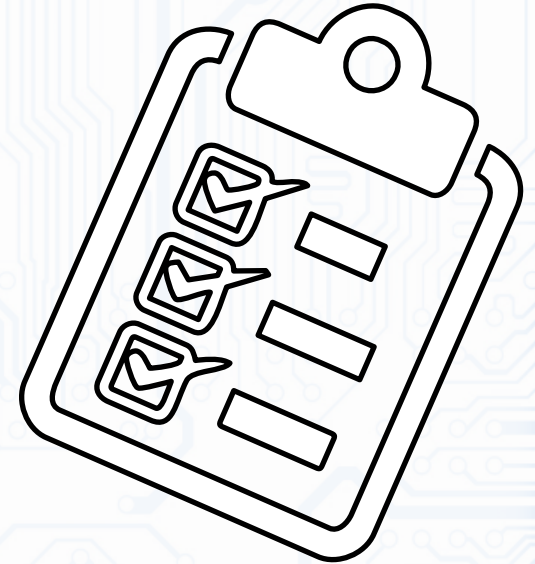




Agenda

– Review the example and introduce some blocks

- MIT App Inventor Text Blocks
- MIT App Inventor Notifier
- MIT App Inventor Variable Blocks
- MIT App Inventor Logic Blocks



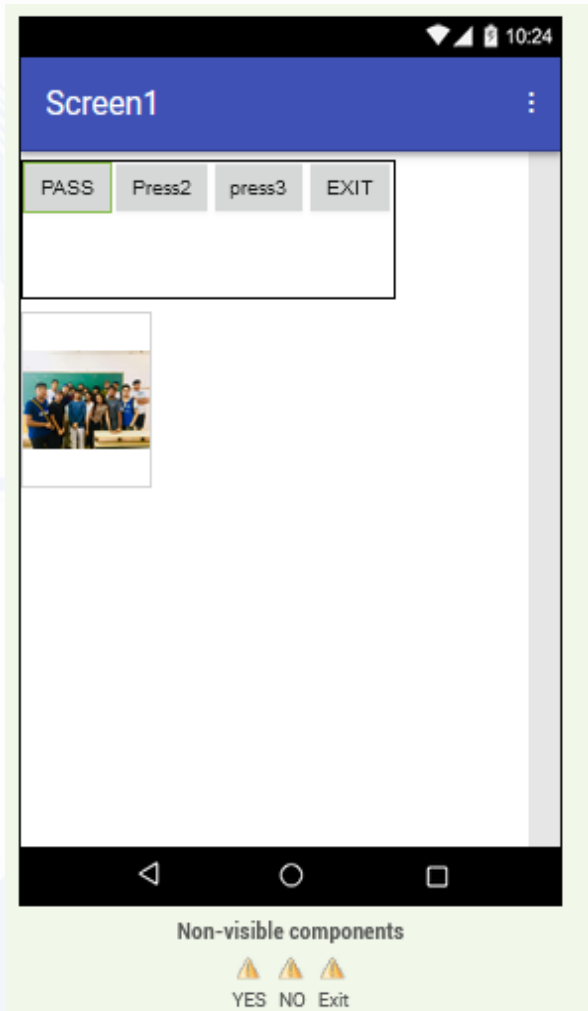


- We will find each needed block in this example and describe blocks and part to design an app
- Let us see our example

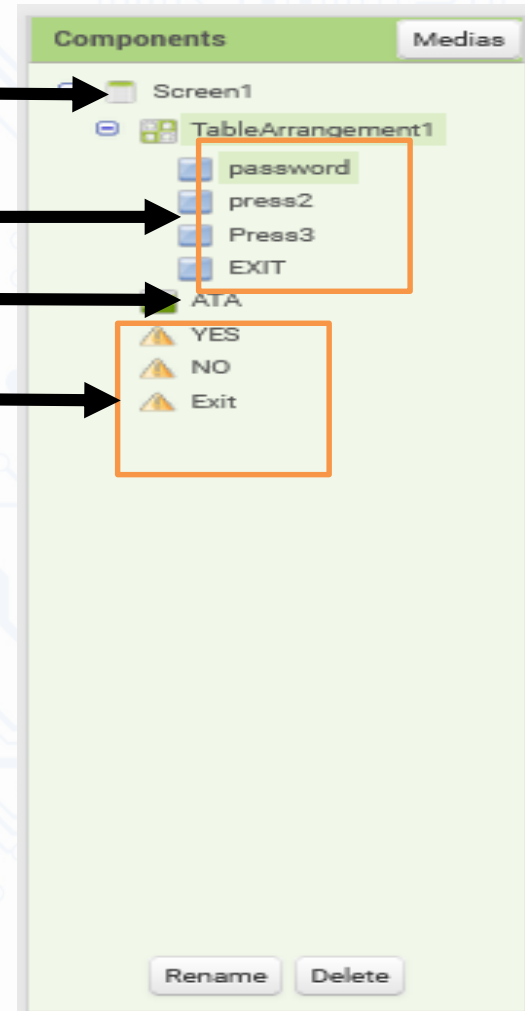




Project part in Designer section



- @Screen 1
- Layout arrangement
- Button *4
- Image
- Notification section





The example

Screenshot of the App Inventor 2 web interface running in a browser (www.BANDICAM.com). The browser address bar shows <https://app.wxbit.com/?locale=en#657897>. The App Inventor 2 interface includes a Palette on the left with various UI components like Button, Switch, Label, Image, AnimationImage, TextBox, PasswordTextBox, RadioButton, CheckBox, Spinner, HorizontalSlider, VerticalSlider, Notifier, LayoutDialog, ListPicker, ListView, File Picker, and Color Picker. The central Viewer shows a mobile app preview for 'Screen1' with buttons labeled 'PASS', 'Press2', 'press3', and 'EXIT'. Below the preview, there are 'Non-visible components' including 'ATA', 'YES', 'NO', and 'Exit'. The right side features a Components list (Screen1, TableArrangement1, password, press2, Press3, EXIT, ATA, YES, NO, Exit) and a Properties panel for the selected 'password' component, showing settings like Visible, Enabled, Clickable, TextColor, TextAlignment, Text, FontSize, Width, Height, and BackgroundColor. The bottom of the screenshot shows a Windows taskbar with various application icons and a system clock indicating 16:01 on 04/11/2020.





App Designs Steps.....



App Inventor

Screenshot of the App Inventor 2 web interface. The browser address bar shows <https://app.wxbit.com/?locale=en#657897>. The interface includes a left sidebar with a "Blocks" palette and a "Designer" view. The "Blocks" palette lists categories like Built-in, Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, Procedures, and Screen1. The "Designer" view shows a screen named "ATA_04" with a table arrangement containing buttons for "password", "press2", "press3", "EXIT", and "ATA". The "Logic" blocks area displays several event-driven scripts:

- Initialize global name to:** A block to initialize a global variable.
- when EXIT.Click:** A script that triggers the "close screen" block.
- when password.Click:** A script that triggers the "call YES.ShowPasswordDialog" block, which sets the message to "Please enter...", the title to "Password", and the cancelable property to true.
- when YES.AfterTextInput:** A script that triggers the "response" block, which then sets the global name to the response. It includes an "if" statement: if the global name is "ATA", it triggers "call YES.ShowPasswordDialog" with message "YES", title "OK", and cancelable true, followed by "open another screen" with screen name "Pg2". Otherwise, it triggers "call NO.ShowPasswordDialog" with message "No", title "TRY AGIN!!!", and cancelable true.

The bottom of the interface shows a Windows taskbar with various application icons and a system clock indicating 16:01 on 04/11/2020.



江西理工大学信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



Project part in Block section

App Inventor 2
WxBit 汉化版

Projects Connect Build Help

My Projects Gallery Supports English Ata Jahangir

ATA_04 Screen1 Imp / Exp Add Screen Copy Screen Remove Screen Designer Blocks

Blocks

- Built-in
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Dictionaries
 - Colors
 - Variables
 - Procedures
- Screen1
 - TableArrangement1
 - password
 - press2
 - Press3
 - EXIT
 - ATA
 - YES
 - NO
 - Exit
- Any component
 - Any AndroidViewCom
 - Any Button
 - Any Image
 - Any Notification

Viewer

1 when EXIT.Click do close screen

2 when password.Click do call YES.ShowPasswordDialog message Please enter ... title Password cancelable true

3 initialize global name to ""

when YES.AfterTextInput response do set global name to get response if ATA == get global name then call YES.ShowPasswordDialog message YES title OK cancelable true open another screen screenName Pg2 else call NO.ShowPasswordDialog message No title TRY AGIN!!! cancelable true

Hide Warnings

Theme Color Dark Mode

Copyright © www.wxbit.com | Privacy Policy and Terms of Use





MIT App Inventor Text Blocks



- string
- join
- length
- is empty
- compare texts
- trim
- upcase
- downcase
- starts at
- contains
- contains any
- contains all
- split at first
- split at first of any
- split
- split at any
- split at spaces
- segment
- replace all
- obfuscated text
- is a string?
- reverse
- replace all mappings



Built-in

-  Control
-  Logic
-  Math
-  Text
-  Lists
-  Dictionaries
-  Colors
-  Variables
-  Procedures





MIT App Inventor Text Blocks



” “ (string block)



Contains a text string.

This string can contain any characters (letters, numbers, or other special characters).

On App Inventor it will be considered a Text object.

length



- Returns the number of characters including spaces in the string.
- This is the length of the given text string.

join



- Appends all of the inputs to make a single string.
- If no inputs, returns an empty string

is empty



- Returns whether or not the string contains any characters (including spaces).
- When the string length is 0, returns true otherwise it returns false.





MIT App Inventor Text Blocks



compare texts < > = ≠



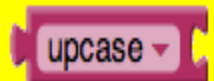
- Returns whether or not the first string is lexicographically <, >, =, or ≠ the second string depending on which dropdown is selected.
- A string is considered lexicographically greater than another if it is alphabetically greater than the other string.
- Essentially, it would come after it in the dictionary. All uppercase letters are considered smaller or to occur before lowercase letters. cat would be > Cat.

trim



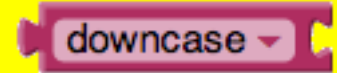
- Removes any spaces leading or trailing the input string and returns the result.

upcase



- Returns a copy of its text string argument converted to all upper case.

downcase



- Returns a copy of its text string argument converted to all lower case.

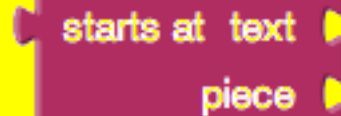




MIT App Inventor Text Blocks

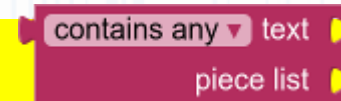


starts at



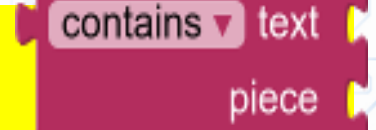
Returns the character position where the first character of *piece* first appears in text, or 0 if not present. For example, the location of *ana* in *havana banana* is 4.

contains any



- Returns true if any of the pieces in the piece list appear in text; otherwise, returns false.
- This block can be obtained by changing the dropdown on the contains block

contains



Returns true if piece appears in text; otherwise, returns false.

contains all



- Returns true if all of the pieces in the piece list appear in text; otherwise, returns false.
- This block can be obtained by changing the dropdown on the contains block.

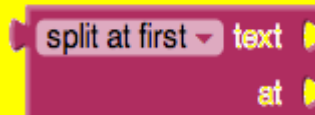




MIT App Inventor Text Blocks

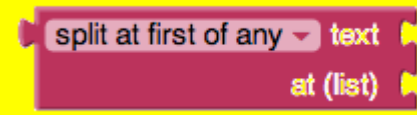


split at first



Divides the given text into two pieces using the location of the first occurrence of at as the dividing point, and returns a two-item list consisting of the piece before the dividing point and the piece after the dividing point. Splitting apple,banana,cherry,dogfood with a comma as the splitting point returns a list of two items: the first is the text apple and the second is the text banana,cherry,dogfood. Notice that the comma after apple doesn't appear in the result, because that is the dividing point.

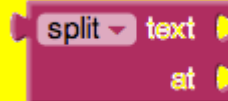
split at first of any



Divides the given text into a two-item list, using the first location of any item in the list at as the dividing point.

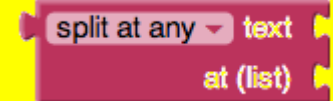
Splitting i love apples bananas apples grapes by the list [ba,ap] would result in a list of two items the first being i love and the second ples bananas apples grapes.

split



- Divides text into pieces using at as the dividing points and produces a list of the results.
- Splitting one,two,three,four at , (comma) returns the list (one two three four). Splitting one-potato,two-potato,three-potato,four at -potato, returns the list (one two three four).

split at any



- Divides the given text into a list, using any of the items in at as the dividing point, and returns a list of the results.
- Splitting appleberry,banana,cherry,dogfood with at as the two-element list whose first item is a comma and whose second item is rry returns a list of four items: (applebe banana che dogfood).





MIT App Inventor Text Blocks



split at spaces



Divides the given text at any occurrence of a space, producing a list of the pieces.

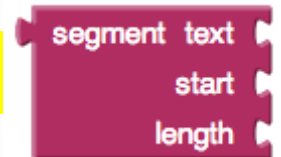
replace all



Returns a new text string obtained by replacing all occurrences of the substring with the replacement.

Replace all with She loves eating. She loves writing. She loves coding as the text, She as the segment, and Hannah as the replacement would result in Hannah loves eating. Hannah loves writing. Hannah loves coding.

segment



Extracts part of the text starting at start position and continuing for length characters.

obfuscated text



Produces text, like a text block. The difference is that the text is not easily discoverable by examining the app's contents.

Use this when creating apps to distribute that include confidential information, for example, API keys.

Warning: This provides only very low security against expert adversaries.





MIT App Inventor Text Blocks



is a string?

is a string? thing

Returns true if *thing* is a text object, otherwise false.

reverse

reverse

- Reverse the given text. For example, “reverse” would become “esrever”.

replace all mappings

replace all mappings
in text
preferring longest string first order

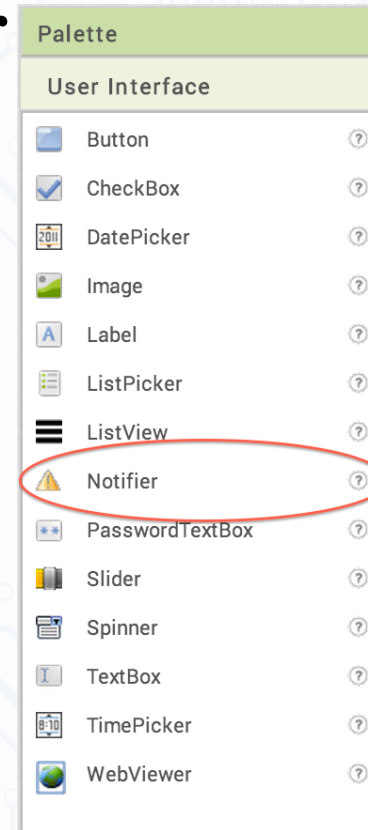
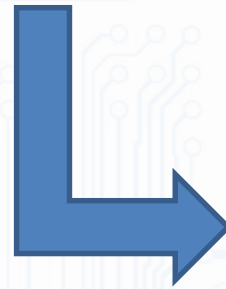
Given a dictionary of mappings as input, replaces the key entries in the text with the corresponding values in the dictionary. Returns the text with the mappings applied.



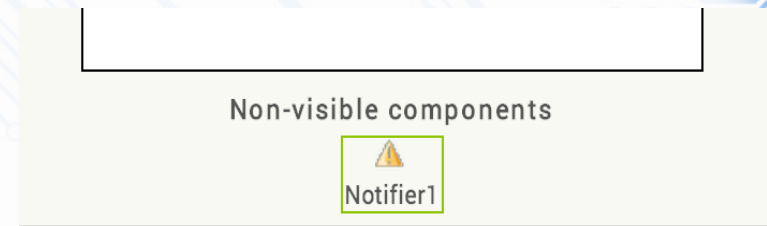


MIT App Inventor Notifier

To create a pop up warning dialog, use the *Notifier* control that appears in the Designer's Palette:



Drag the Notifier icon into the app user interface design – the Notifier is used as a non-visible component, just below the user interface:

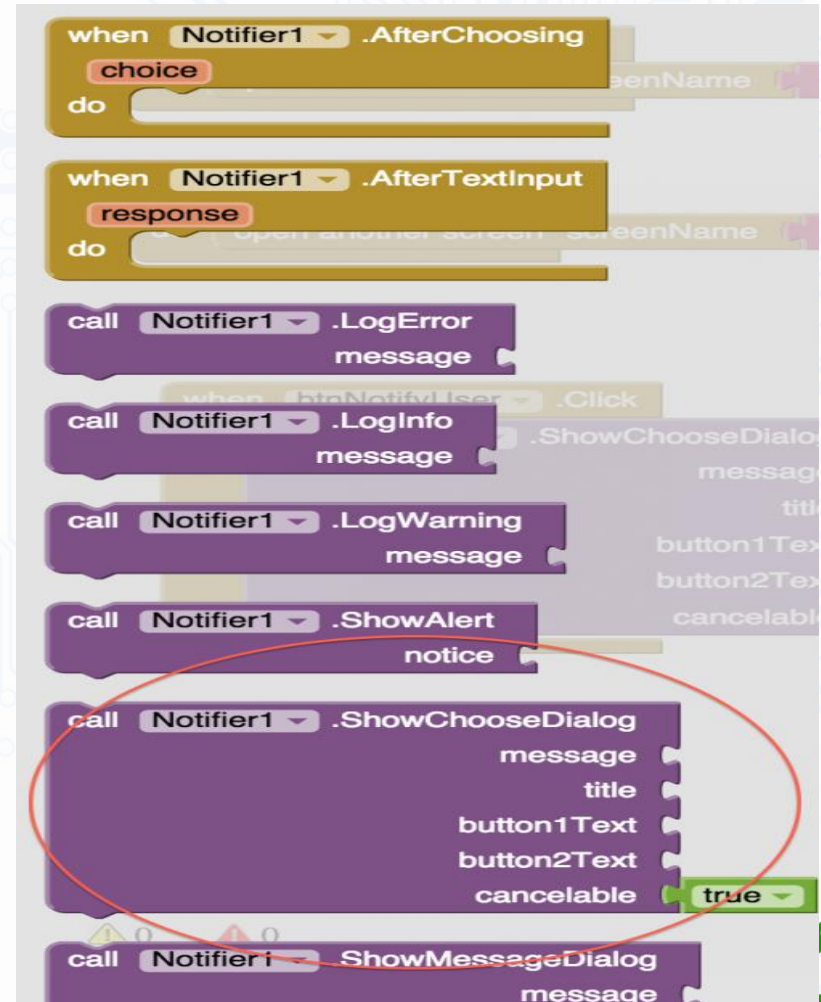




MIT App Inventor Notifier



- Switch to the Blocks view, click on the Notifier1 block to view the available methods.
- partial list of methods is shown here – the item circled in red is the one we will use in this example:





MIT App Inventor Notifier



In response to an on screen button press (*btnNotifyUser's Click* event), the app displays the dialog message:

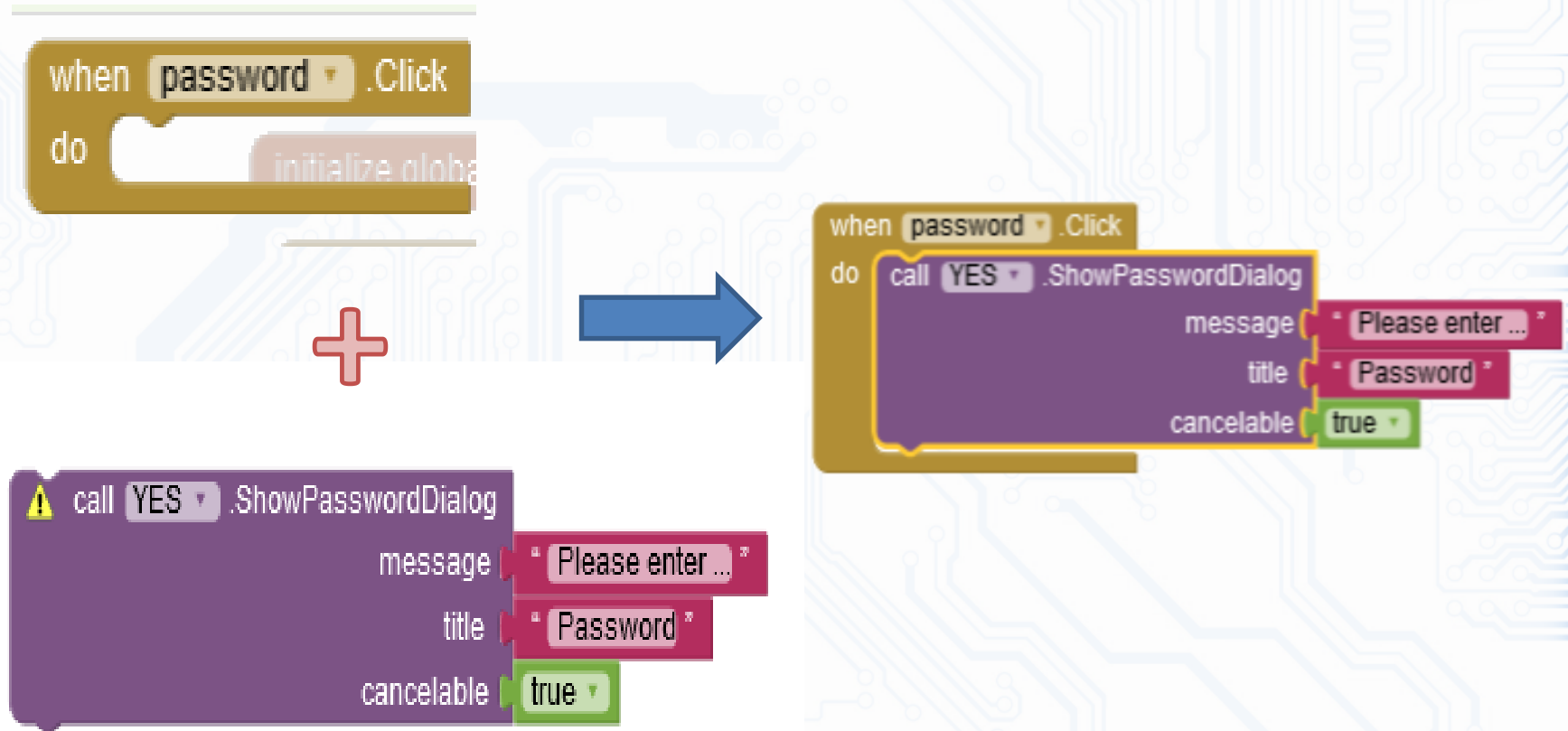


- This block can be inserted anywhere a processing block is allowed such as the result of an if-then-else conditional block.
- You can also change the text or background color of the app to something more interesting than the “black and white” default colors shown in this example.
- ***Notifier*** provides a quick and easy way to display short alert messages to the user of the app.





How to make the notifier





MIT App Inventor Variable Blocks

- There are five main types of variable blocks:
 - initialize global name to
 - get
 - set
 - initialize local name to in (do)
 - initialize local name to in (return)

Built-in

-  Control
-  Logic
-  Math
-  Text
-  Lists
-  Dictionaries
-  Colors
-  Variables
-  Procedures





MIT App Inventor Variable Blocks



initialize global name to

initialize global **name** to

- This block is used to create global variables. It takes in any type of value as an argument.
- Clicking on *name* will change the name of this global variable.
- Global variables are used in all procedures or events so this block will stand alone.
- Global variables can be changed while an app is running and can be referred to and changed from any part of the app even within procedures and event handlers.
- can rename this block at any time and any associated blocks referring to the old name will be updated automatically.





MIT App Inventor Variable Blocks



get



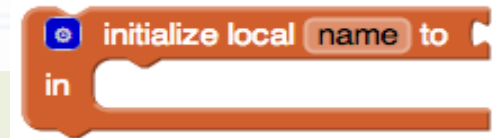
This block provides a way to get any variables you may have created.

set



- This block follows the same rules as get.
- Only variables in scope will be available in the dropdown.
- Once a variable v is selected, you can attach a block to give v a new value.

initialize local name to - in (do)



This block is a mutator that allows you to create new variables that are only used in the procedure you run in the DO part of the block. This way all variables in this procedure will all start with the same value each time the procedure is run.

NOTE: This block differs from the block described below because it is a DO block.

You can attach *statements* to it. Statements *do* things.

That is why this block has space inside for statement blocks to be attached.

You can rename the variables in this block at any time and any corresponding blocks elsewhere in your program that refer to the old name will be updated automatically





MIT App Inventor Variable Blocks



initialize local name to - in (return)



This block is a mutator that allows you to create new variables that are only used in the procedure you run in the RETURN part of the block.

This way all variables in this procedure will all start with the same value each time the procedure is run.

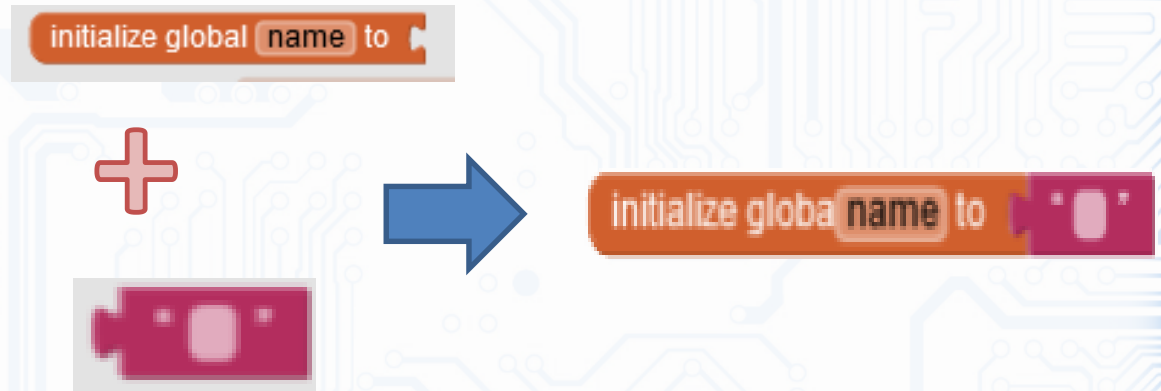
NOTE: This block differs from the block described above because it is a RETURN block.

- You can attach *expressions* to it. Expressions *return* a value. That is why this block has a socket for plugging in expressions.
- You can rename the variables in this block at any time and any corresponding blocks elsewhere in your program that refer to the old name will be updated automatically





Variable section





MIT App Inventor Logic Blocks



- There are Seven main types of variable blocks:

- True
- False
- = {#=}
- Not
- ≠ {#not=}
- And
- or



Built-in

- Control
- Logic
- Math
- Text
- Lists
- Dictionaries
- Colors
- Variables
- Procedures

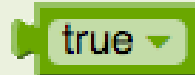




MIT App Inventor Logic Blocks

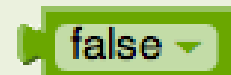


true



- Represents the constant value true. Use it for setting boolean property values of components, or as the value of a variable that represents a condition.

false



- Represents the constant value false. Use it for setting boolean property values of components, or as the value of a variable that represents a condition.

= {#=}



- Tests whether its arguments are equal.**

Two numbers are equal if they are numerically equal, for example, 1 is equal to 1.0.

Two text blocks are equal if they have the same characters in the same order, with the same case. For example, banana is not equal to Banana.

Numbers and text are equal if the number is numerically equal to a number that would be printed with that text. For example, 12.0 is equal to the result of joining the first character of 1A to the last character of Teafor2.

Two lists are equal if they have the same number of elements and the corresponding elements are equal.

Acts exactly the same as the = block found in Math

not



- Performs logical negation, returning false if the input is true, and true if the input is false.





MIT App Inventor Logic Blocks



≠ {#not=}



Tests to see whether two arguments are not equal.

or

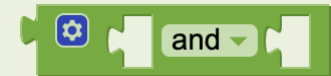


Tests whether any of a set of logical conditions are true. The result is true if one or more of the tested conditions are true.

The number of tests can be expanded using the mutator.

The conditions are tested left to right, and the testing stops as soon as one of the conditions is true. If there are no conditions to test, then the result is false.

and

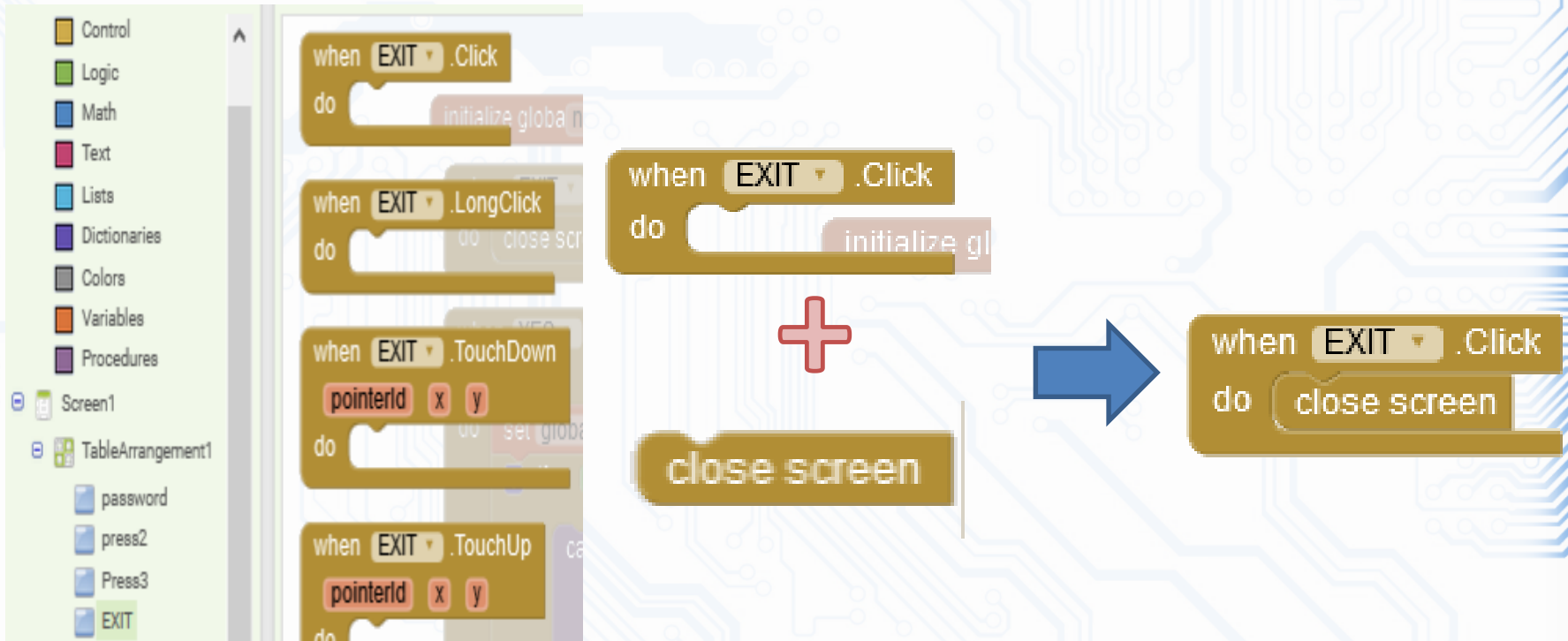


- Tests whether all of a set of logical conditions are true. The result is true if and only if all the tested conditions are true. The number of tests can be expanded using the mutator.
- The conditions are tested left to right, and the testing stops as soon as one of the conditions is false. If there are no conditions to test, then the result is true. You can consider this to be a logician's joke.





Bottom block actions





All needed part for our example



App Inventor

ATA = get global name

call YES .ShowPasswordDialog
message YES
title OK
cancelable true
open another screen screenName Pg2

call NO .ShowPasswordDialog
message No
title TRY AGIN!!!
cancelable true

Procedures

Screen1

- TableArrangement1
 - password
 - press2
 - Press3
 - EXIT
- ATA
- YES
- NO
- Exit

when YES .AfterTextInput
do
response
if
ATA

when YES .TextInputCanceled
do
call YES .ShowNotifier
notice

set global name to get response

if ATA = get global name

then

call YES .ShowPasswordDialog
message YES
title OK
cancelable true
open another screen screenName Pg2

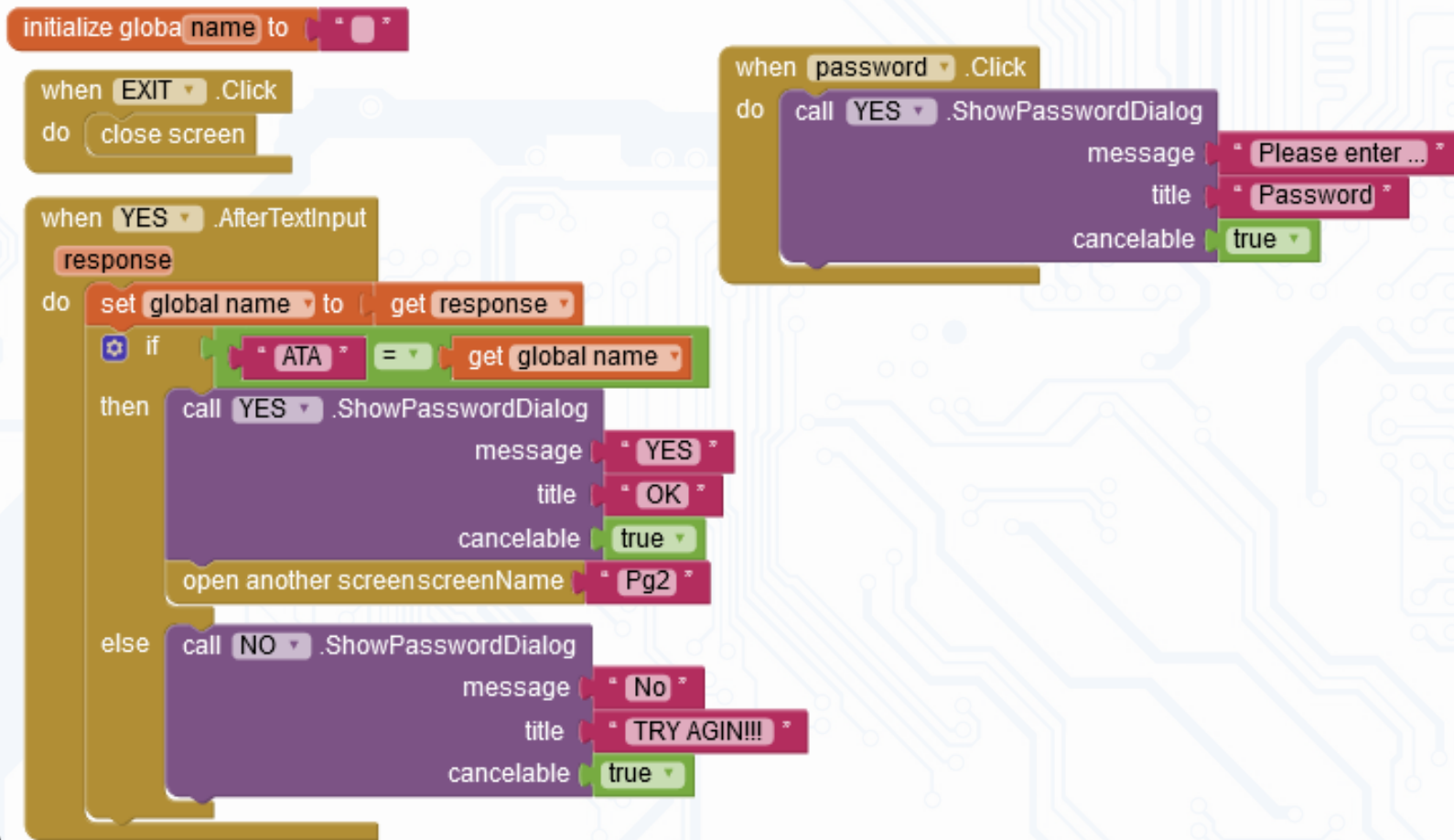
else

call NO .ShowPasswordDialog
message No
title TRY AGIN!!!
cancelable true





All needed part for our example





App Designs Steps



App Inventor

App Inventor 2 interface showing a project named "ATA_04". The interface includes a Palette of components (User Interface, Non-visible components), a Viewer (Screen1) displaying a mobile app preview with buttons (PASS, Press2, press3, EXIT) and an image, and a Properties panel for the selected component (ATA). The Properties panel shows attributes like Visible, Clickable, ScalePictureToFit, Picture (ATA1.jpg...), Shape, Width, Height, and ContentDescription. A "Create new App Inventor project" dialog box is open, prompting for a Project name (New a).

www.BANDICAM.COM

1366x768 - (0, 0), (1366, 768) - Display 1

Record / Stop

Copyright © www.wxbit.com | Privacy Policy and Terms of Use



江西理工大学信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



App Designs Steps



App Inventor 2 interface showing a web browser window with the URL <https://app.wxbit.com/?locale=en#659126>. The interface includes a Palette of components (User Interface, Media, etc.), a Viewer showing a mobile app preview with buttons and an image, and a Properties panel for the selected component (Button1).

The app preview in the Viewer shows a screen titled "Screen1" with a blue header bar. Below the header, there are four buttons labeled "Button1", "Button2", "Button3", and "Button4". Below the buttons is an image of a man and a robot, with the text "JXUST....." below it.

The Properties panel for Button1 shows the following settings:

- Visible: ☒
- Enabled: ☒
- Clickable: ☒
- TextColor: Default
- TextAlignment: center : 1
- Text: Button1
- FontSize: 14.0
- Width: Automatic...
- Height: Automatic...
- BackgroundColor: Default





App Designs Steps



App Inventor

App Inventor 2 interface showing a web browser window displaying the App Inventor website (https://app.wxbit.com/?locale=en#659126). The interface includes a Blocks palette on the left, a Designer view on the right, and a top toolbar with various controls and a Record/Stop button. The Blocks palette lists categories like Built-in, Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, Procedures, and Screen1. The Designer view shows a screen with two buttons: EXIT and PASSWORD. The EXIT button has a Click event with a 'close screen' block. The PASSWORD button has a Click event with a 'ShowMessageDialog' block. The 'ShowMessageDialog' block has a message of 'Please Enter the pass', a title of 'Password', and a button text of 'true'. The top toolbar includes a BANDICAM watermark, a timer (00:00:00), and a Record/Stop button. The bottom status bar shows the Windows taskbar with various application icons and the system clock (18:11, 04/11/2020).



江西理工大学信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



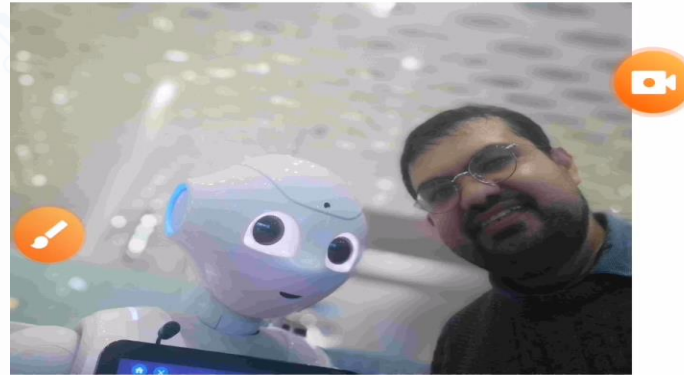
App Inventor



Designed app performance



There is one problem



JXUST.....



江西理工大学信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



App Designs Steps



App Inventor 2

www.BANDICAM.com

https://app.wxbit.com/?locale=en#659126

Getting Started PID Basic functions related... بوش زبان C JRM | Fuji Technology ...

App Inventor 2
WxBit 汉化版

Projects Connect Build Help

My Projects Gallery Supports English Ata Jahangir

New app_01 Screen1 Imp / Exp Add Screen Copy Screen Remove Screen Designer Blocks

Blocks

- Built-in
 - Control
 - Control
 - Logic
 - Math
 - Text
 - Lists
 - Dictionaries
 - Colors
 - Variables
 - Procedures
 - Screen1
 - TableArrangement1
 - PASSWORD
 - Press2
 - Press3
 - EXIT
 - ATA
 - Label1
 - YES
 - NO
 - Exit
 - Any component
 - Any AndroidViewCom
 - Any Button
 - Any Image

Viewer

when EXIT.Click
do close screen

when PASSWORD.Click
do call YES.ShowDialog
message Please Enter the pass
title Password
buttonText true

initialize global name to ""

when YES.AfterTextInput
response
do set global name to get response
if ATA = get response
then call YES.ShowDialog
message YES
title OK
buttonText true
open another screen screenName Screen: Screen2
else call NO.ShowDialog
message NO
title TRY AGAIN
buttonText true

Hide Warnings

Theme Color

Copyright © www.wxbit.com | Privacy Policy and Terms of Use

18:19
04/11/2020



App Designs Steps



App Inventor 2 interface showing a web browser window with the URL <https://app.wxbit.com/?locale=en#659126>. The interface includes a Palette of components, a Viewer window displaying the app design, and a Properties panel. A progress dialog box is overlaid on the Viewer, indicating the progress for building the APK (60%) and the status "Compiling part 2 (please wait)".

The progress dialog box is titled "Progress for building APK" and shows a green progress bar at 60%. Below the bar, it says "Compiling part 2 (please wait)".

The App Inventor interface also shows a list of components in the Palette, including Button, Switch, Label, Image, AnimationImage, TextBox, PasswordTextBox, RadioButton, CheckBox, Spinner, HorizontalSlider, VerticalSlider, Notifier, LayoutDialog, ListPicker, ListView, File Picker, and Color Picker. The Properties panel on the right shows settings for the selected component, including PackageName, AppName, Icon, Title, AboutScreen, AlignHorizontal, AlignVertical, Sizing, Scrollable, KeepScreenOn, and ScreenOrientation.

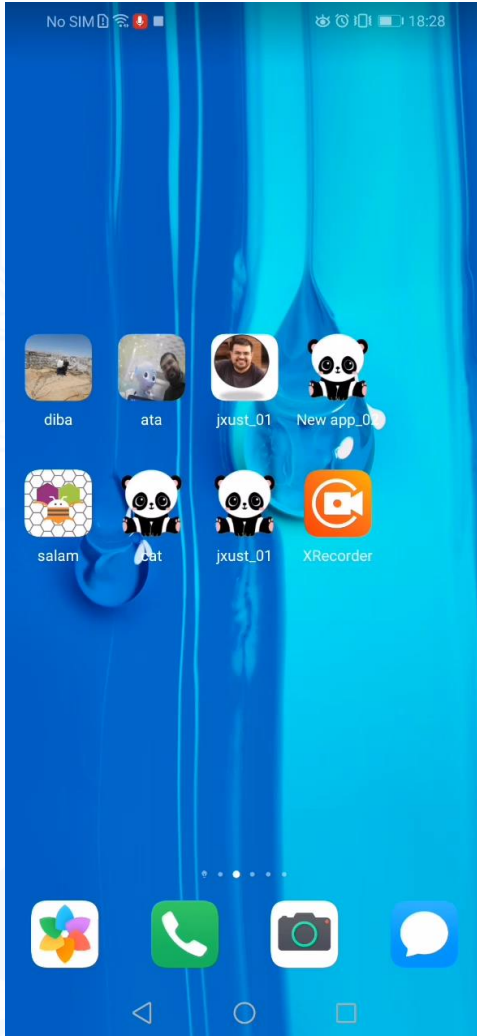




Final App



App Inventor



**How we can make
this design better ?**

you should write the
pass in caps how we can
make the case
independent of caps or
small?



江西理工大学信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



App Inventor



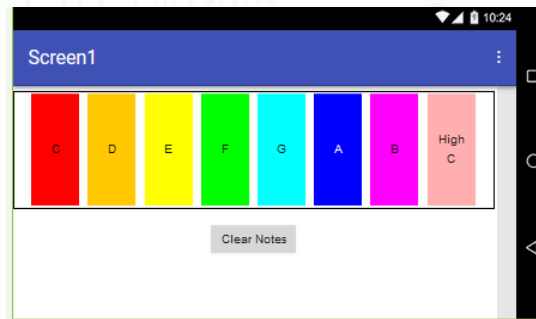
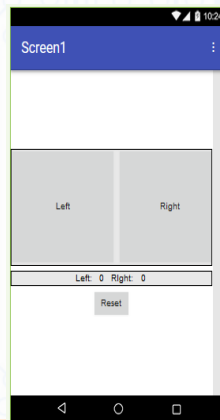
Student Task_7



1. Repaet the example in the ppt and report and sort the problem

Whats your prefer name for this app?

2. Design and follow two example shown on task book 1, you should send the design process and working add clips inside your ppt



Next lecture

- Just MOOC
- Your file should have this format of name
<Task number><student name><Student ID>.ppt





Reference

- <https://coldstreams.com/appinventor/2014/09/08/display-warning-and-alert-box-messages-in-app-inventor-apps/>
- <https://appinventor.mit.edu/explore/content/alertme.html>
- **Teaching with AppInventor** <http://appinventor.mit.edu/explore/teach.html>
- **AppInventor Tutorials:** <http://appinventor.mit.edu/explore/ai2/tutorials.html>
- **Sounds** <http://www.soundbible.com>
- **App Inventor:** <http://appinventor.googlelabs.com/>
- **Appinventor.org:** <http://www.appinventor.org/>
- **Wolber, Abelson et al. text:** <http://www.appinventor.org/text2011>
- **Group:** <http://groups.google.com/group/app-inventor-instructors>
- **Wolber course:** <http://appinventor.org/course-in-a-box>
- **Morelli course:** <http://turing.cs.trincoll.edu/~ram/cpsc110/>



“We are one
society. We are
one globe.”

STEVEN CHU
Nobel Prize in Physics 1997



江西理工大学

Jiangxi University of Science and Technology

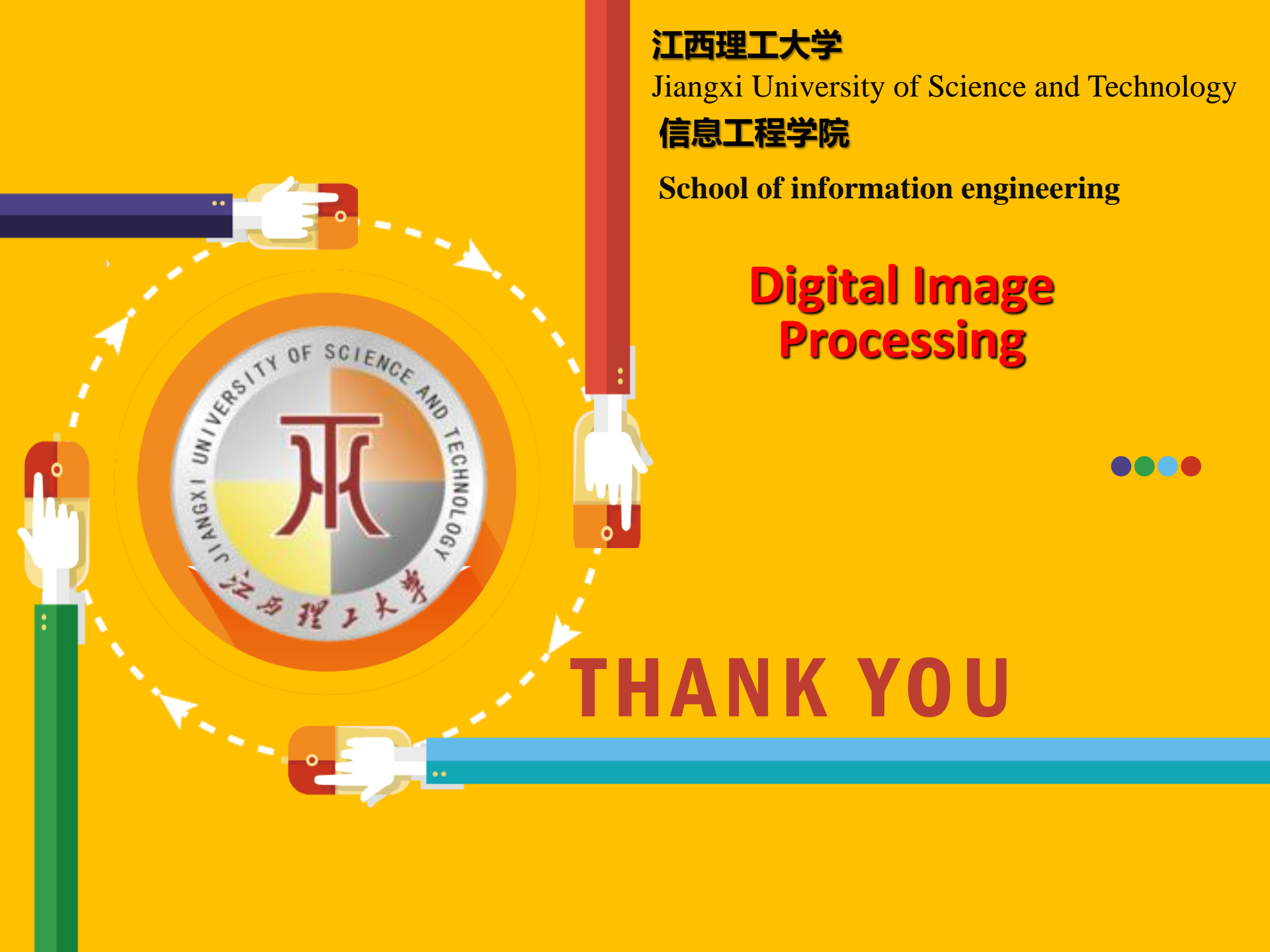
信息工程学院

School of information engineering

Digital Image Processing



THANK YOU





**“BE HUMBLE. BE HUNGRY.
AND ALWAYS BE THE
HARDEST WORKER
IN THE ROOM.”**

