



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

移动应用开发



## Lecture 015: APP Inventor\_enviroment

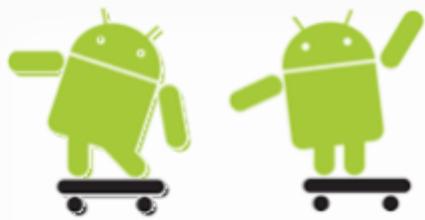
Prof Associate ,  
School of information engineering Jiangxi  
university of science and technology, China

Dr Ata Jahangir Moshayedi



EMAIL: [ajm@jxust.edu.cn](mailto:ajm@jxust.edu.cn)

Autumn \_2021



Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT

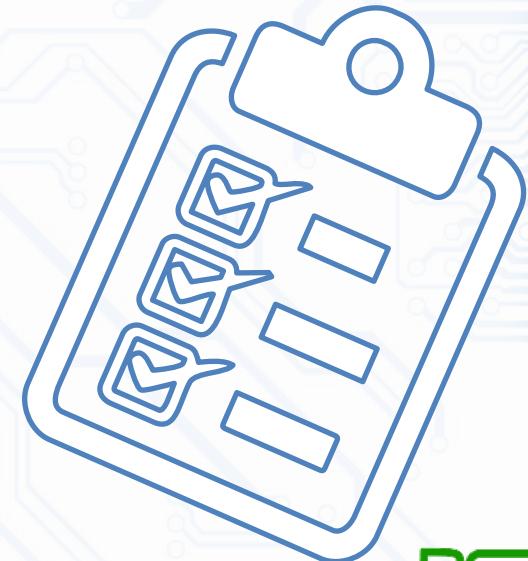
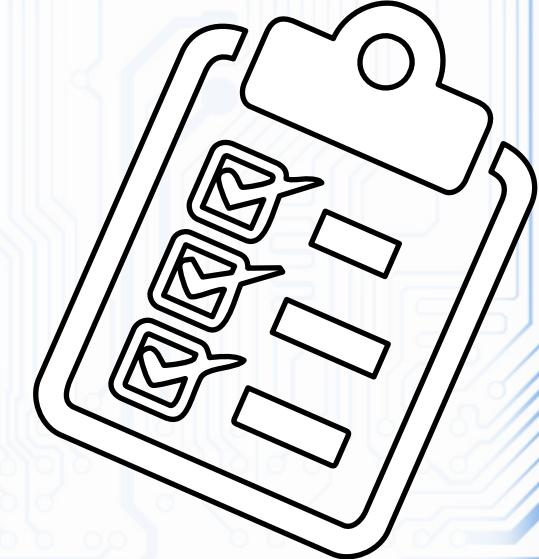
**LECTURE 015:** **APP Inventor\_enviroment**

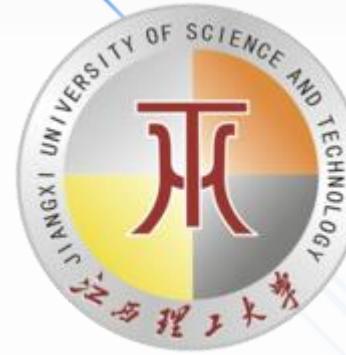
**Storage and Connectivity with example**



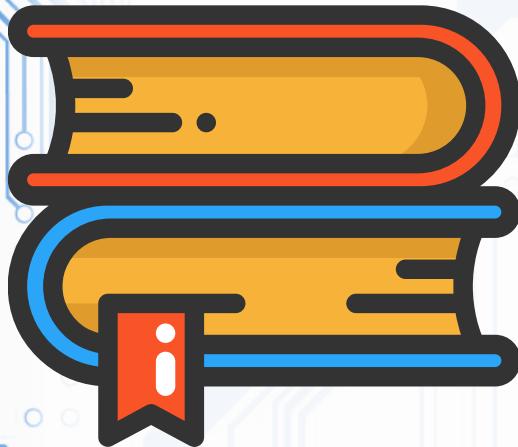
# Agenda

- Storage and Connectivity
- 5 example on tiny DB





Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT

**Example 00:**  
Toggle Menu\_clips

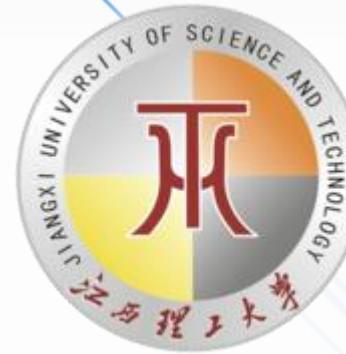
# App Inventor- Toggle Menu



# Process Video [Example 00]

The screenshot displays a dual-monitor setup. The left monitor shows the App Inventor 2 IDE with a project titled "Four\_Example\_Ti". The right monitor shows OBS Studio. In the OBS interface, a camera source is connected to a scene named "Scene 1". The scene includes a "Display Capture" source and an "Audio Mixer" with a "Desktop Audio" input. A "Scene Transitions" effect is applied to the scene, and the "Controls" menu is open, showing options like "Start Streaming", "Start Recording" (which is highlighted), and "Start Virtual Camera". The status bar at the bottom indicates "LIVE: 00:00:00" and "REC: 00:00:00".





Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT

**Example 01:**

**Storage Save and upload  
files File. READ/ WRITE**



## Example 01:

# Storage Save and upload files File READ/ WRITE



## Example aim :

- We can save text in a file and later retrieve them.
- Control File can only be saved as plain text.
- If the file name is **sample.txt** and we are in debug mode with **MIT AI2 Companion**, it is saved in the SD card, specifically in:

**AppInventor/data/sample.txt,**

ie: /mnt/sdcard/AppInventor/data/sample.txt

Also test file://mnt/sdcard/AppInventor/assets/sample.txt



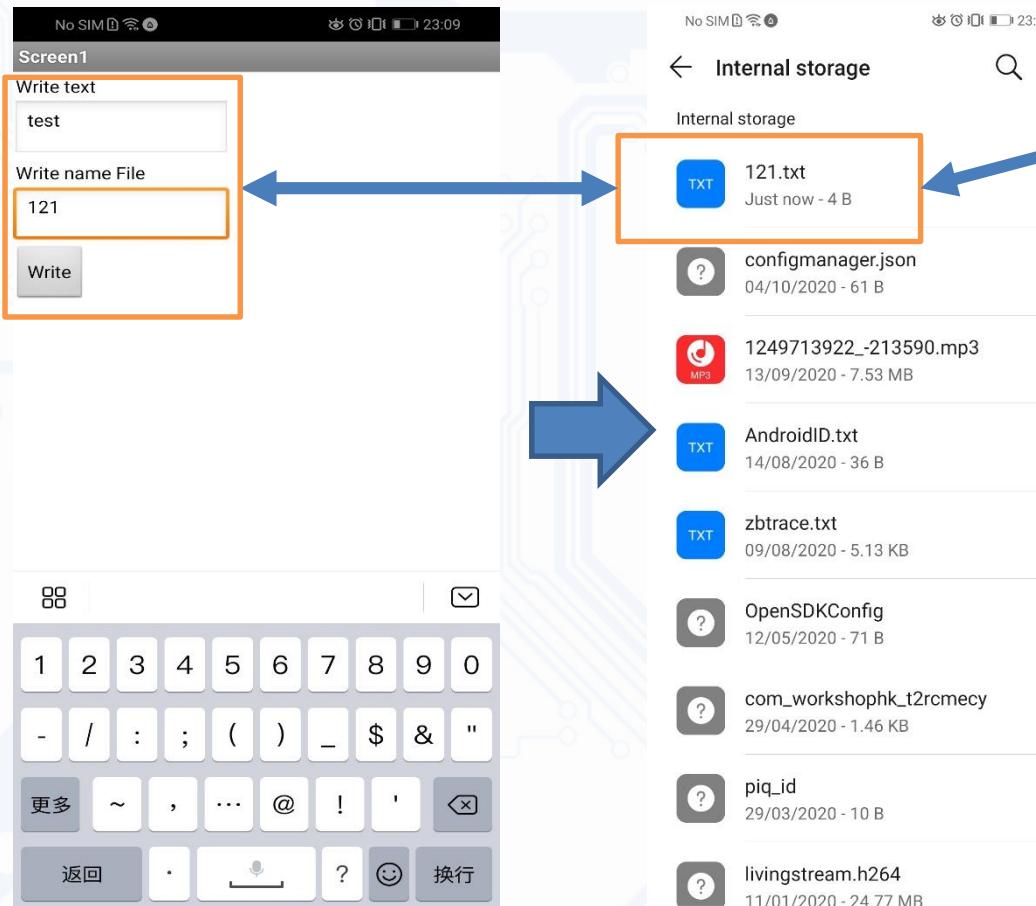
App Inventor

# Example 01:

## Storage Save and upload files File READ/ WRITE



### The file stored location



You can verify that the file is saved in that location using a **File Manager**, you have installed on your Android.

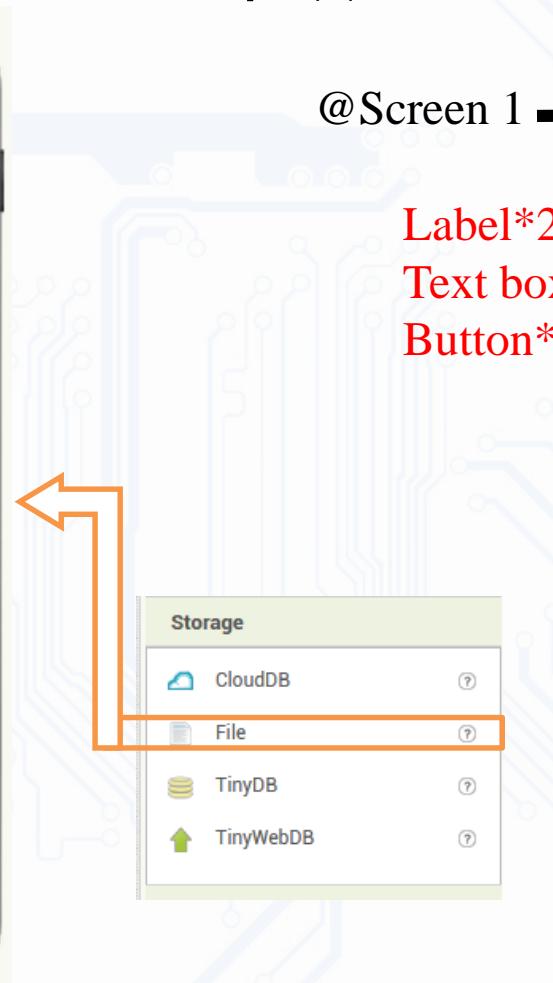
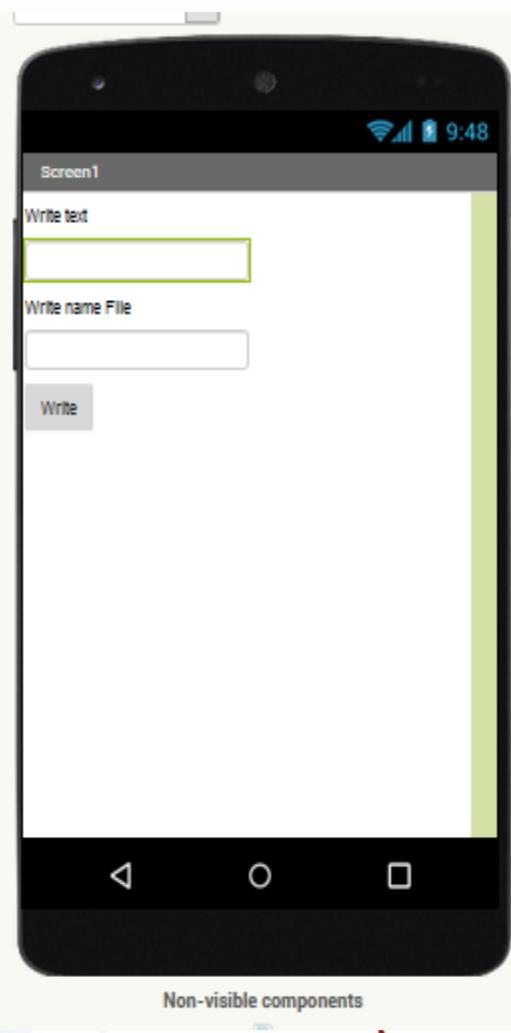


App Inventor



## Example 01:

# Storage Save and upload files File READ/ WRITE





## Example 01:

# Storage Save and upload files File READ/ WRITE



```
initialize global my_text to [red puzzle piece]  
initialize global my_file to [red puzzle piece]  
  
when Button1 .Click  
do  
    set global my_text to TextBox1 . Text  
    set global my_file to TextBox2 . Text  
    call File1 .SaveFile  
        text get global my_text  
        fileName join " / "  
            get global my_file  
            ".txt "
```



# Example 01:

## Storage Save and upload files File READ/ WRITE



MIT App Inventor

www.BANDICAM.com

ai2.appinventor.mit.edu/#4626766707949568

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

Palette

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- Switch
- TextBox
- TimePicker
- WebViewer

viewer

Display hidden components in Viewer

Phone size (505,320)

Screen1

Components

Properties

Screen1

AboutScreen

AccentColor Default

AlignHorizontal Left: 1

AlignVertical Top: 1

AppName ATA\_012

BackgroundColor Default

BackgroundImage None...

BlocksToolkit All

CloseScreenAnimation Default

Icon None...

OpenScreenAnimation Default

PrimaryColor Default

PrimaryColorDark

Rename Delete

Media

Upload File ...

The screenshot shows the MIT App Inventor interface. The central area is the 'viewer' showing a smartphone screen labeled 'Screen1'. To the left is the 'Palette' with various UI components like Button, CheckBox, DatePicker, etc. To the right are the 'Components' and 'Properties' panels for the selected 'Screen1' component. The properties panel includes settings for accent color, alignment, app name, background color, and various animation types. At the bottom, there's a taskbar with icons for Windows, search, and other applications, along with system status information.



# Example 01:

## Storage Save and upload files File READ/ WRITE



MIT App Inventor x + www.BANDICAM.com

Getting Started PID Basic functions related... C بورش زبان JRM | Fuji Technology ... (36) YouTube

MIT APP INVENTOR Projects Connect Build Settings Help My Projects View Trash Guide Report an Issue English moshaydi@gmail.com

ATA\_012 Screen1 Add Screen ... Remove Screen Publish to Gallery Designer Blocks

Blocks Viewer

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

Screen1 Label1 TextBox1 Label2 TextBox2 Button1 File1

Show Warnings

Download audio from this page ? X

Media

Windows taskbar: Search, File, Microsoft Word, Microsoft Excel, Microsoft Powerpoint, Microsoft Edge, Microsoft Teams, Firefox, Task View, Taskbar icons, 23:49, ENG, 24/11/2020



江西理工大学  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

App Inventor



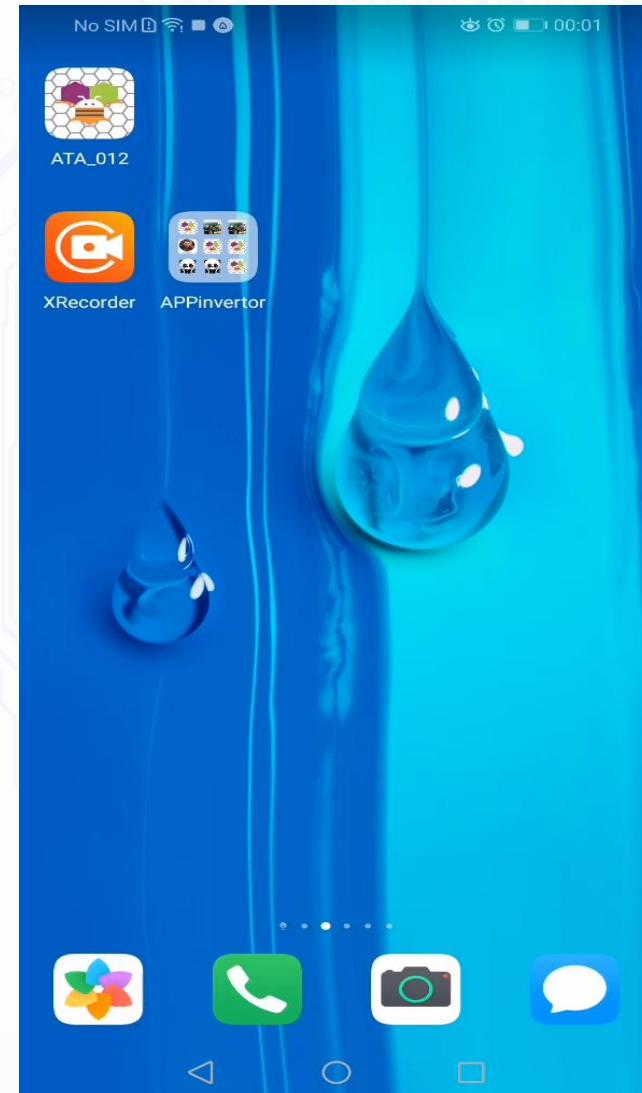
## Example 01:

# Storage Save and upload files File READ/ WRITE



Storage Save and  
upload files File  
WRITE

Demo APP



App Inventor



江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

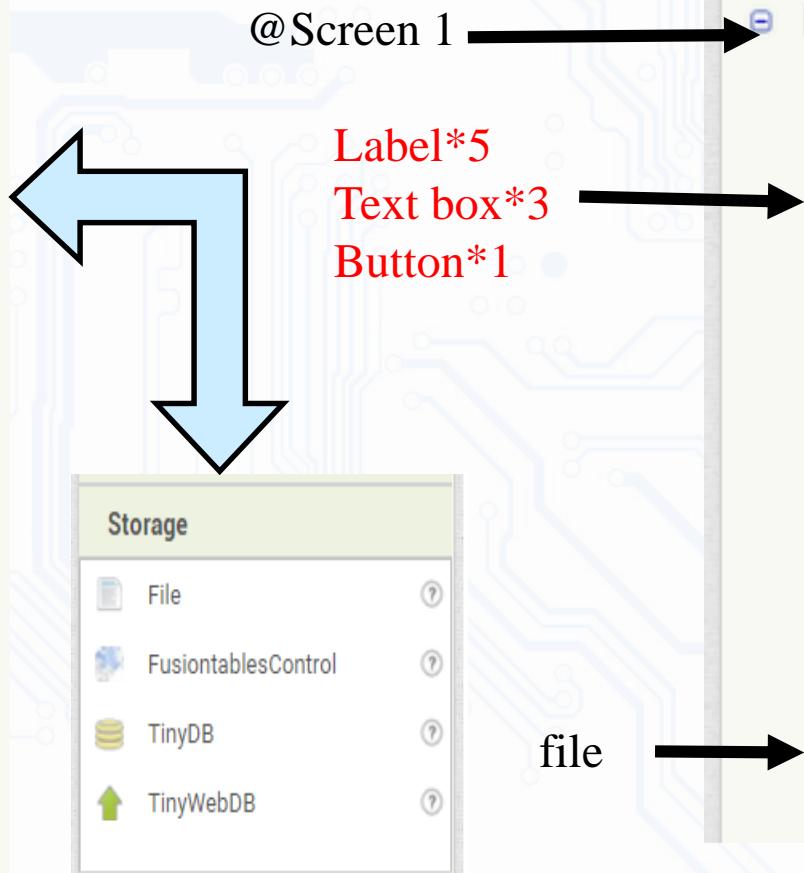
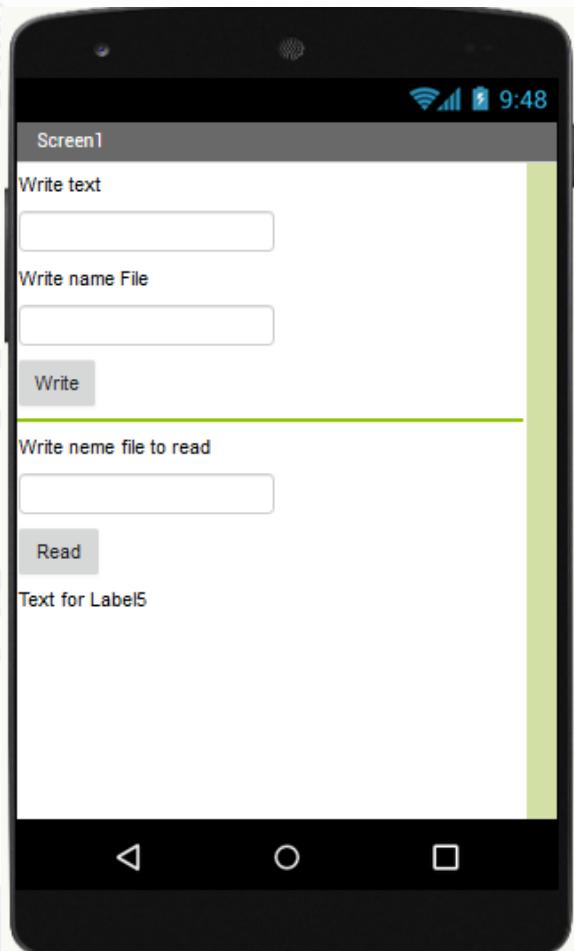


# EXTEND TO READ

We extend the previous project.  
Get the file contents in the Label5.



App Inventor



App Inventor



Non-visible components  
江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

## Example 01:



# Storage Save and upload files File READ/ WRITE



initialize global [my\_text] to [“”]

initialize global [my\_file] to [“”]

when [Button1] .Click

do

- set [global my\_text] to [TextBox1 . Text]
- set [global my\_file] to [TextBox2 . Text]
- call [File1 . SaveFile]
  - text: [get [global my\_text]]
  - fileName: [join [“/”][get [global my\_file]] [“.txt”]]

when [File1] .GotText

[text]

do

- set [Label5 . Text] to [get [text]]

when [Button2] .Click

do

- set [global my\_file] to [TextBox3 . Text]
- call [File1 . ReadFrom]
  - fileName: [join [“/”][get [global my\_file]] [“.txt”]]



App Inventor



JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



App Inventor

MIT App Inventor    New Tab    www.BANDICAM.com

Getting Started   PID   Basic functions related...   C   بورش زبان   JRM | Fuji Technology ...   YouTube (36)

Palette   Viewer   Components   Properties

Search Components...

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- Switch
- TextBox
- TimePicker
- WebViewer

Display hidden components in Viewer  
Phone size (505,320)

Screen1

Screen1

Write text

Write name File

Write

Rename   Delete

Components

- Screen1
  - Label1
  - TextBox1
  - Label2
  - TextBox2
  - Button1
  - File1

Properties

Screen1

AboutScreen

AccentColor Default

AlignHorizontal Left : 1

AlignVertical Top : 1

AppName ATA\_012

BackgroundColor Default

BackgroundImage None...

BlocksToolkit All

CloseScreenAnimation Default

Icon None...

OpenScreenAnimation Default

Media

Upload File ...



江西理工大学  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

00:09  
25/11/2020

ENG

App Inventor



App Inventor

MIT App Inventor    New Tab    www.BANDICAM.com

Getting Started   PID   Basic functions related...   سوچش زبان   JRM | Fuji Technology ...   YouTube

Palette   Viewer   Components   Properties

Search Components...

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- Switch
- TextBox
- TimePicker
- WebViewer

Display hidden components in Viewer   Phone size (505,320)

Screen1

Write text  
[Text Box]

Write name File  
[Text Box]

Write  
[Text Button]

Write neme file to read  
[Text Box]

Read  
[Text Button]

Text for Label5

Components

- Screen1
  - Label1
  - TextBox1
  - Label2
  - TextBox2
  - Button1
  - Label3
  - Label4
  - TextBox3
  - Button2
  - Label5
  - File1

Properties

Label3

BackgroundColor: None

FontBold: False

FontItalic: False

FontSize: 14.0

FontTypeface: default

HTMLFormat: False

HasMargins: True

Height: Automatic...

Width: Fill parent...

Text:

TextAlignment: left : 0

TextColor:

Rename   Delete

Media

Upload File ...



江西理工大学  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

00:12  
25/11/2020



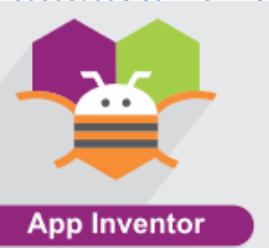
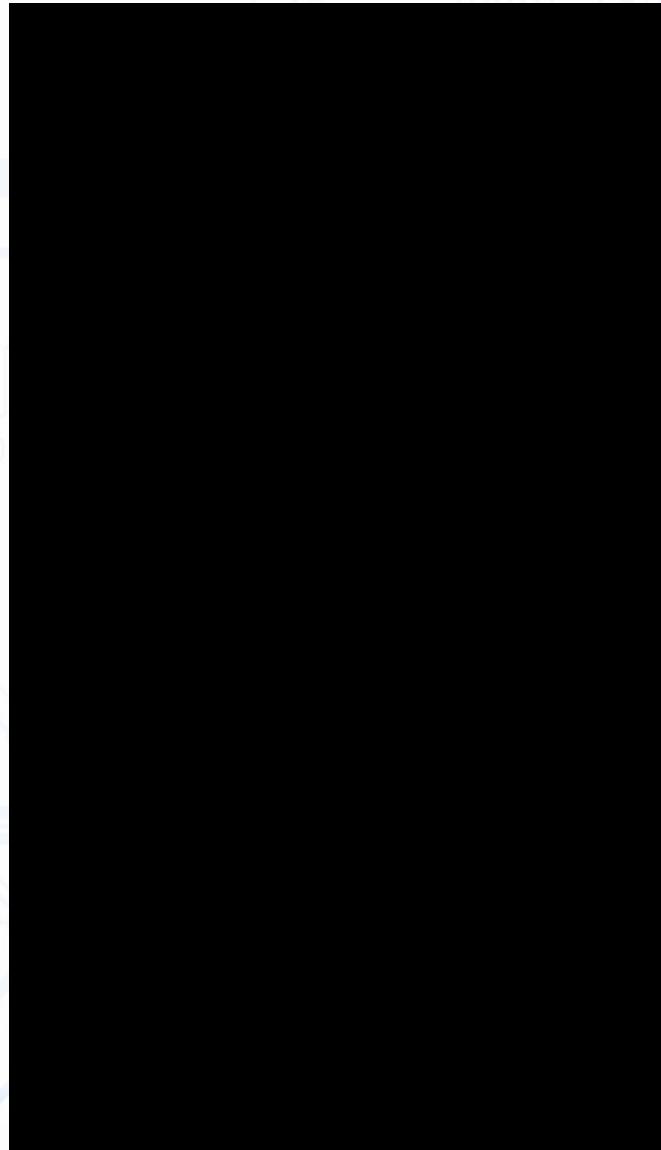
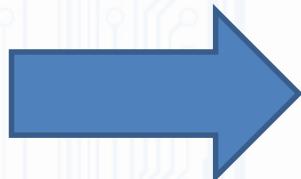
App Inventor

## Example 01:

# Storage Save and upload files File. READ/ WRITE



DEMO APP

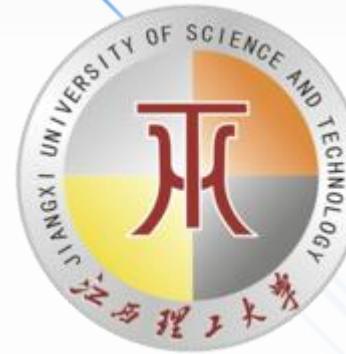


App Inventor



江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT



Example 02:  
**TinyDB (I)**



# Example 02: TinyBD (I)



- **Example Scope:** **TinyDB**, is a database that is in your mobile.
- **Aim:**
  - It is save a person's name and age. We write on **name** of a person, their **age** and click the **Save Button**.
  - Then we get name and age click the **View Button**.
  - The **Delete Button** deletes all data visible on the screen.

TAG Name	Value Age
Name1	Age1
Name2	Age2
Name3	Age3



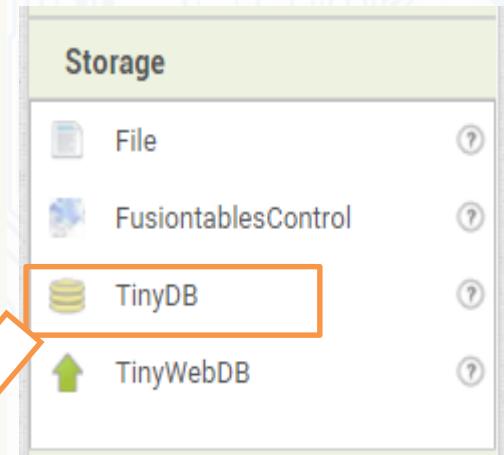
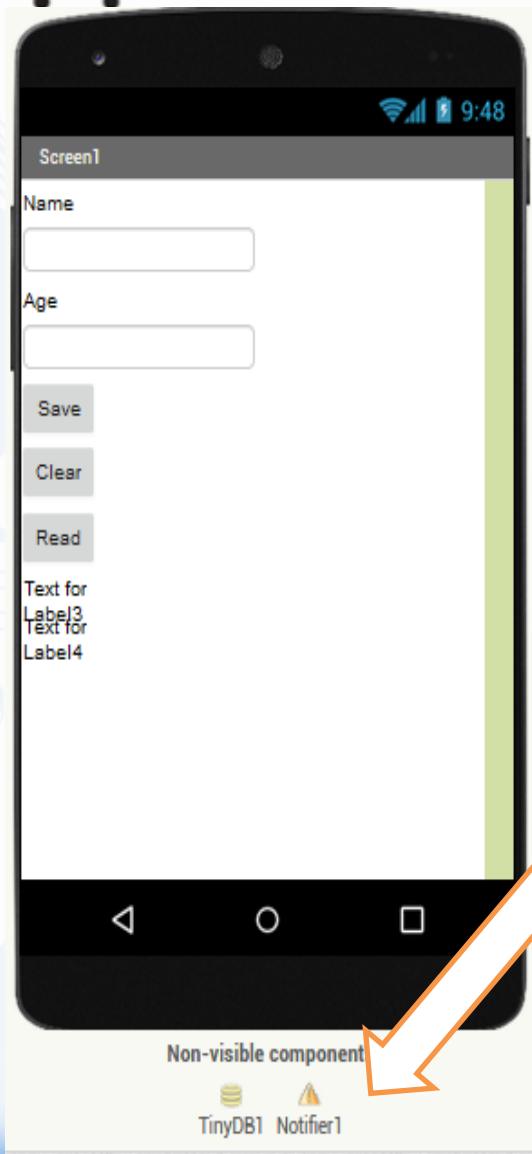
Tag Name	Value Age
Juan	22
Pedro	33
Luis	44



App Inventor



# Example 02: TinyBD (I)



@Screen 1

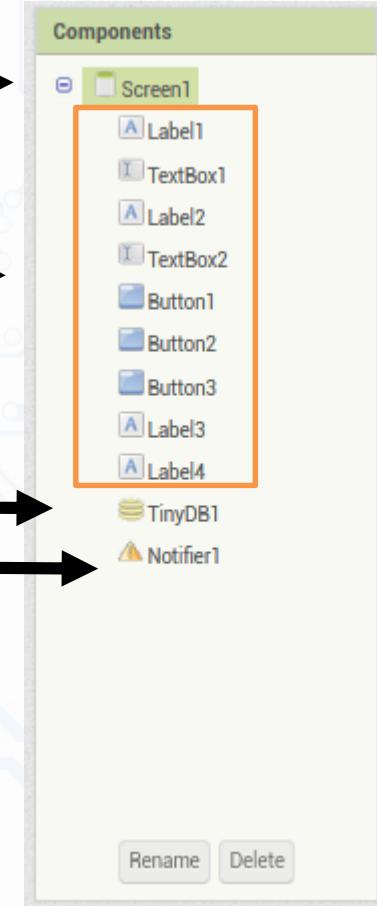
Label\*4

Text box\*2

Button\*3

TinyDB

Notifier



App Inventor



App Inventor



# Example 02: TinyBD (I)



```
when Button1 .Click
do call TinyDB1 .StoreValue
    tag TextBox1 .Text
    valueToStore TextBox2 .Text
call Notifier1 .ShowAlert
    notice " Saved Information "
```

```
when Button2 .Click
do set TextBox1 .Text to " "
    set TextBox2 .Text to " "
    set Label3 .Text to " "
    set Label4 .Text to " "
```

```
when Button3 .Click
do set Label3 .Text to TextBox1 .Text
    set Label4 .Text to call TinyDB1 .GetValue
        tag TextBox1 .Text
        valueIfTagNotThere " NO exist "
```



App Inventor



# Example 02: TinyBD (I)



App Inventor

MIT App Inventor    +    www.BANDICAM.COM

Getting Started   PID   Basic functions related...   بروژ زبان ...   JRM | Fuji Technology ...   (36) YouTube

MIT APP INVENTOR   Projects   Connect   Build   Settings   Help   My Projects   View Trash   Guide   Report an Issue   English   moshaydi@gmail.com

ATA\_014   Screen1   Add Screen ...   Remove Screen   Publish to Gallery   Designer   Blocks

Palette   Search Components...

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- Switch
- TextBox
- TimePicker
- WebViewer

Viewer

Display hidden components in Viewer   Phone size (505,320)

Screen1

Components

Properties

Screen1

AboutScreen

AccentColor Default

AlignHorizontal Left : 1

AlignVertical Top : 1

AppName ATA\_014

BackgroundColor Default

BackgroundImage None...

BlocksToolkit All

CloseScreenAnimation Default

Icon None...

Media

Rename   Delete



App Inventor



# Example 02: TinyBD (I)

MIT App Inventor

www.BANDICAM.com

ai2.appinventor.mit.edu/#6249657044172800

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

90%

Screen1

Name

Age

Save

Clear

Read

Text for Label3

Text for Label4

Non-visible components

TinyDB1 Notifier1

Label2

TextBox2

Button1

Button2

Button3

Label3

Label4

TinyDB1

Notifier1

AccentColor Default

AlignHorizontal Left : 1

AlignVertical Top : 1

AppName ATA\_014

BackgroundColor Default

BackgroundImage None...

BlocksToolkit All

CloseScreenAnimation Default

Icon None...

OpenScreenAnimation Default

PrimaryColor Default

PrimaryColorDark Default

ScreenOrientation Unspecified

Scalable

Rename Delete

Media

Upload File ...





# Example 02: TinyBD (I)



MIT App Inventor

www.BANDICAM.com

ai2.appinventor.mit.edu/#6249657044172800

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

MIT APP INVENTOR Projects Connect Build Settings Help My Projects View Trash Guide Report an Issue English moshaydi@gmail.com

ATA\_014 Screen1 Add Screen ... Remove Screen Publish to Gallery Designer Blocks

Blocks Viewer

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

Screen1 Label1 TextBox1 Label2 TextBox2 Button1 Button2 Button3

Show Warnings 0 0

Designer Blocks

Media

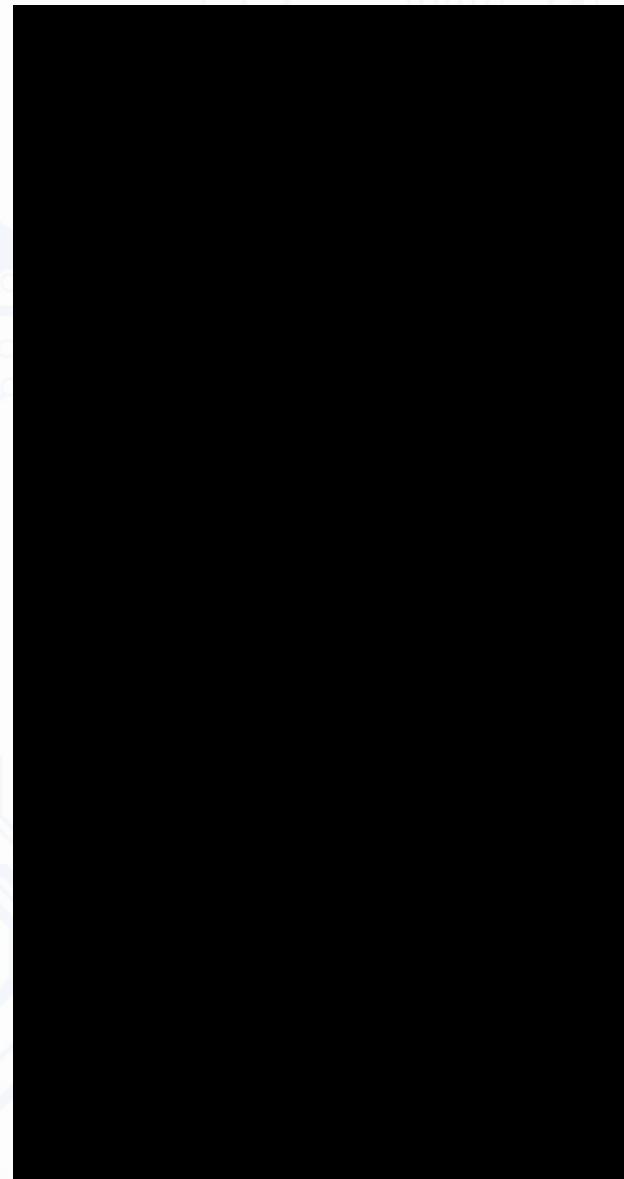
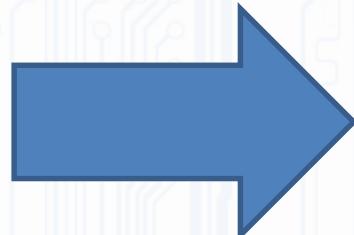
Windows taskbar icons: File Explorer, Edge, Word, Excel, Powerpoint, Edge, Firefox, Camera, Task View, 00:32, ENG, 25/11/2020





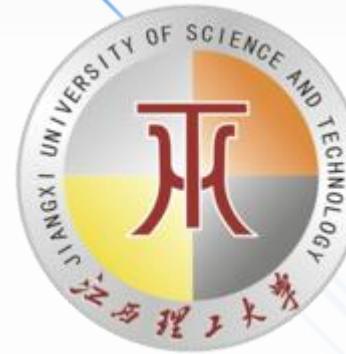
# Example 02: TinyBD (I)

APP DEMO



江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT



## Example 03:

### TinyDB with List (II)

More complicated example of TinyDB



# Example 03:

## TinyDB with List (II)

- In last example We have seen in the above example we can **relate a tag with a value** in a TinyDB, so we keep a Label with their respective value. Sending get the tag name value.
- In this case, each value contains only one element, but let's assume that we want to save the **Name, Age and City** of a person (**Name**) in principle can not do it because each label contains only one value. But through a **List** we can make the value has **several elements**, the elements of the table.



TAG Name	Value Surname, Age, City
Name1	Surname1, Age1, City1
Name2	Surname, Age2, City2
Name3	Surname3, Age3, City3



Tag Name	Value Surname, Age, City
Juan	Perez 22 Cadiz
Pedro	Sanchez 33 Sevilla
Luis	Rodriguez 44 Jerez





# Example 03: TinyDB with List (II)



- In this example We created a list called **person**.

Each time you press the **Button1**, delete the **List**. It is an auxiliary List, use it to enter three values in each name. But it is not accumulative List that will keep all the names, so it is deleted upon arrival of each new name.

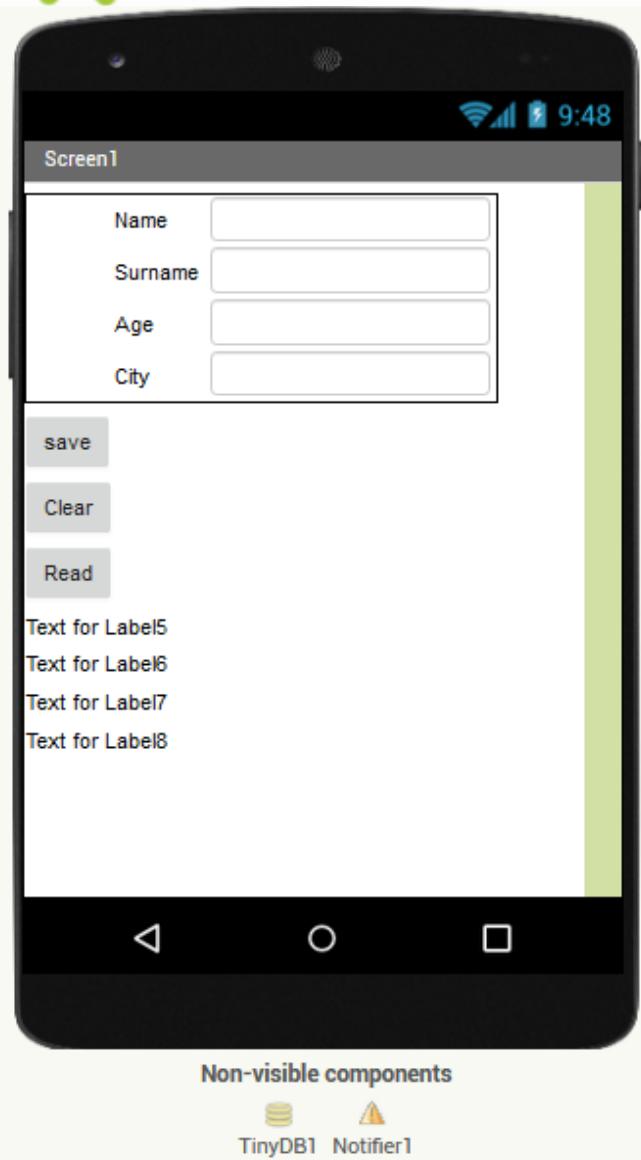
- We enter the data in the **TextBox person**. TinyDB keep on the Label and its corresponding value **that is the person List**.



App Inventor

# Example 03:

## TinyDB with List (II)



@Screen 1

Label\*4  
Text box\*4

Label\*4  
Text box\*4

Label\*4  
Text box\*4

TinyDB  
Notifier



App Inventor

Components

Screen1  
TableArrangement1  
Name  
Textbox  
TextBox2  
TextBox3  
TextBox4  
Surname  
age  
City

Button1  
Button2  
Button3  
Label5  
Label6  
Label7  
Label8

TinyDB1  
Notifier1  
Rename Delete

App Inventor



## Example 03:

# TinyDB with List (II)



- Create a table called empty **sites**.
- - When you press the **Button1**, we can switch between GPS coordinates to obtain or enter them manually. Will press to enter them manually.
- - If we want to obtain GPS, GPS is enabled and the longitude and latitude in lockers.
- - Also added to the **List sites**, the name of the sites.
  - Also introduced into the **ListView1** all elements of the places List.
- 



App Inventor



# Example 03:

## TinyDB with List (II)



- When you press the **ListView1**, we will have the elements previously saved **ListView1.items = sites**
- When you click an item in the **ListView1**, it is taken from the **TinyDB** the value of the selected item, this value, in turn, contains two parts, the one that would be the longitude and latitude.
- They are written in their **TextBox** appropriate. That is, **the tag** is the name of the selected item and **value** is the value of that element, which in turn contains two terms, since they were introduced by a list of two parts.
- Again, do not confuse the **sites List**, containing the names of places only and another List that has no name that contains the two coordinates in each of its elements.
- Each time the screen starts, take the tag of TinyDB and introduced into the places list. The list elemenos of places, are introduced into the ListPicker.
- The **Button3** use it to **clean the entire** database.
- The last item will last in the **ListView1**.

location (Tag TinyDB) TextBox1	longitude, latitude is introduced both in a list and stored. TextBox2 and TextBox3
Grazalema	-6.16, 36.21
Jerez	-6.15, 36.24
Puerto Real	-6.16, 36.52



App Inventor



# Example 03:

## TinyDB with List (II)



```
initialize global person to create empty list  
when Button1 .Click  
do set global person to create empty list  
  add items to list list get global person  
    item TextBox2 .Text  
    item TextBox3 .Text  
    item TextBox4 .Text  
call TinyDB1 .StoreValue  
  tag Textbox .Text  
  valueToStore get global person  
call Notifier1 .LogWarning  
  message " Saved information "
```

```
when Button2 .Click  
do set TextBox .Text to *  
  set TextBox2 .Text to *  
  set TextBox3 .Text to *  
  set TextBox4 .Text to *  
  set Label5 .Text to *  
  set Label6 .Text to *  
  set Label7 .Text to *  
  set Label8 .Text to *  
  
when Button3 .Click  
do set global person to call TinyDB1 .GetValue  
  tag Textbox .Text  
  valueIfTagNotThere * NO EXIST *  
  set Label5 .Text to Textbox .Text  
  set Label6 .Text to select list item list get global person  
    index 1  
  set Label7 .Text to select list item list get global person  
    index 2  
  set Label8 .Text to select list item list get global person  
    index 3
```





# Example 03:

## TinyDB with List (II)

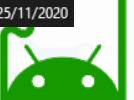


Screenshot of the MIT App Inventor interface showing the creation of a mobile application.

The browser address bar shows: [ai2.appinventor.mit.edu/#4966462885593088](http://ai2.appinventor.mit.edu/#4966462885593088)

The interface includes:

- Palette:** A sidebar containing categories like User Interface (Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker, WebViewer), Layout, Media, Drawing and Animation, Maps, and Sensors.
- Viewer:** Displays a smartphone screen titled "Screen1" with a blank white background.
- Components:** A list of components including "Screen1".
- Properties:** A panel showing properties for "Screen1", such as AccentColor (Default), AlignHorizontal (Left : 1), AlignVertical (Top : 1), AppName (ATA\_015), BackgroundColor (Default), BackgroundImage (None...), BlocksToolkit (All), CloseScreenAnimation (Default), Icon (None...), OpenScreenAnimation (Default), PrimaryColor (Default), PrimaryColorDark (Default), ScreenOrientation (Unspecified), and Scrollable.



App Inventor



# Example 03:

## TinyDB with List (II)



Screenshot of the MIT App Inventor 2 interface showing a project titled "ATA\_015".

The project interface includes:

- Palette:** On the left, categories include Notifier, User Interface (Layout: HorizontalArrangement, HorizontalScrollView, TableArrangement, VerticalArrangement, VerticalScrollView), Media, Drawing and Animation, Maps, Sensors, Social, Storage, Connectivity, LEGO® MINDSTORMS®, and Experimental.
- Viewer:** Shows a smartphone screen for "Screen1" with four text input fields for Name, Surname, Age, and City, and three buttons for save, Clear, and Read. Below the screen are four text labels: Text for Label5, Text for Label6, Text for Label7, and Text for Label8.
- Components:** A tree view of components used in the screen:
  - Screen1
    - TableArrangement1
      - Name
      - TextBox
      - TextBox2
      - TextBox3
      - TextBox4
      - Surname
      - age
      - City
    - Button1
    - Button2
    - Button3
    - Label5
    - Label6
    - Label7
    - Label8
    - TinyDB1
- Properties:** A panel on the right showing properties for the selected component (Button3). Properties include:
  - BackgroundColor: Default
  - Enabled: checked
  - FontBold: unchecked
  - FontItalic: unchecked
  - FontSize: 14.0
  - FontTypeface: default
  - Height: Automatic...
  - Width: Automatic...
  - Image: None...
  - Shape: default
  - ShowFeedback: checked



# Example 03:

## TinyDB with List (II)

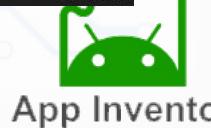


Screenshot of the MIT App Inventor 2 environment showing a project titled "ATA\_015".

The project contains the following blocks:

- when Button1 .Click**:
  - do
    - set global person to create empty list
    - add items to list item TextBox2 .Text to get global person
    - item TextBox3 .Text to get global person
    - item TextBox4 .Text to get global person
  - call TinyDB1 .StoreValue tag Textbox .Text valueToStore get global person
  - call Notifier1 .LogWarning message "Saved information"
- when Button2 .Click**:
  - do
    - set Textbox .Text to
    - set TextBox2 .Text to
    - set TextBox3 .Text to
    - set TextBox4 .Text to
    - set Label5 .Text to
    - set Label6 .Text to
    - set Label7 .Text to
    - set Label8 .Text to
- when Button3 .Click**:
  - do
    - set global person to call TinyDB1 .GetValue tag valueIfTagNotThere
  - set Label5 .Text to Textbox .Text
  - set Label5 .Text to

The interface includes a sidebar with "Blocks" and "Viewer" tabs, and a toolbar at the bottom.





# Example 03:

## TinyDB with List (II)



Screenshot of the MIT App Inventor 2 environment showing a project titled "ATA\_015".

The project contains two screens:

- Screen1:** Contains four text boxes (TextBox2, TextBox3, TextBox4) and four labels (Label5, Label6, Label7, Label8). It has the following blocks:
  - when Button1.Click: set global person to create empty list; add items to list list get global person item TextBox2.Text item TextBox3.Text item TextBox4.Text; call TinyDB1.StoreValue tag TextBox1.Text valueToStore get global person; call Notifier1.LogWarning message "Saved information"
  - when Button2.Click: set TextBox1.Text to [ ]; set TextBox2.Text to [ ]; set TextBox3.Text to [ ]; set TextBox4.Text to [ ]; set Label5.Text to [ ]; set Label6.Text to [ ]; set Label7.Text to [ ]; set Label8.Text to [ ]
  - when Button3.Click: set global person to call TinyDB1.GetValue tag TextBox1.Text valueIfTagNotThere "NO EXIST"; set Label6.Text to select list item list get global person index 1; set Label7.Text to select list item list get global person index 2; set Label8.Text to select list item list get global person index 3
- Screen2:** Contains a single text box (TextBox1) and three labels (Label1, Label2, Label3). It has the following blocks:
  - when Button1.Click: set global person to create empty list; add items to list list get global person item TextBox1.Text item TextBox2.Text item TextBox3.Text; call TinyDB1.StoreValue tag TextBox1.Text valueToStore get global person; call Notifier1.LogWarning message "Saved information"
  - when Button2.Click: set TextBox1.Text to [ ]; set TextBox2.Text to [ ]; set TextBox3.Text to [ ]
  - when Button3.Click: set global person to call TinyDB1.GetValue tag TextBox1.Text valueIfTagNotThere "NO EXIST"; set Label1.Text to select list item list get global person index 1; set Label2.Text to select list item list get global person index 2; set Label3.Text to select list item list get global person index 3

The sidebar shows the blocks catalog under "Built-in" and "Screen1". The status bar at the bottom shows system icons and the time "20:39 25/11/2020".



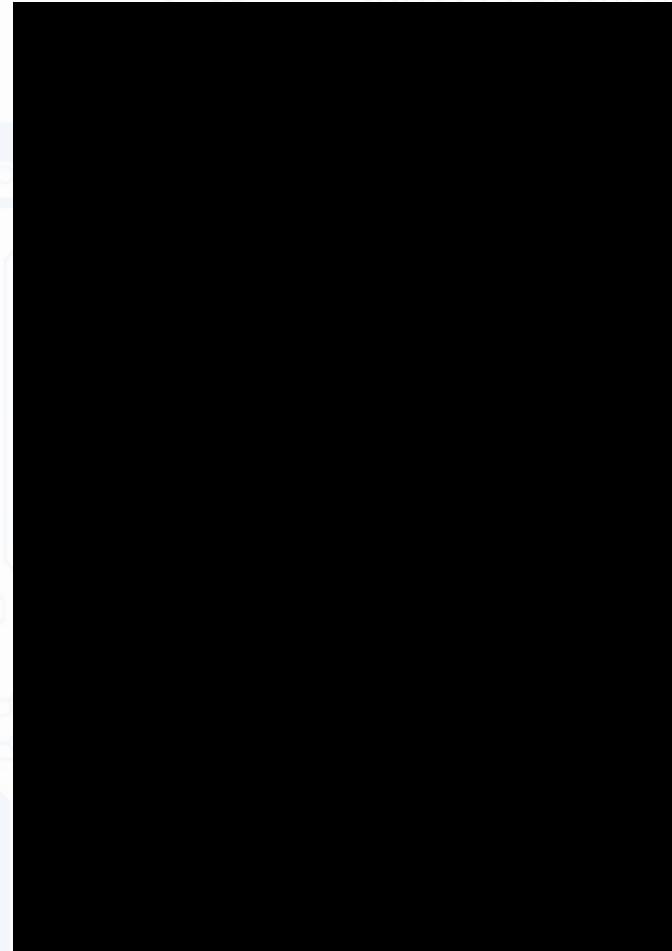
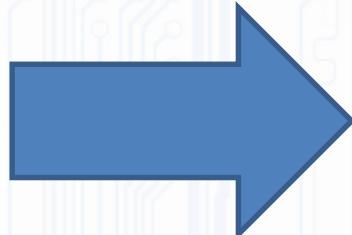
App Inventor



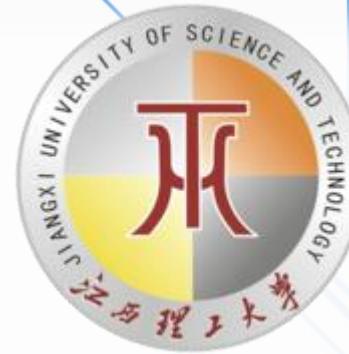
## Example 03: TinyDB with List (II)



# APP DEMO



App Inventor



Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT

**Example 04:**

**TinyDB (III)**

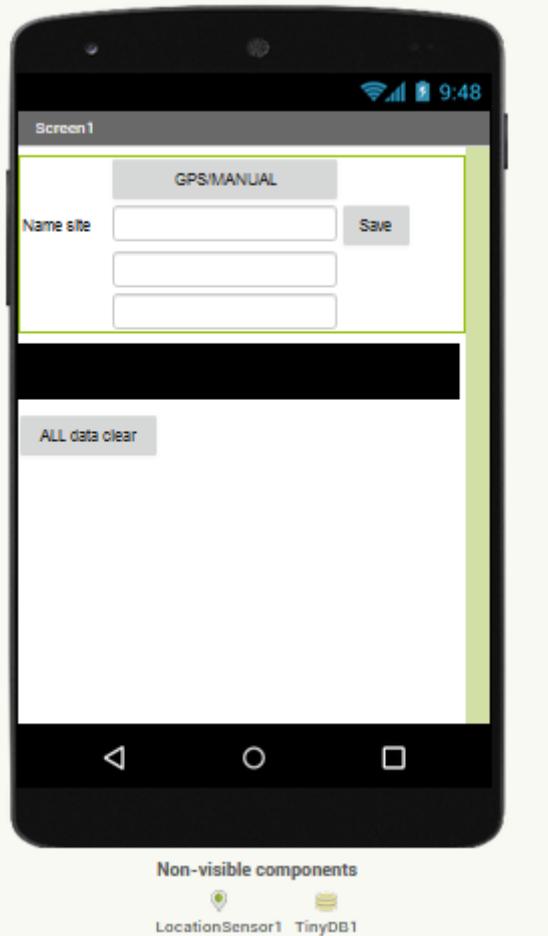
**Example TinyDB more difficult.**



# Example 04:



## TinyDB (III) Example TinyDB more difficult.



@Screen 1

Button \*2  
Label \*1  
Text box\*4

Listview  
Button \*1

Location sensor1  
TinyDB

Components

Screen1  
TableArrangement1

Button1  
Label1  
TextBox1  
TextBox2  
Button2  
TextBox3

ListView1  
Button3  
LocationSensor1  
TinyDB1



App Inventor



江西理工大学  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



# Example 04:

## TinyDB (III)

### Example TinyDB more difficult.

```
when Button1 . Click
do if LocationSensor1 . Enabled .
then set LocationSensor1 . Enabled . to false .
set Button1 . Text . to " Intro datas "
else set LocationSensor1 . Enabled . to true .
set Button1 . Text . to " Datas by GPS "
set TextBox2 . Text . to LocationSensor1 . Longitude .
set TextBox2 . Text . to LocationSensor1 . Latitude .
```

KIO4.COM

```
when LocationSensor1 . LocationChanged
latitude longitude altitude
do set TextBox2 . Text . to get longitude .
set TextBox3 . Text . to get latitude .
```

```
when Button2 . Click
do if is in list? thing TextBox1 . Text .
list get global sites .
then call TinyDB1 . ClearTag
tag TextBox1 . Text .
remove list item list get global sites .
index index in list thing TextBox1 . Text .
list get global sites .
call TinyDB1 . StoreValue
tag TextBox1 . Text .
valueToStore make a list TextBox2 . Text .
TextBox3 . Text .
add items to list list get global sites .
item TextBox1 . Text .
set ListView1 . Elements . to get global sites .
set TextBox1 . Text . to " " .
set TextBox2 . Text . to " " .
set TextBox3 . Text . to " " .
```

```
when ListView1 . AfterPicking
do set TextBox1 . Text . to ListView1 . Selection .
set TextBox2 . Text . to select list item list index
call TinyDB1 . GetValue
tag ListView1 . Selection .
valuelfTagNotThere " Not exist. "
set TextBox3 . Text . to select list item list index
call TinyDB1 . GetValue
tag ListView1 . Selection .
valuelfTagNotThere " Not exist. "
```

```
when Screen1 . Initialize
do set global sites . to call TinyDB1 . GetTags
set ListView1 . Elements . to get global sites .
```



# Example 04:

TinyDB (III) Example TinyDB more difficult.



App Inventor

- When you press the **Button2**, it is checked whether the place name already exists in the list **sites**.  
- If there is, remove it (**ClearTag**) of the Database list TinyDB and **sites**.
- Whether or not, **stores** the writing element in the **TextBos1** on a **tag** of TinyDB and lockers of longitude and latitude, put them in **another list of two elements** and saves as a value in the **TinyDB**.
- That is, keep **the tag** that would be the name of the town and two items using a list, which would be the longitude and latitude. Do not confuse this list of two elements with locations list, they are different.



App Inventor



# Example 04:

## TinyDB (III) Example TinyDB more difficult.



- Also added to the **List sites**, the name of the sites.
- Also introduced into the **ListView1** all elements of the places List.
- When you press the **ListView1**, we will have the elements previously saved  
**ListView1.items** = sites
- When you click an item in the **ListView1**, it is taken from the **TinyDB** the value of the selected item, this value, in turn, contains two parts, the one that would be the longitude and latitude.
- They are written in their **TextBox** appropriate. That is, **the tag** is the name of the selected item and **value** is the value of that element, which in turn contains two terms, since they were introduced by a list of two parts.
- Again, do not confuse the **sites List**, containing the names of places only and another List that has no name that contains the two coordinates in each of its elements.
- Each time the screen starts, take the tag of TinyDB and introduced into the places list. The list elemenos of places, are introduced into the ListPicker.
- The **Button3** use it to **clean the entire** database.
- The last item will last in the ListView1.

Location(Tag TinyDB) Text Box1	Longitude,latitude is introduced both in list and stored Textbox 2 and Textbox3
Name1	Long,lat
Name2	Long,lat
Name3	Long,lat



App Inventor



## Example 04:

# TinyDB (III), Example TinyDB more difficult.



App Inventor

```

initialize global sites to create empty list
when Button1 .Click
do if LocationSensor1 .Enabled then
    set LocationSensor1 .Enabled to false
    set Button1 .Text to "Intro datas"
else
    set LocationSensor1 .Enabled to true
    set Button1 .Text to "Datas by GPS"
    set TextBox2 .Text to LocationSensor1 .Longitude
    set TextBox2 .Text to LocationSensor1 .Latitude

when ListView1 .AfterPicking
do set TextBox1 .Text to ListView1 .Selection
set TextBox1 .Text to select list item list call TinyDB1 .GetValue
tag ListView1 .Selection
valueIfTagNotThere "No exist"
index 1

when Screen1 .BackPressed
do set global sites to call TinyDB1 .GetTags
set ListView1 .Elements to get global sites

```

```

when LocationSensor1 .LocationChanged
latitude longitude altitude speed
do set TextBox2 .Text to get longitude
set TextBox2 .Text to get altitude

when Button2 .Click
do if is in list? thing TextBox1 .Text
list get global sites
then call TinyDB1 .ClearTag
tag TextBox1 .Text
remove list item list get global sites
index index in list thing TextBox1 .Text
list get global sites

call TinyDB1 .StoreValue
tag TextBox1 .Text
valueToStore make a list TextBox2 .Text
TextBox2 .Text

add items to list list get global sites
item TextBox1 .Text
set ListView1 .Elements to get global sites
set TextBox1 .Text to ""
set TextBox2 .Text to ""

```



江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



App Inventor



## Example 04:

### TinyDB (III), Example TinyDB more difficult.



App Inventor

```
when LocationSensor1 .LocationChanged
  latitude longitude altitude speed
do
  set TextBox2 .Text to get longitude
  set TextBox2 .Text to get altitude
```

```
when Button2 .Click
do
  if is in list? thing TextBox1 .Text
    list get global sites
  then call TinyDB1 .ClearTag
    tag TextBox1 .Text
    remove list item list get global sites
    index index in list thing TextBox1 .Text
    list get global sites
  call TinyDB1 .StoreValue
    tag TextBox1 .Text
    valueToStore make a list TextBox2 .Text
    TextBox2 .Text
  add items to list list get global sites
    item TextBox1 .Text
  set ListView1 .Elements to get global sites
  set TextBox1 .Text to ""
  set TextBox2 .Text to ""
```

```
initialize global sites to create empty list
when Button1 .Click
do
  if LocationSensor1 .Enabled
  then set LocationSensor1 .Enabled to false
    set Button1 .Text to "Intro data"
  else set LocationSensor1 .Enabled to true
    set Button1 .Text to "Datas by GPS"
  set TextBox2 .Text to LocationSensor1 .Longitude
  set TextBox2 .Text to LocationSensor1 .Latitude
when ListView1 .AfterPicking
do
  set TextBox1 .Text to ListView1 .Selection
  set TextBox1 .Text to select list item list call TinyDB1 .GetValue
    tag ListView1 .Selection
    valueIfTagNotThere "No exist"
    index 1
when Screen1 .BackPressed
do
  set global sites to call TinyDB1 .GetTags
  set ListView1 .Elements to get global sites
```



App Inventor



## Example 04:

# TinyDB (III), Example TinyDB more difficult.



App Inventor

App inventor. File. TinyDB. Tiny X MIT App Inventor X www.BANDICAM.com

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

Palette Viewer Components Properties

Search Components...

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- Switch
- TextBox
- TimePicker
- WebViewer

Display hidden components in Viewer  
Phone size (505,320)

Screen1

Screen1

Rename Delete

Media

Upload File ...

AccentColor Default

AlignHorizontal Left : 1

AlignVertical Top : 1

AppName ATA\_016

BackgroundColor Default

BackgroundImage None...

BlocksToolkit All

CloseScreenAnimation Default

Icon None...

OpenScreenAnimation Default

www.BANDICAM.com

21:44 ENG 25/11/2020



# Example 04:

## TinyDB (III), Example TinyDB more difficult.



App Inventor



## Example 04:

# TinyDB (III), Example TinyDB more difficult.



App Inventor

App inventor. File. TinyDB. Tiny X MIT App Inventor + www.BANDICAM.com

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

Blocks Viewer

```
initialize global sites to create empty list
when Button1 .Click
do
  if LocationSensor1 . Enabled
    then set LocationSensor1 . Enabled to false
    set Button1 . Text to "Intro datas"
  else set LocationSensor1 . Enabled to true
    set Button1 . Text to "Datas by GPS"
    set TextBox2 . Text to LocationSensor1 . Longitude
    set TextBox2 . Text to LocationSensor1 . Latitude

```

Lists Dictionaries Colors Variables Procedures Screen1 Button1 TableArrangement1 Button2 Label TextBox1 TextBox2 TextBox3 ListView1 Button3 TinyDB LocationSensor1 Any component Rename Delete Media Upload File ...

21:56 25/11/2020

This screenshot shows the MIT App Inventor interface for creating an Android application. The title bar indicates the file is named 'TinyDB. Tiny'. The main workspace displays a Scratch-like script editor with blocks for initializing global variables, handling button clicks, and setting sensor and text box values. The left sidebar lists available components: Lists, Dictionaries, Colors, Variables, Procedures, Screen1, Button1, TableArrangement1, Button2, Label, TextBox1, TextBox2, TextBox3, ListView1, Button3, TinyDB, LocationSensor1, and Any component. Buttons for 'Rename' and 'Delete' are also present. The bottom taskbar includes icons for various Windows applications like File Explorer, Task Manager, and a browser, along with the system clock and date.



# Example 04:

## TinyDB (III), Example TinyDB more difficult.



App Inventor

MIT App Inventor    www.BANDICAM.com

ai2.appinventor.mit.edu/#4973125316247552

Blocks    Viewer

Blocks List:

- Built-in
  - Control
  - Logic
  - Math
  - Text
  - Lists
  - Dictionaries
  - Colors
  - Variables
  - Procedures
- Screen1
- TableArrangement1
  - Button2
  - Label1
  - TextBox1
  - TextBox2
  - TextBox3

Viewer Area:

```
set TextBox2 . Text to LocationSensor1 . Longitude
set TextBox2 . Text to LocationSensor1 . Latitude
then
    call TinyDB1 . ClearTag
    tag TextBox1 . Text
    remove list item list get global sites
    index index in list thing TextBox1 . Text
    list get global sites
call TinyDB1 . StoreValue
    tag TextBox1 . Text
    valueToStore make a list TextBox2 . Text
    TextBox3 . Text
    add items to list list get global sites
    item TextBox1 . Text
set ListView1 . Elements to get global sites
set TextBox1 . Text to *
set TextBox2 . Text to *
set TextBox3 . Text to *
```

Warnings:

- ⚠ 0
- ✖ 0

Show Warnings

Media

Upload File ...

Windows Taskbar:

22:03 25/11/2020 ENG

App Inventor

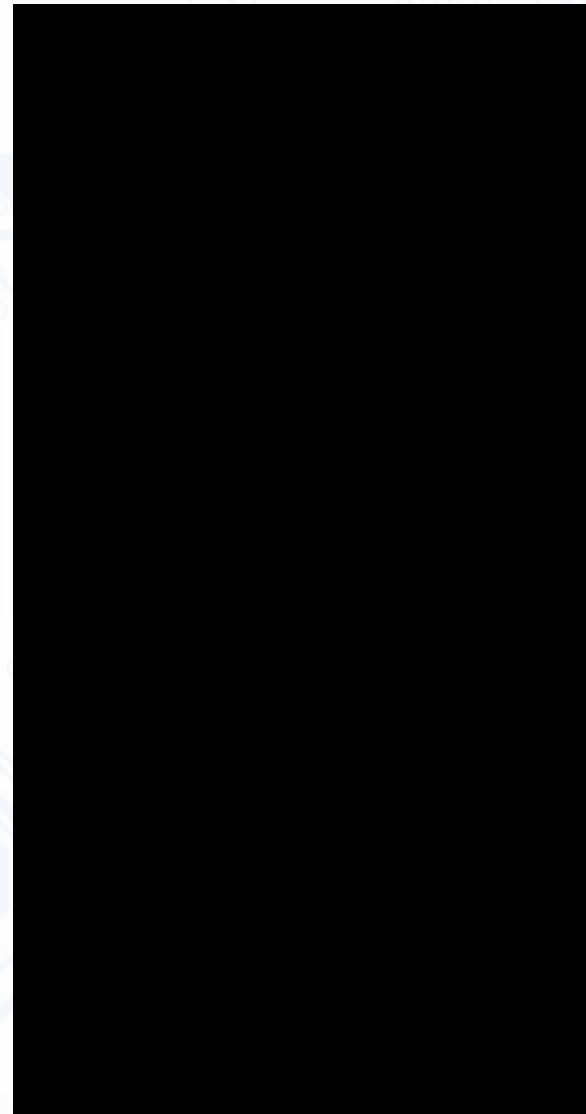
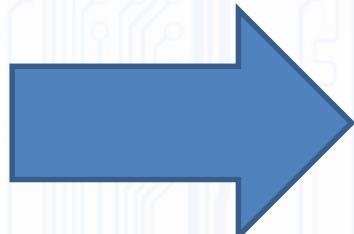


## Example 04:

**TinyDB (III), Example TinyDB more difficult.**



# APP DEMO



江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



App Inventor



# Student Task\_11



**Repeat this examples and make based on our task format**

- We need the video of processing for example 00 and GPS ( full format) rest of example we need demo
- Add this part to example GPS( example 04):
  - Solve the location problem
  - After push the clear remove all data and make data base empty

**Next lecture**

- You have time to send your task before 9 am of lecture
- Send the file in PPT(power point format) to this email :  
**[drajm@yahoo.com](mailto:drajm@yahoo.com)**
- Your file should have this format of name  
**<Task number><student name><Student ID>.ppt**



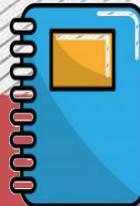


# Android



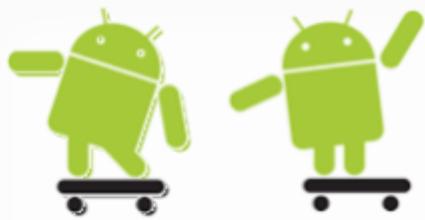
Dr Ata Jahangir Moshayedi

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

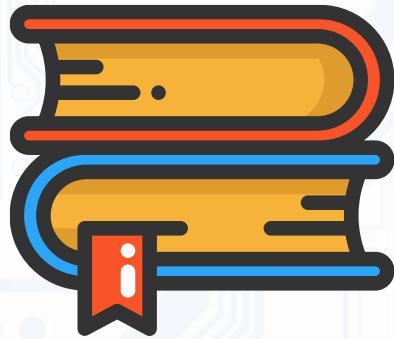


EMAIL: [ajm@jxust.edu.cn](mailto:ajm@jxust.edu.cn)

Autumn \_2020



Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT

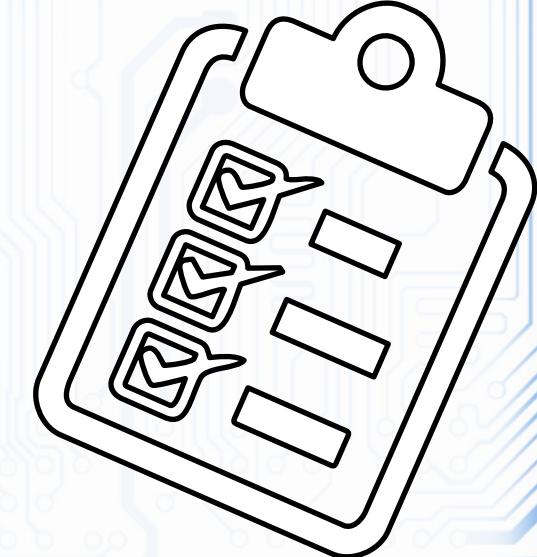
**LECTURE 015:** **APP Inventor\_enviroment**

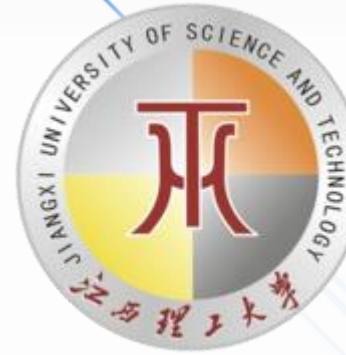
**Storage and Connectivity with example**



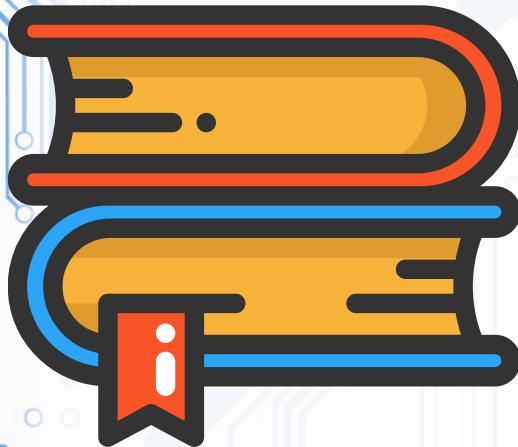
# Agenda

- Storage and Connectivity
- 5 example on tiny DB





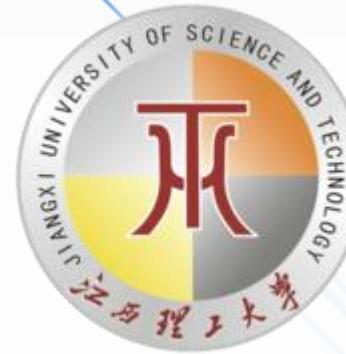
Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT

**Example 00:**  
Toggle Menu\_clips

# App Inventor- Toggle Menu



Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT

**Example 01:**

**Storage Save and upload  
files File. READ/ WRITE**



## Example 01:

# Storage Save and upload files File READ/ WRITE



## Example aim :

- We can save text in a file and later retrieve them.
- Control File can only be saved as plain text.
- If the file name is **sample.txt** and we are in debug mode with **MIT AI2 Companion**, it is saved in the SD card, specifically in:

**AppInventor/data/sample.txt,**

ie: /mnt/sdcard/AppInventor/data/sample.txt

Also test file://mnt/sdcard/AppInventor/assets/sample.txt



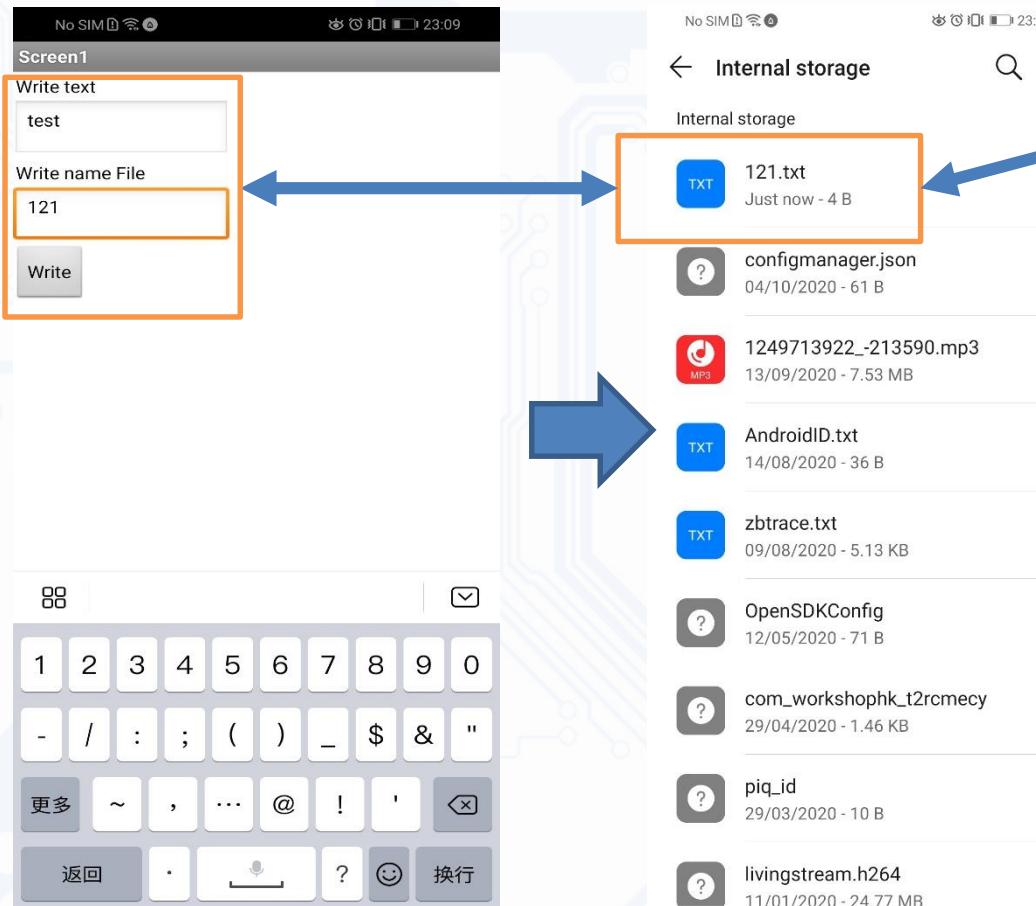
App Inventor

# Example 01:

## Storage Save and upload files File READ/ WRITE



### The file stored location



You can verify that the file is saved in that location using a **File Manager**, you have installed on your Android.

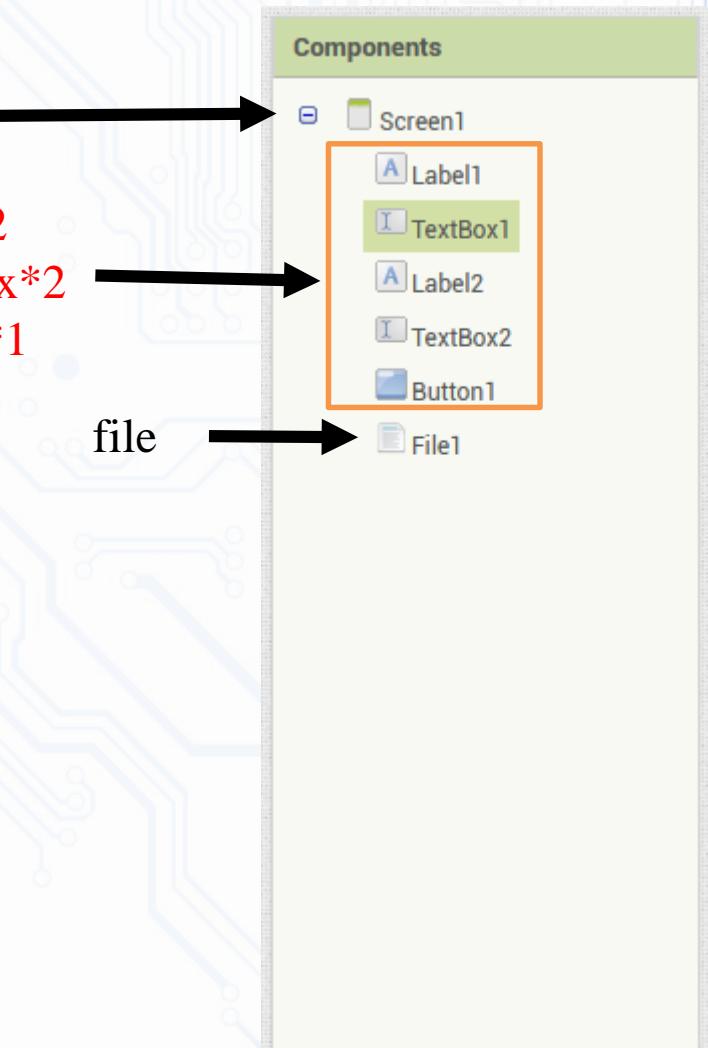
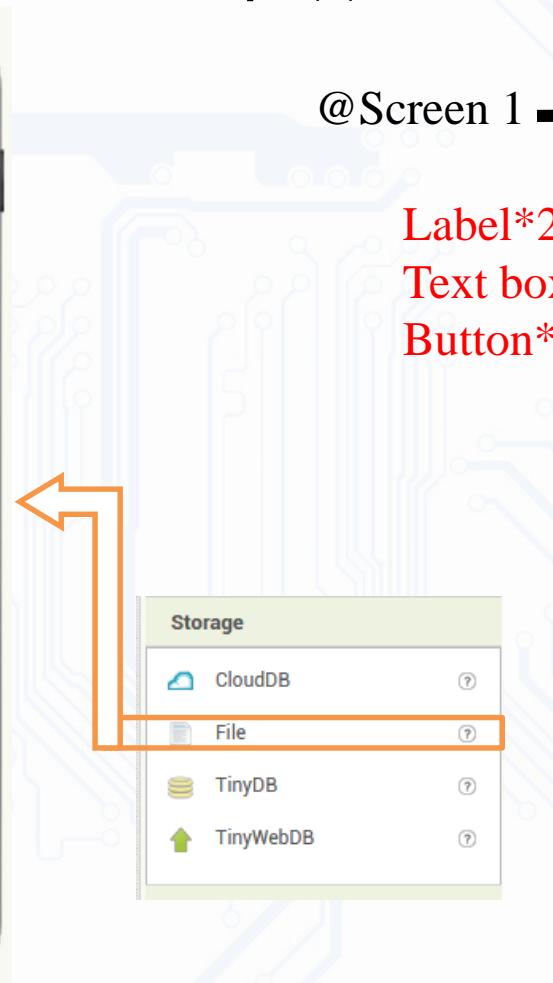
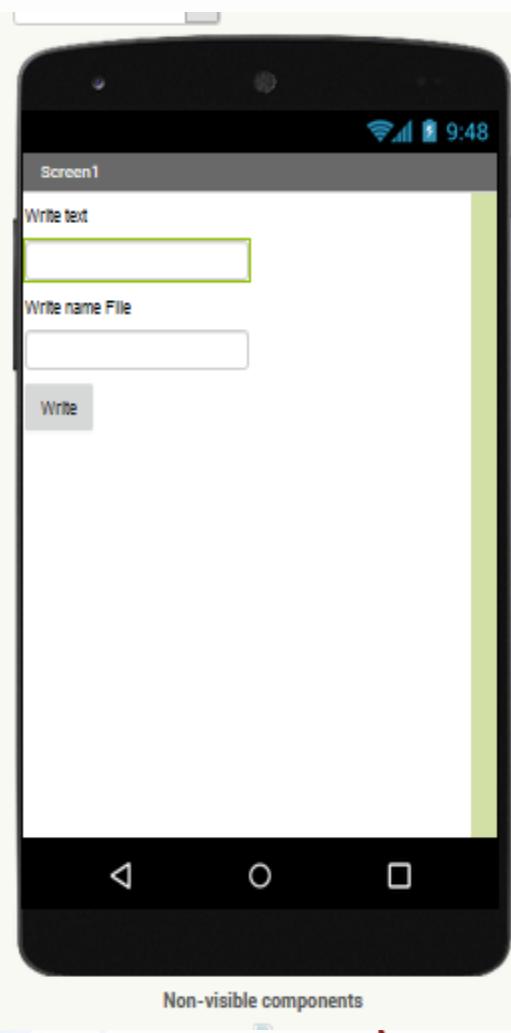


App Inventor



## Example 01:

# Storage Save and upload files File READ/ WRITE





## Example 01:

# Storage Save and upload files File READ/ WRITE



```
initialize global my_text to [red puzzle piece]  
initialize global my_file to [red puzzle piece]  
  
when Button1 .Click  
do  
    set global my_text to TextBox1 . Text  
    set global my_file to TextBox2 . Text  
    call File1 .SaveFile  
        text get global my_text  
        fileName join " / "  
            get global my_file  
            ".txt "
```



# Example 01:

## Storage Save and upload files File READ/ WRITE



MIT App Inventor

www.BANDICAM.com

ai2.appinventor.mit.edu/#4626766707949568

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

Palette

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- Switch
- TextBox
- TimePicker
- WebViewer

viewer

Display hidden components in Viewer

Phone size (505,320)

Screen1

Components

Properties

Screen1

- AboutScreen
- AccentColor Default
- AlignHorizontal Left: 1
- AlignVertical Top: 1
- AppName ATA\_012
- BackgroundColor Default
- BackgroundImage None...
- BlocksToolkit All
- CloseScreenAnimation Default
- Icon None...
- OpenScreenAnimation Default
- PrimaryColor Default
- PrimaryColorDark

Media

Upload File ...

Rename Delete

The screenshot shows the MIT App Inventor development environment. In the center is a smartphone icon representing 'Screen1'. To the left is the 'Palette' with various UI components like Button, CheckBox, and ListView. To the right are the 'Components' and 'Properties' panes for 'Screen1', which currently have no components added. The status bar at the bottom shows system icons and the date/time.

# Example 01:

## Storage Save and upload files File READ/ WRITE



MIT App Inventor x + www.BANDICAM.com

Getting Started PID Basic functions related... C بورش زبان JRM | Fuji Technology ... (36) YouTube

MIT APP INVENTOR Projects Connect Build Settings Help My Projects View Trash Guide Report an Issue English moshaydi@gmail.com

ATA\_012 Screen1 Add Screen ... Remove Screen Publish to Gallery Designer Blocks

Blocks Viewer

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

Screen1 Label1 TextBox1 Label2 TextBox2 Button1 File1

Show Warnings

Download audio from this page ? X

Media

Windows taskbar: Search, File, Microsoft Word, Microsoft Excel, Microsoft Powerpoint, Microsoft Edge, Microsoft Teams, Firefox, Task View, Taskbar icons, 23:49, ENG, 24/11/2020



江西理工大学  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

App Inventor



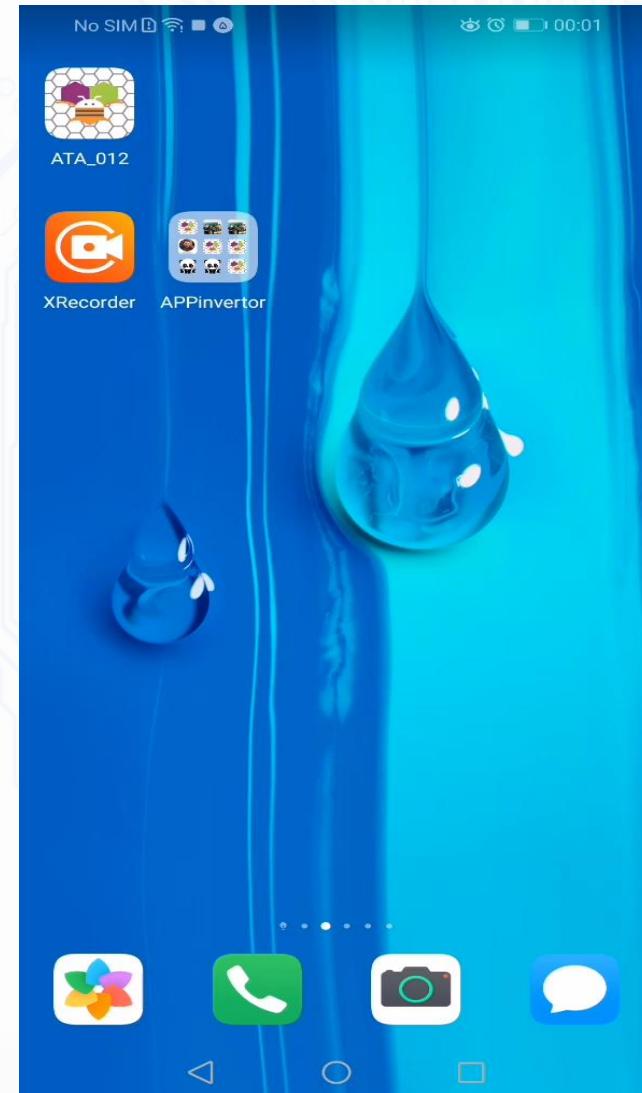
## Example 01:

# Storage Save and upload files File READ/ WRITE



Storage Save and  
upload files File  
WRITE

Demo APP



App Inventor



江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

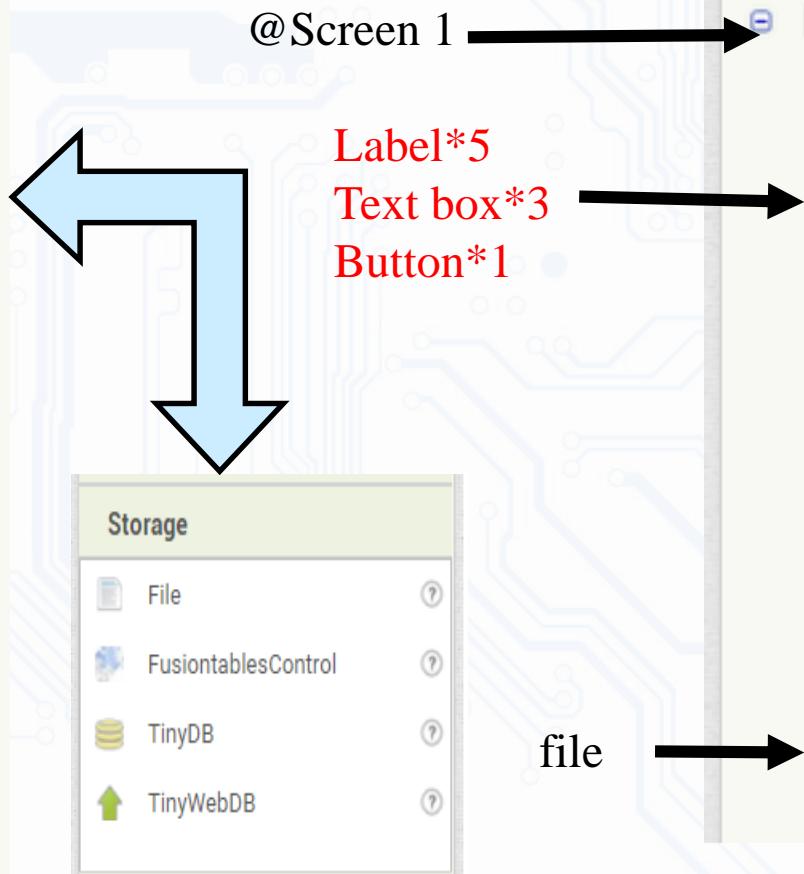
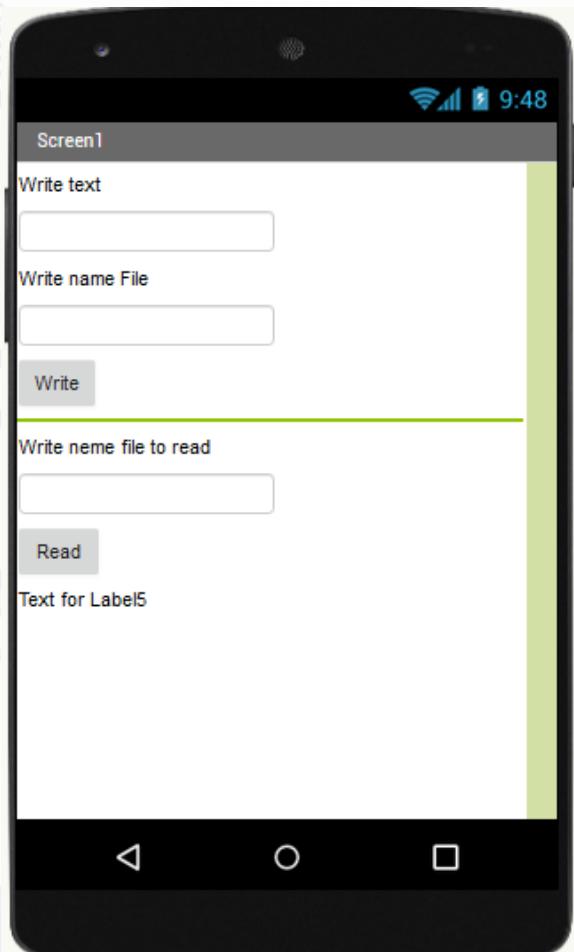


# EXTEND TO READ

We extend the previous project.  
Get the file contents in the Label5.



App Inventor



Components

- Screen1
- Label1
- TextBox1
- Label2
- TextBox2
- Button1
- Label3
- Label4
- TextBox3
- Button2
- Label5
- File1



## Example 01:



# Storage Save and upload files File READ/ WRITE



initialize global [my\_text] to [“”]

initialize global [my\_file] to [“”]

when [Button1].Click

do

- set [global my\_text] to [TextBox1.Text]
- set [global my\_file] to [TextBox2.Text]
- call [File1].SaveFile
  - text: [get [global my\_text]]
  - fileName: [join [“/”] [get [global my\_file]] [“.txt”]]

when [File1].GotText

[text]

do

- set [Label5.Text] to [get [text]]

when [Button2].Click

do

- set [global my\_file] to [TextBox3.Text]
- call [File1].ReadFrom
  - fileName: [join [“/”] [get [global my\_file]] [“.txt”]]



App Inventor



JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



App Inventor

MIT App Inventor    New Tab    www.BANDICAM.com

Getting Started   PID   Basic functions related...   C   بورش زبان   JRM | Fuji Technology ...   YouTube (36)

Palette   Viewer   Components   Properties

Search Components...

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- Switch
- TextBox
- TimePicker
- WebViewer

Display hidden components in Viewer  
Phone size (505,320)

Screen1

Screen1

Write text

Write name File

Write

Rename   Delete

Components

- Screen1
  - Label1
  - TextBox1
  - Label2
  - TextBox2
  - Button1
  - File1

Properties

Screen1

AboutScreen

AccentColor Default

AlignHorizontal Left : 1

AlignVertical Top : 1

AppName ATA\_012

BackgroundColor Default

BackgroundImage None...

BlocksToolkit All

CloseScreenAnimation Default

Icon None...

OpenScreenAnimation Default

Media

Upload File ...



江西理工大学  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

00:09  
25/11/2020

App Inventor



App Inventor

MIT App Inventor    New Tab    www.BANDICAM.com

Getting Started   PID   Basic functions related...   سوچش زبان   JRM | Fuji Technology ...   YouTube

Palette   Viewer   Components   Properties

Search Components...

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- Switch
- TextBox
- TimePicker
- WebViewer

Display hidden components in Viewer   Phone size (505,320)

Screen1

Write text  
[Text Box]

Write name File  
[Text Box]

Write  
[Text Button]

Write neme file to read  
[Text Box]

Read  
[Text Button]

Text for Label5

Components

- Screen1
  - Label1
  - TextBox1
  - Label2
  - TextBox2
  - Button1
  - Label3
  - Label4
  - TextBox3
  - Button2
  - Label5
  - File1

Properties

Label3

BackgroundColor: None

FontBold: False

FontItalic: False

FontSize: 14.0

FontTypeface: default

HTMLFormat: False

HasMargins: True

Height: Automatic...

Width: Fill parent...

Text:

TextAlignment: left : 0

TextColor:

Rename   Delete

Media

Upload File ...



江西理工大学  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

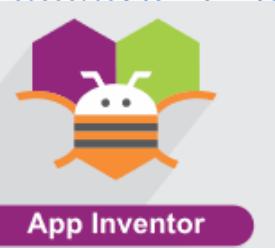
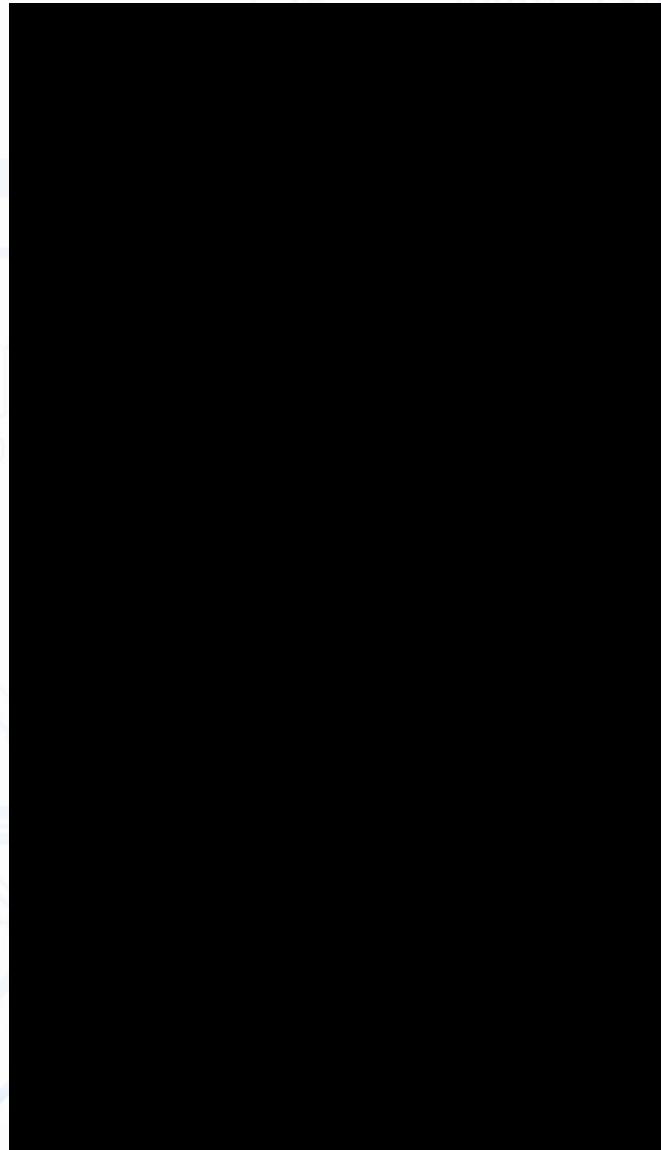
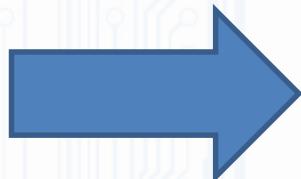
00:12  
25/11/2020  
App Inventor

## Example 01:

# Storage Save and upload files File. READ/ WRITE



DEMO APP

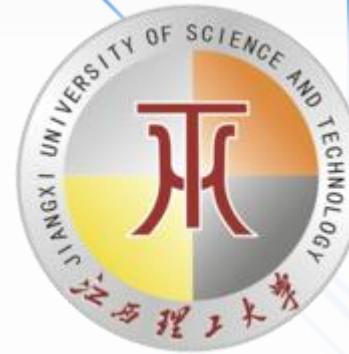


App Inventor



江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT



Example 02:  
**TinyDB (I)**



# Example 02: TinyBD (I)



- **Example Scope:** **TinyDB**, is a database that is in your mobile.
- **Aim:**
  - It is save a person's name and age. We write on **name** of a person, their **age** and click the **Save Button**.
  - Then we get name and age click the **View Button**.
  - The **Delete Button** deletes all data visible on the screen.

TAG Name	Value Age
Name1	Age1
Name2	Age2
Name3	Age3



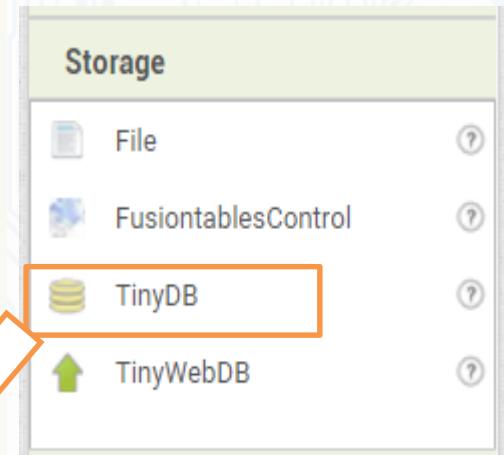
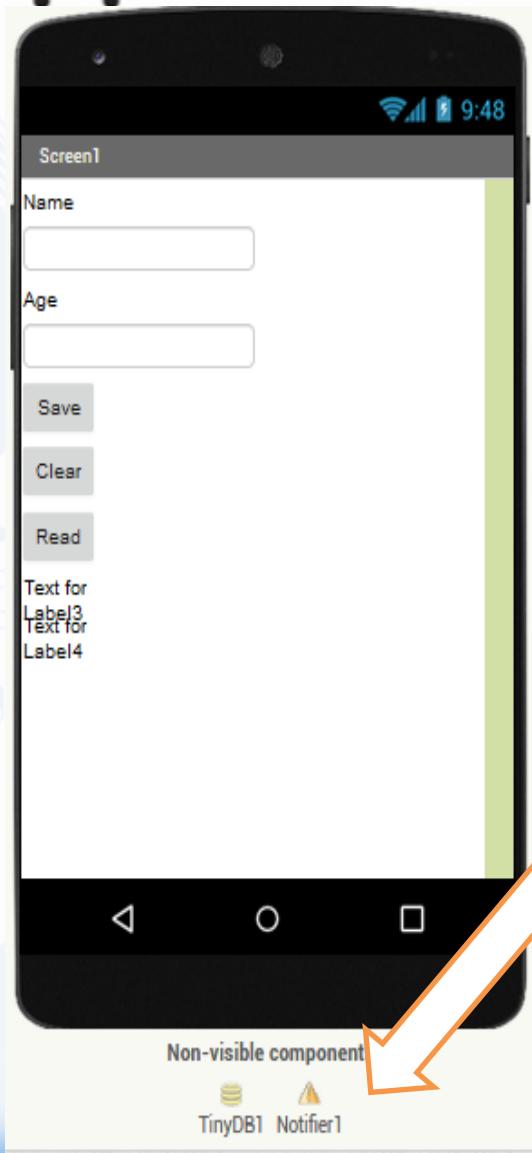
Tag Name	Value Age
Juan	22
Pedro	33
Luis	44



App Inventor



# Example 02: TinyBD (I)



@Screen 1

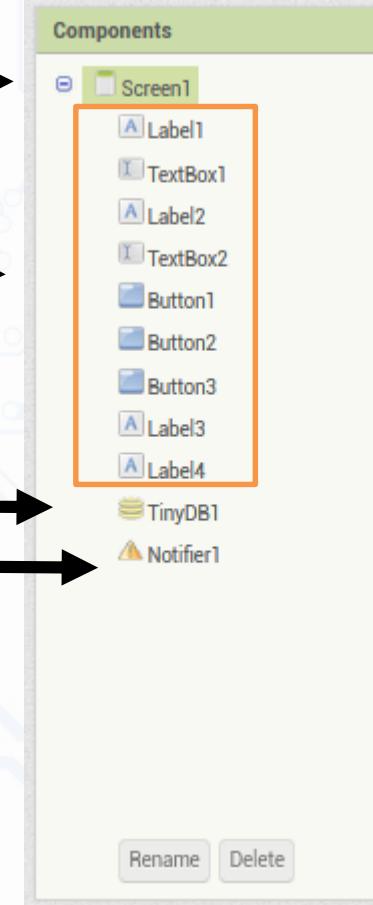
Label\*4

Text box\*2

Button\*3

TinyDB

Notifier



App Inventor



App Inventor



# Example 02: TinyBD (I)



```
when Button1 .Click
do call TinyDB1 .StoreValue
    tag TextBox1 .Text
    valueToStore TextBox2 .Text
call Notifier1 .ShowAlert
    notice " Saved Information "
```

```
when Button2 .Click
do set TextBox1 .Text to " "
    set TextBox2 .Text to " "
    set Label3 .Text to " "
    set Label4 .Text to " "
```

```
when Button3 .Click
do set Label3 .Text to TextBox1 .Text
    set Label4 .Text to call TinyDB1 .GetValue
        tag TextBox1 .Text
        valueIfTagNotThere " NO exist "
```



App Inventor



# Example 02: TinyBD (I)



App Inventor

MIT App Inventor    +    www.BANDICAM.COM

Getting Started   PID   Basic functions related...   بروژ زبان ...   JRM | Fuji Technology ...   (36) YouTube

MIT APP INVENTOR   Projects   Connect   Build   Settings   Help   My Projects   View Trash   Guide   Report an Issue   English   moshaydi@gmail.com

ATA\_014   Screen1   Add Screen ...   Remove Screen   Publish to Gallery   Designer   Blocks

Palette   Search Components...

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- Switch
- TextBox
- TimePicker
- WebViewer

Viewer

Display hidden components in Viewer   Phone size (505,320)

Screen1

Components

Properties

Screen1

AboutScreen

AccentColor Default

AlignHorizontal Left : 1

AlignVertical Top : 1

AppName ATA\_014

BackgroundColor Default

BackgroundImage None...

BlocksToolkit All

CloseScreenAnimation Default

Icon None...

Media

Rename   Delete



App Inventor



# Example 02: TinyBD (I)

MIT App Inventor

www.BANDICAM.com

ai2.appinventor.mit.edu/#6249657044172800

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

90%

Screen1

Name

Age

Save

Clear

Read

Text for Label3

Text for Label4

Non-visible components

TinyDB1 Notifier1

Label2

TextBox2

Button1

Button2

Button3

Label3

Label4

TinyDB1

Notifier1

AccentColor Default

AlignHorizontal Left : 1

AlignVertical Top : 1

AppName ATA\_014

BackgroundColor Default

BackgroundImage None...

BlocksToolkit All

CloseScreenAnimation Default

Icon None...

OpenScreenAnimation Default

PrimaryColor Default

PrimaryColorDark Default

ScreenOrientation Unspecified

Scalable

Rename Delete

Media

Upload File ...





# Example 02: TinyBD (I)



MIT App Inventor x

www.BANDICAM.com

ai2.appinventor.mit.edu/#6249657044172800

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

Projects Connect Build Settings Help My Projects View Trash Guide Report an Issue English moshaydi@gmail.com

MIT APP INVENTOR ATA\_014 Screen1 Add Screen ... Remove Screen Publish to Gallery Designer Blocks

Blocks Viewer

Built-in

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

Screen1

- Label1
- TextBox1
- Label2
- TextBox2
- Button1
- Button2
- Button3

Rename Delete Show Warnings

Media

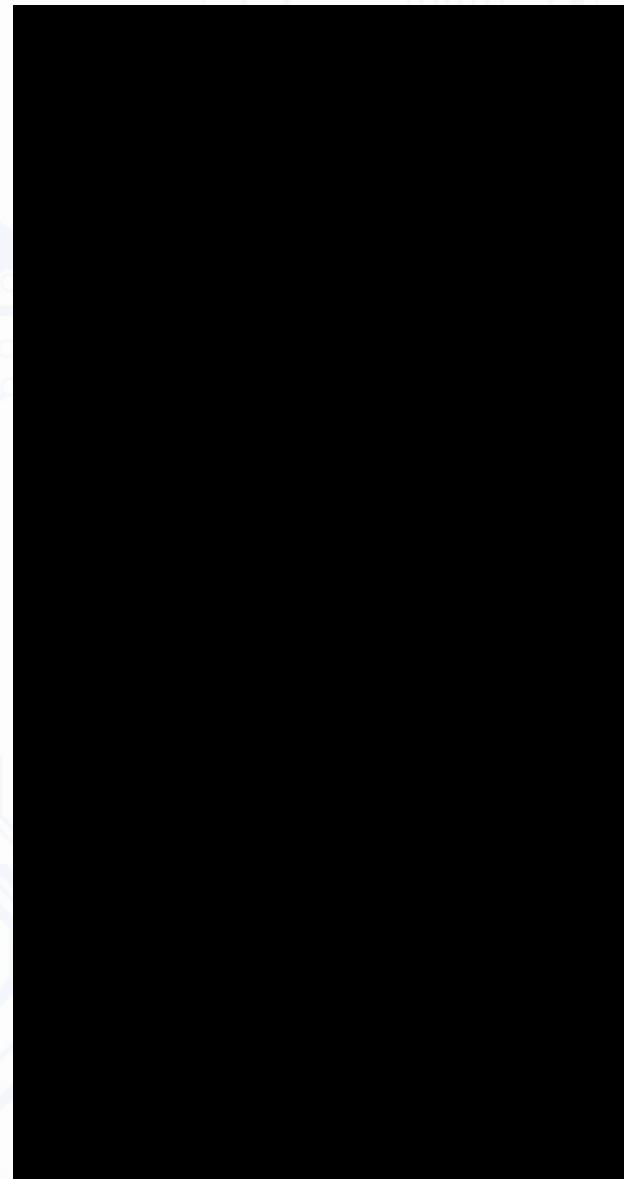
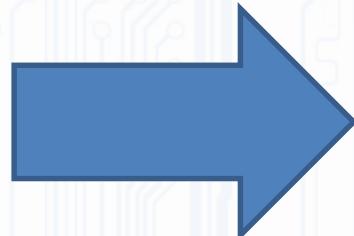
00:32 25/11/2020





# Example 02: TinyBD (I)

APP DEMO



江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



# Student Task\_11



**How java bridge can help us in MIT app programming**

**Why we need the Java bridge?**

**Can we run the MIT app cod in Java editor?**

**Next lecture**

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





# Student Task\_Group



- Each student is responsible to send two task and should not copy from others.
- Copy task from other will cut both student mark.
- Both task has the 25 mark and counted as midterm exam
- The student should send all files with their name and task name
  1. Complete the report in word file
  2. Send the viso file based on completed task (separate file)
  3. Send aia and apk file ( separate file)
  4. Send the process video and working demo (separate file)

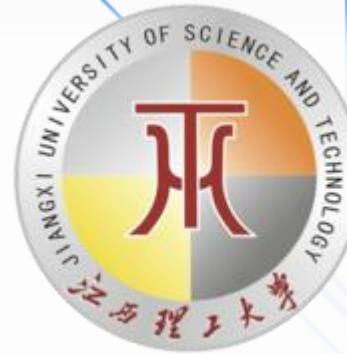
The task I will send for you based on table

Your file should have this format of name

<Task number><student name><Student ID>.ppt

2 week time  
Before  
1 DEC





Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT



## Example 03:

### TinyDB with List (II)

More complicated example of TinyDB



# Example 03:

## TinyDB with List (II)

- In last example We have seen in the above example we can **relate a tag with a value** in a TinyDB, so we keep a Label with their respective value. Sending get the tag name value.
- In this case, each value contains only one element, but let's assume that we want to save the **Name, Age and City** of a person (**Name**) in principle can not do it because each label contains only one value. But through a **List** we can make the value has **several elements**, the elements of the table.



TAG Name	Value Surname, Age, City
Name1	Surname1, Age1, City1
Name2	Surname, Age2, City2
Name3	Surname3, Age3, City3



Tag Name	Value Surname, Age, City
Juan	Perez 22 Cadiz
Pedro	Sanchez 33 Sevilla
Luis	Rodriguez 44 Jerez



App Inventor



江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



# Example 03:

## TinyDB with List (II)



- In this example We created a list called **person**.

Each time you press the **Button1**, delete the **List**. It is an auxiliary List, use it to enter three values in each name. But it is not accumulative List that will keep all the names, so it is deleted upon arrival of each new name.

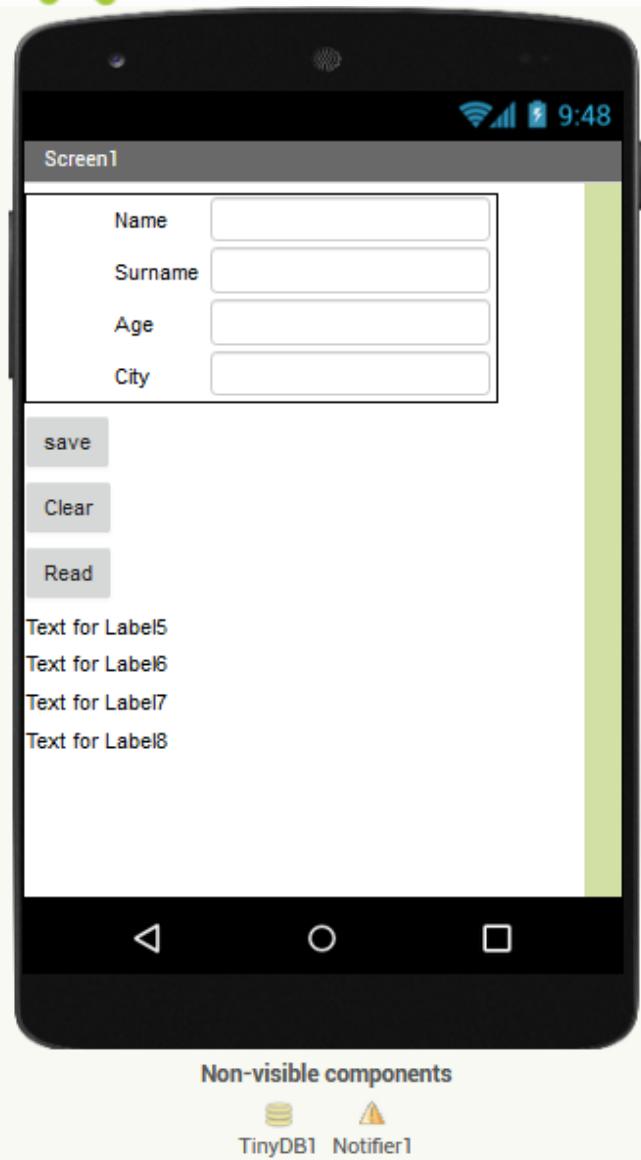
- We enter the data in the **TextBox person**. TinyDB keep on the Label and its corresponding value **that is the person List**.



App Inventor

# Example 03:

## TinyDB with List (II)



@Screen 1

Label\*4  
Text box\*4

Label\*4  
Text box\*4

Label\*4  
Text box\*4

TinyDB  
Notifier



App Inventor

Components

Screen1  
TableArrangement1  
Name  
Textbox  
TextBox2  
TextBox3  
TextBox4  
Surname  
age  
City

Button1  
Button2  
Button3  
Label5  
Label6  
Label7  
Label8

TinyDB1  
Notifier1  
Rename  
Delete

App Inventor



## Example 03:

# TinyDB with List (II)



- Create a table called empty **sites**.
- - When you press the **Button1**, we can switch between GPS coordinates to obtain or enter them manually. Will press to enter them manually.
- - If we want to obtain GPS, GPS is enabled and the longitude and latitude in lockers.
- - Also added to the **List sites**, the name of the sites.
  - Also introduced into the **ListView1** all elements of the places List.
- 



App Inventor



# Example 03:

## TinyDB with List (II)



- When you press the **ListView1**, we will have the elements previously saved **ListView1.items = sites**
- When you click an item in the **ListView1**, it is taken from the **TinyDB** the value of the selected item, this value, in turn, contains two parts, the one that would be the longitude and latitude.
- They are written in their **TextBox** appropriate. That is, **the tag** is the name of the selected item and **value** is the value of that element, which in turn contains two terms, since they were introduced by a list of two parts.
- Again, do not confuse the **sites List**, containing the names of places only and another List that has no name that contains the two coordinates in each of its elements.
- Each time the screen starts, take the tag of TinyDB and introduced into the places list. The list elemenos of places, are introduced into the ListPicker.
- The **Button3** use it to **clean the entire** database.
- The last item will last in the **ListView1**.

location (Tag TinyDB) TextBox1	longitude, latitude is introduced both in a list and stored. TextBox2 and TextBox3
Grazalema	-6.16, 36.21
Jerez	-6.15, 36.24
Puerto Real	-6.16, 36.52



App Inventor



# Example 03:

## TinyDB with List (II)



App Inventor

```
initialize global person to create empty list  
when Button1 .Click  
do set global person to create empty list  
  add items to list list get global person  
    item TextBox2 .Text  
    item TextBox3 .Text  
    item TextBox4 .Text  
call TinyDB1 .StoreValue  
  tag Textbox .Text  
  valueToStore get global person  
call Notifier1 .LogWarning  
  message " Saved information "
```

```
when Button2 .Click  
do set TextBox .Text to *  
set TextBox2 .Text to *  
set TextBox3 .Text to *  
set TextBox4 .Text to *  
set Label5 .Text to *  
set Label6 .Text to *  
set Label7 .Text to *  
set Label8 .Text to *  
  
when Button3 .Click  
do set global person to call TinyDB1 .GetValue  
  tag Textbox .Text  
  valueIfTagNotThere * NO EXIST *  
  set Label5 .Text to Textbox .Text  
  set Label6 .Text to select list item list get global person  
    index 1  
  set Label7 .Text to select list item list get global person  
    index 2  
  set Label8 .Text to select list item list get global person  
    index 3
```



江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



App Inventor



# Example 03:

## TinyDB with List (II)



Screenshot of the MIT App Inventor interface showing the creation of a mobile application.

The browser address bar shows: [ai2.appinventor.mit.edu/#4966462885593088](http://ai2.appinventor.mit.edu/#4966462885593088)

The interface includes:

- Palette:** A sidebar containing categories like User Interface (Button, CheckBox, DatePicker, Image, Label, ListPicker, ListView, Notifier, PasswordTextBox, Slider, Spinner, Switch, TextBox, TimePicker, WebViewer), Layout, Media, Drawing and Animation, Maps, and Sensors.
- Viewer:** Displays a smartphone screen titled "Screen1" with a blank white background.
- Components:** A list of components including "Screen1".
- Properties:** A panel showing properties for "Screen1", such as AccentColor (Default), AlignHorizontal (Left : 1), AlignVertical (Top : 1), AppName (ATA\_015), BackgroundColor (Default), BackgroundImage (None...), BlocksToolkit (All), CloseScreenAnimation (Default), Icon (None...), OpenScreenAnimation (Default), PrimaryColor (Default), PrimaryColorDark (Default), ScreenOrientation (Unspecified), and Scrollable.



App Inventor



# Example 03:

## TinyDB with List (II)



Screenshot of the MIT App Inventor 2 interface showing a project titled "ATA\_015".

The project interface includes:

- Palette:** On the left, categories include Notifier, User Interface (Layout: HorizontalArrangement, HorizontalScrollView, TableArrangement, VerticalArrangement, VerticalScrollView), Media, Drawing and Animation, Maps, Sensors, Social, Storage, Connectivity, LEGO® MINDSTORMS®, and Experimental.
- Viewer:** Shows a smartphone screen for "Screen1" with four text input fields for Name, Surname, Age, and City, and three buttons for save, Clear, and Read. Below the screen are four text labels: Text for Label5, Text for Label6, Text for Label7, and Text for Label8.
- Components:** A tree view of components used in the screen:
  - Screen1
    - TableArrangement1
      - Name
      - TextBox
      - TextBox2
      - TextBox3
      - TextBox4
      - Surname
      - age
      - City
    - Button1
    - Button2
    - Button3
    - Label5
    - Label6
    - Label7
    - Label8
    - TinyDB1
- Properties:** A panel on the right showing properties for the selected component (Button3). Properties include:
  - BackgroundColor: Default
  - Enabled: checked
  - FontBold: unchecked
  - FontItalic: unchecked
  - FontSize: 14.0
  - FontTypeface: default
  - Height: Automatic...
  - Width: Automatic...
  - Image: None...
  - Shape: default
  - ShowFeedback: checked



# Example 03:

## TinyDB with List (II)



Screenshot of the MIT App Inventor 2 environment showing a project titled "ATA\_015".

The project contains the following blocks:

- when Button1 .Click**:
  - do
    - set global person to create empty list
    - add items to list item TextBox2 .Text to get global person
    - item TextBox3 .Text to
    - item TextBox4 .Text to
    - call TinyDB1 .StoreValue tag TextBox .Text valueToStore get global person
    - call Notifier1 .LogWarning message "Saved information"
- when Button2 .Click**:
  - do
    - set TextBox1 .Text to
    - set TextBox2 .Text to
    - set TextBox3 .Text to
    - set TextBox4 .Text to
    - set Label5 .Text to
    - set Label6 .Text to
    - set Label7 .Text to
    - set Label8 .Text to
- when Button3 .Click**:
  - do
    - set global person to call TinyDB1 .GetValue tag valueIfTagNotThere
    - set Label5 .Text to Textbox1 .Text
    - set Label5 .Text to

The interface includes a sidebar with categories like Built-in, Screen1, and TableArrangement1, and a toolbar at the bottom.





# Example 03:

## TinyDB with List (II)



Screenshot of the MIT App Inventor 2 environment showing a project titled "ATA\_015".

The project contains two screens:

- Screen1:** Contains four text boxes (TextBox2, TextBox3, TextBox4) and four labels (Label5, Label6, Label7, Label8). It has the following blocks:
  - when Button1.Click: set global person to create empty list; add items to list list get global person item TextBox2.Text item TextBox3.Text item TextBox4.Text; call TinyDB1.StoreValue tag TextBox1.Text valueToStore get global person; call Notifier1.LogWarning message "Saved information"
  - when Button2.Click: set TextBox1.Text to [ ]; set TextBox2.Text to [ ]; set TextBox3.Text to [ ]; set TextBox4.Text to [ ]; set Label5.Text to [ ]; set Label6.Text to [ ]; set Label7.Text to [ ]; set Label8.Text to [ ]
  - when Button3.Click: set global person to call TinyDB1.GetValue tag TextBox1.Text valueIfTagNotThere "NO EXIST"; set Label6.Text to select list item list get global person index 1; set Label7.Text to select list item list get global person index 2; set Label8.Text to select list item list get global person index 3
- Screen2:** Contains a single text box (TextBox1) and three labels (Label1, Label2, Label3). It has the following blocks:
  - when Button1.Click: set global person to create empty list; add items to list list get global person item TextBox1.Text item TextBox2.Text item TextBox3.Text; call TinyDB1.StoreValue tag TextBox1.Text valueToStore get global person; call Notifier1.LogWarning message "Saved information"
  - when Button2.Click: set TextBox1.Text to [ ]; set TextBox2.Text to [ ]; set TextBox3.Text to [ ]
  - when Button3.Click: set global person to call TinyDB1.GetValue tag TextBox1.Text valueIfTagNotThere "NO EXIST"; set Label1.Text to select list item list get global person index 1; set Label2.Text to select list item list get global person index 2; set Label3.Text to select list item list get global person index 3

The sidebar shows the blocks catalog under "Built-in" and "Screen1". The status bar at the bottom shows system icons and the time "20:39 25/11/2020".



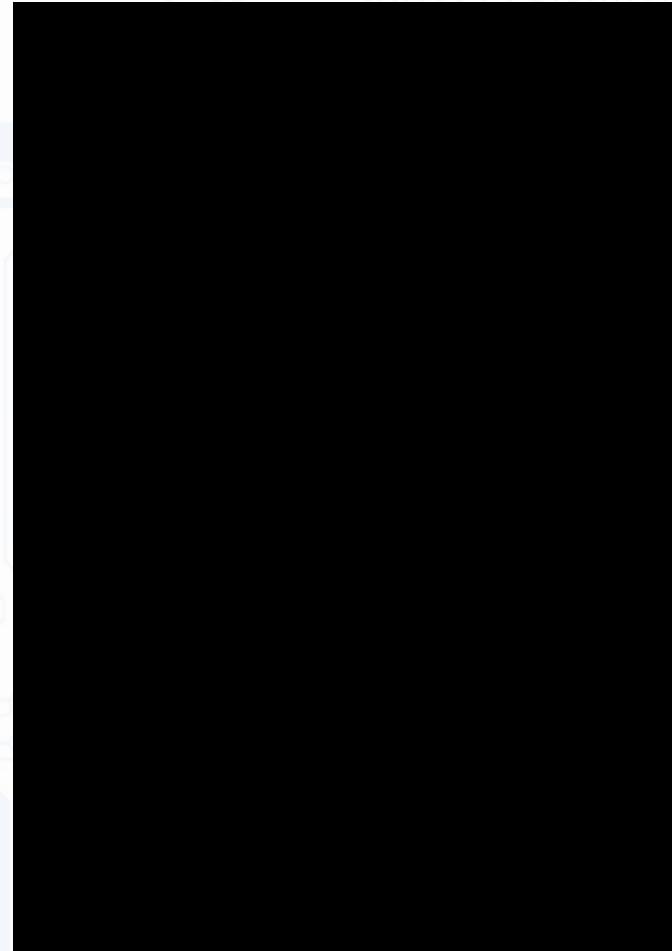
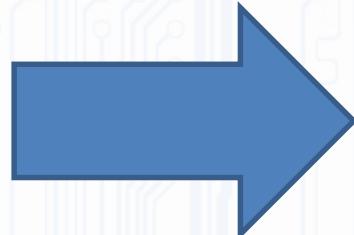
App Inventor



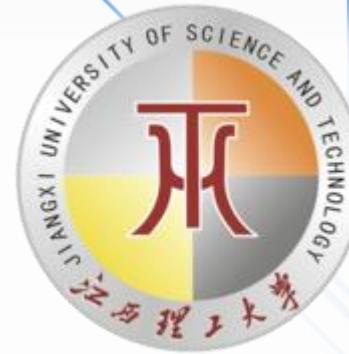
## Example 03: TinyDB with List (II)



# APP DEMO



App Inventor



Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT

**Example 04:**

**TinyDB (III)**

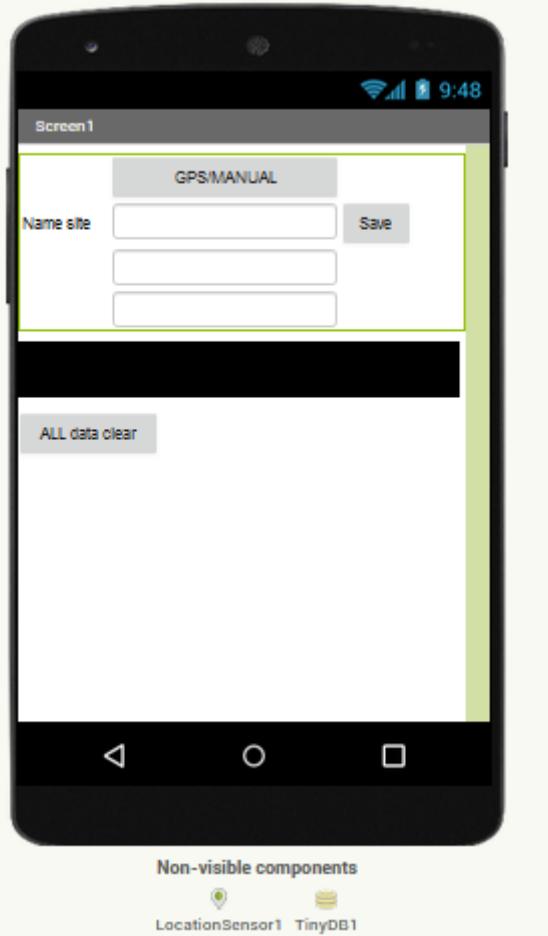
**Example TinyDB more difficult.**



# Example 04:



## TinyDB (III) Example TinyDB more difficult.



@Screen 1

Button \*2  
Label \*1  
Text box\*4

Listview  
Button \*1

Location sensor1  
TinyDB

Components

Screen1  
TableArrangement1

Button1  
Label1  
TextBox1  
TextBox2  
Button2  
TextBox3

ListView1  
Button3  
LocationSensor1  
TinyDB1



App Inventor



江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



# Example 04:

## TinyDB (III)

### Example TinyDB more difficult.

```
when Button1 . Click
do if LocationSensor1 . Enabled .
then set LocationSensor1 . Enabled . to false .
set Button1 . Text . to " Intro datas "
else set LocationSensor1 . Enabled . to true .
set Button1 . Text . to " Datas by GPS "
set TextBox2 . Text . to LocationSensor1 . Longitude .
set TextBox2 . Text . to LocationSensor1 . Latitude .
```

KIO4.COM

```
when LocationSensor1 . LocationChanged
latitude longitude altitude
do set TextBox2 . Text . to get longitude .
set TextBox3 . Text . to get latitude .
```

```
when Button2 . Click
do if is in list? thing TextBox1 . Text .
list get global sites .
then call TinyDB1 . ClearTag
tag TextBox1 . Text .
remove list item list get global sites .
index index in list thing TextBox1 . Text .
list get global sites .
call TinyDB1 . StoreValue
tag TextBox1 . Text .
valueToStore make a list TextBox2 . Text .
TextBox3 . Text .
add items to list list get global sites .
item TextBox1 . Text .
set ListView1 . Elements . to get global sites .
set TextBox1 . Text . to " " .
set TextBox2 . Text . to " " .
set TextBox3 . Text . to " " .
```

```
when ListView1 . AfterPicking
do set TextBox1 . Text . to ListView1 . Selection .
set TextBox2 . Text . to select list item list index
call TinyDB1 . GetValue
tag ListView1 . Selection .
valuelfTagNotThere " Not exist. "
set TextBox3 . Text . to select list item list index
call TinyDB1 . GetValue
tag ListView1 . Selection .
valuelfTagNotThere " Not exist. "
```

```
when Screen1 . Initialize
do set global sites . to call TinyDB1 . GetTags
set ListView1 . Elements . to get global sites .
```



# Example 04:

TinyDB (III) Example TinyDB more difficult.



- When you press the **Button2**, it is checked whether the place name already exists in the list **sites**.  
- If there is, remove it (**ClearTag**) of the Database list TinyDB and **sites**.
- Whether or not, **stores** the writing element in the **TextBos1** on a **tag** of TinyDB and lockers of longitude and latitude, put them in **another list of two elements** and saves as a value in the **TinyDB**.
- That is, keep **the tag** that would be the name of the town and two items using a list, which would be the longitude and latitude. Do not confuse this list of two elements with locations list, they are different.



App Inventor



# Example 04:

## TinyDB (III) Example TinyDB more difficult.



- Also added to the **List sites**, the name of the sites.
- Also introduced into the **ListView1** all elements of the places List.
- When you press the **ListView1**, we will have the elements previously saved  
**ListView1.items** = sites
- When you click an item in the **ListView1**, it is taken from the **TinyDB** the value of the selected item, this value, in turn, contains two parts, the one that would be the longitude and latitude.
- They are written in their **TextBox** appropriate. That is, **the tag** is the name of the selected item and **value** is the value of that element, which in turn contains two terms, since they were introduced by a list of two parts.
- Again, do not confuse the **sites List**, containing the names of places only and another List that has no name that contains the two coordinates in each of its elements.
- Each time the screen starts, take the tag of TinyDB and introduced into the places list. The list elemenos of places, are introduced into the ListPicker.
- The **Button3** use it to **clean the entire** database.
- The last item will last in the ListView1.

Location(Tag TinyDB) Text Box1	Longitude,latitude is introduced both in list and stored Textbox 2 and Textbox3
Name1	Long,lat
Name2	Long,lat
Name3	Long,lat



App Inventor



## Example 04:

# TinyDB (III), Example TinyDB more difficult.



App Inventor

```

initialize global sites to create empty list
when Button1 .Click
do if LocationSensor1 .Enabled then set LocationSensor1 .Enabled to false
set Button1 .Text to "Intro datas"
else set LocationSensor1 .Enabled to true
set Button1 .Text to "Datas by GPS"
set TextBox2 .Text to LocationSensor1 .Longitude
set TextBox2 .Text to LocationSensor1 .Latitude

when ListView1 .AfterPicking
do set TextBox1 .Text to ListView1 .Selection
set TextBox1 .Text to select list item list call TinyDB1 .GetValue
tag ListView1 .Selection
valueIfTagNotThere "No exist"
index 1

when Screen1 .BackPressed
do set global sites to call TinyDB1 .GetTags
set ListView1 .Elements to get global sites

```

```

when LocationSensor1 .LocationChanged
latitude longitude altitude speed
do set TextBox2 .Text to get longitude
set TextBox2 .Text to get altitude

when Button2 .Click
do if is in list? thing TextBox1 .Text
list get global sites
then call TinyDB1 .ClearTag
tag TextBox1 .Text
remove list item list get global sites
index index in list thing TextBox1 .Text
list get global sites

call TinyDB1 .StoreValue
tag TextBox1 .Text
valueToStore make a list TextBox2 .Text
TextBox2 .Text

add items to list list get global sites
item TextBox1 .Text
set ListView1 .Elements to get global sites
set TextBox1 .Text to ""
set TextBox2 .Text to ""

```



江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



App Inventor



## Example 04:

### TinyDB (III), Example TinyDB more difficult.



App Inventor

```
when LocationSensor1 .LocationChanged
  latitude longitude altitude speed
do
  set TextBox2 .Text to get longitude
  set TextBox2 .Text to get altitude
```

```
when Button2 .Click
do
  if is in list? thing TextBox1 .Text
    list get global sites
  then call TinyDB1 .ClearTag
    tag TextBox1 .Text
    remove list item list get global sites
    index index in list thing TextBox1 .Text
    list get global sites
  call TinyDB1 .StoreValue
    tag TextBox1 .Text
    valueToStore make a list TextBox2 .Text
    TextBox2 .Text
  add items to list list get global sites
    item TextBox1 .Text
  set ListView1 .Elements to get global sites
  set TextBox1 .Text to ""
  set TextBox2 .Text to ""
```

```
initialize global sites to create empty list
when Button1 .Click
do
  if LocationSensor1 .Enabled
  then set LocationSensor1 .Enabled to false
    set Button1 .Text to "Intro data"
  else set LocationSensor1 .Enabled to true
    set Button1 .Text to "Datas by GPS"
  set TextBox2 .Text to LocationSensor1 .Longitude
  set TextBox2 .Text to LocationSensor1 .Latitude
when ListView1 .AfterPicking
do
  set TextBox1 .Text to ListView1 .Selection
  set TextBox1 .Text to select list item list call TinyDB1 .GetValue
    tag ListView1 .Selection
    valueIfTagNotThere "No exist"
    index 1
when Screen1 .BackPressed
do
  set global sites to call TinyDB1 .GetTags
  set ListView1 .Elements to get global sites
```



App Inventor



## Example 04:

# TinyDB (III), Example TinyDB more difficult.



App Inventor

App inventor. File. TinyDB. Tiny X MIT App Inventor X www.BANDICAM.com

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

Palette Viewer Components Properties

Search Components...

User Interface

- Button
- CheckBox
- DatePicker
- Image
- Label
- ListPicker
- ListView
- Notifier
- PasswordTextBox
- Slider
- Spinner
- Switch
- TextBox
- TimePicker
- WebViewer

Display hidden components in Viewer  
Phone size (505,320)

Screen1

Screen1

Rename Delete

Media

Upload File ...

AccentColor Default

AlignHorizontal Left : 1

AlignVertical Top : 1

AppName ATA\_016

BackgroundColor Default

BackgroundImage None...

BlocksToolkit All

CloseScreenAnimation Default

Icon None...

OpenScreenAnimation Default

www.BANDICAM.com

21:44 ENG 25/11/2020



## Example 04:

# TinyDB (III), Example TinyDB more difficult.



App Inventor

The screenshot shows the MIT App Inventor 2 interface. At the top, there are two tabs: "App inventor. File. TinyDB. Tiny" and "MIT App Inventor". The URL in the address bar is "ai2.appinventor.mit.edu/#4973125316247552". The main menu includes "Getting Started", "PID", "Basic functions related...", "C بوزش زبان", "JRM | Fuji Technology ...", and "YouTube". The navigation bar has links for "My Projects", "View Trash", "Guide", "Report an Issue", "English", and an email address "moshaydi@gmail.com".

The project title is "ATA\_016". The screen is titled "Screen1". There are buttons for "Add Screen ...", "Remove Screen", and "Publish to Gallery". On the right, there are buttons for "Designer" and "Blocks".

The "Blocks" pane on the left lists categories: Built-in (Control, Logic, Math, Text, Lists, Dictionaries, Colors, Variables, Procedures), Screen1 (Button1), and TableArrangement1 (Button2, Label1, TextBox1, TextBox2, TextBox3). The "Viewer" pane shows a graphical representation of the code. It features two buttons at the bottom labeled "Show Warnings". To the right of the viewer are icons for a trash can, a plus sign, a minus sign, and a circular arrow.

The taskbar at the bottom of the screen includes icons for various applications like File Explorer, Microsoft Word, and Microsoft Excel. The system tray shows the date and time as "25/11/2020 21:51". The status bar at the bottom right also displays the date and time.



## Example 04:

# TinyDB (III), Example TinyDB more difficult.



App Inventor

App inventor. File. TinyDB. Tiny X MIT App Inventor + www.BANDICAM.com

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

Blocks Viewer

```
initialize global sites to create empty list
when Button1 .Click
do
  if LocationSensor1 . Enabled
    then set LocationSensor1 . Enabled to false
    set Button1 . Text to "Intro datas"
  else set LocationSensor1 . Enabled to true
    set Button1 . Text to "Datas by GPS"
    set TextBox2 . Text to LocationSensor1 . Longitude
    set TextBox2 . Text to LocationSensor1 . Latitude

```

Lists Dictionaries Colors Variables Procedures Screen1 Button1 TableArrangement1 Button2 Label TextBox1 TextBox2 TextBox3 ListView1 Button3 TinyDB LocationSensor1 Any component Rename Delete Media Upload File ...

21:56 25/11/2020

This screenshot shows the MIT App Inventor interface for a project titled "App inventor. File. TinyDB. Tiny". The project is running on a browser at "www.BANDICAM.com". The code in the Blocks Editor is designed to handle button clicks. It initializes a global variable "sites" as an empty list. When the "Button1" is clicked, it checks if the "LocationSensor1" is enabled. If it is, it sets the sensor's enable status to false and changes the button's text to "Intro datas". If it's not enabled, it sets the sensor's enable status to true, changes the button's text to "Datas by GPS", and updates two text boxes ("TextBox2") with the current longitude and latitude values from the location sensor.



# Example 04:

## TinyDB (III), Example TinyDB more difficult.



App Inventor

MIT App Inventor    www.BANDICAM.com

ai2.appinventor.mit.edu/#4973125316247552

Blocks    Viewer

Blocks List:

- Built-in
  - Control
  - Logic
  - Math
  - Text
  - Lists
  - Dictionaries
  - Colors
  - Variables
  - Procedures
- Screen1
- TableArrangement1
  - Button2
  - Label1
  - TextBox1
  - TextBox2
  - TextBox3

Viewer Area:

```
set TextBox2 . Text to LocationSensor1 . Longitude
set TextBox2 . Text to LocationSensor1 . Latitude
then
    call TinyDB1 . ClearTag
    tag TextBox1 . Text
    remove list item list get global sites
    index index in list thing TextBox1 . Text
    list get global sites
call TinyDB1 . StoreValue
    tag TextBox1 . Text
    valueToStore make a list TextBox2 . Text
    TextBox3 . Text
    add items to list list get global sites
    item TextBox1 . Text
set ListView1 . Elements to get global sites
set TextBox1 . Text to *
set TextBox2 . Text to *
set TextBox3 . Text to *
```

Warnings:

- ⚠ 0
- ✖ 0

Show Warnings

Media

Upload File ...

Windows Taskbar:

22:03 25/11/2020 ENG

App Inventor

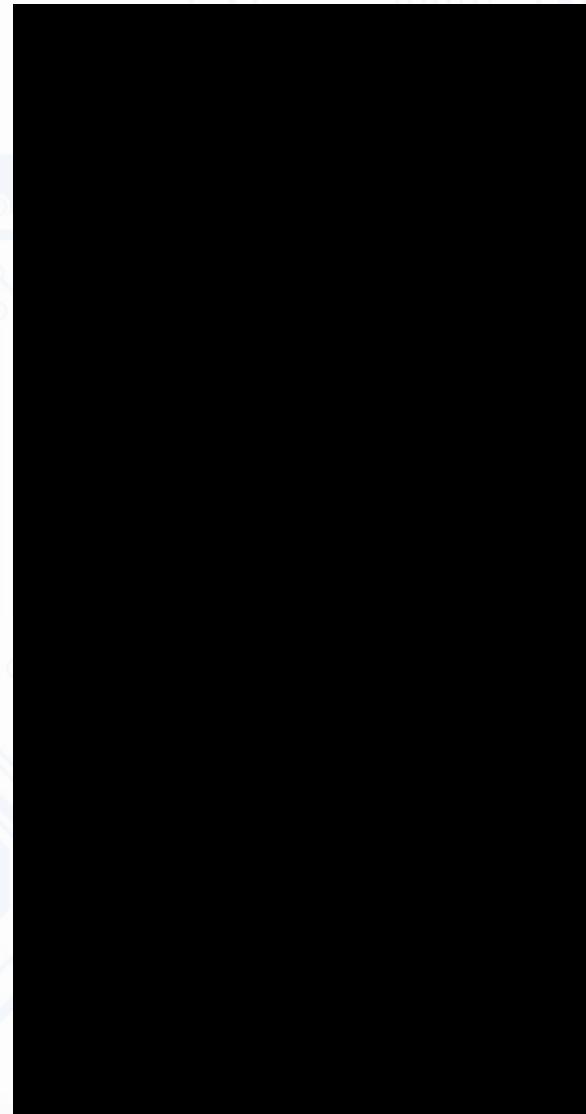
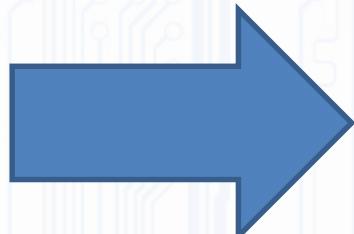


## Example 04:

**TinyDB (III), Example TinyDB more difficult.**



# APP DEMO

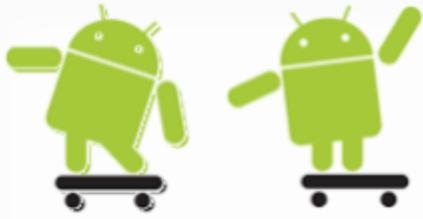


江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



App Inventor



Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT

## LECTURE 016: APP Inventor \_Example

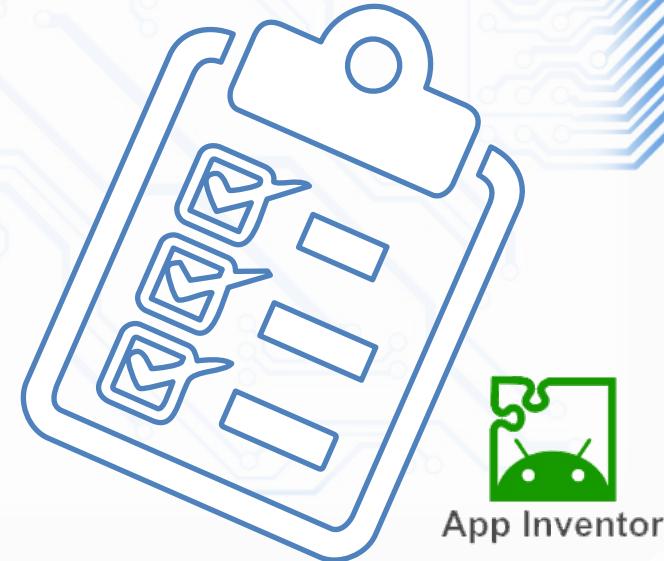
Using WEB TinyDB in App Inventor  
extra mark task

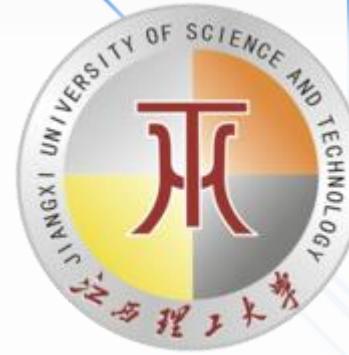


# Agenda

## Using TinyDB in App Inventor

- Example 01: TinyWebDB
- Example 02: MiniTinyDB with several fields.
- Example 03: TinyWebDB data.





Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT

**Example 01: TinyWebDB**



# Example 01: TinyWebDB



- TinyWebDB, is a database that is on the Internet.
- With TinyWebDB we can save the information on the web. App Inventor offers a learning aid MiniTinyDB: <http://appinvtinywebdb.appspot.com/> so that they can rise to 1000 values to your website.  
These values are shared by all users of App Inventor, so that when those values are exhausted 1000 new arriving values are overwritten.
- You can create a custom database by the programmer with its own server, the more complicated as we see in this tutorial ...
- <http://appinventor.mit.edu/explore/ai2/custom-tinywebdb.html>
- It is this simple example we will introduce data states and their capitals.
- We keep it And then typing the name of the State and clicking the View button, we can see its Capital.



# Example 01: TinyWebDB



App Inventor

Storage

- File
- FusiontablesControl
- TinyDB
- TinyWebDB

Components

- Screen1
- Label1
- TextBox1
- Label2
- TextBox2
- Button1
- Button2
- TinyWebDB1
- Notifier1

Properties

TinyWebDB1

ServiceURL

`http://appinvtinywebdb.appspot.com`

KIO4.COM

Non-visible components

TinyWebDB1 Notifier1

Rename Delete

Media

Upload File ...



江西理工大学  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY

APP INVENTOR



# Example 01: TinyWebDB



## Important:

- In the **ServiceURL property** of TinyWebDB, should you set the website where the information is stored, in our example:
- **<http://appinvtinywebdb.appspot.com>**



# Example 01: TinyWebDB



App Inventor

```
when Button1 .Click
do call TinyWebDB1 .StoreValue
    tag TextBox1 .Text
    valueToStore TextBox2 .Text

when TinyWebDB1 .ValueStored
do call Notifier1 .ShowAlert
    notice " Data saved. "

when Button2 .Click
do call TinyWebDB1 .GetValue
    tag TextBox1 .Text

when TinyWebDB1 .GotValue
tagFromWebDB valueFromWebDB
do call Notifier1 .ShowAlert
    notice ⚡ join " Estado: "
        get tagFromWebDB
        " Capital "
        get valueFromWebDB
```



江西理工大学

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



App Inventor

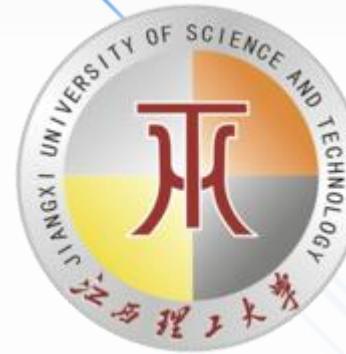


# Example 01: TinyWebDB



- In the database and its corresponding label value, for example is stored:

tag <b>Country</b>	value <b>Capital</b>
<b>Spain</b>	Madrid
<b>Portugal</b>	Lisbon
<b>Italy</b>	Rome



Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT

Example 02:

MiniTinyBD with several fields.





## Example 02:

# MiniTinyDB with several fields.

- In the above case we introduced a **Tag** and its respective **Value**. But in other cases we store a tag containing multiple values, for example:



Tag Name	Value Surname, Age, Country
Juan	Perez 22 Cadiz
Pedro	Sanchez 33 Sevilla
Luis	Rodriguez 44 Jerez



App Inventor



## Example 02:

# MiniTinyDB with several fields.



- To do this we created a **list called persons**. Every time a person is saved, the **List is cleared**, will each element of one of the data **Surname, Age and Country** List.
- And it should be kept the **name as a tag and the person list as the value** of three elements, as indicated by the table I put up.
- It turns out that the **TinyWebDB** does not support saving the list in list format, so we must **pass the registration list to CSV (List to row)** , is something like the separate elements

tag <b>Name</b>	value <b>Surname, Age, Country</b>
Juan	"Perez", "22", "Cadiz"
Pedro	"Sanchez", "33", "Sevilla"
Luis	"Rodriguez", "44", "Jerez"

So yes you can save each record in the **TinyWebDB**. I put **Label5** and **Label6** to visualize how the value data as CSV Record List and as seen.  
We recover data when we do the opposite process, ie, the value data come in **CSV format and registration must convert format list (List from CSV row)** to insert them in persons list.  
Once converted List, we get each element by its index, knowing that the first element is 1.





# Example 02: MiniTinyDB with several fields.

The screenshot shows the KIO4.COM application interface. At the top, there are two checkboxes: "Display hidden components in Viewer" and "Check to see Preview on Tablet size". Below this is a header bar with signal strength, battery, and time (9:48). The main area contains a form with four text input fields labeled "Name", "Surname", "Age", and "City". Below the form are three buttons: "Save", "Text to Label5", and "Text to Label6". A "Read" button is also present. At the bottom, there is a footer with the text "KIO4.COM" and standard Android navigation icons (back, home, recent apps).

**Components**

- Screen1
  - Label1
  - TextBox1
  - Label2
  - TextBox2
  - Label3
  - TextBox3
  - Label4
  - TextBox4
  - Button1
  - Label5
  - Label6
  - Button2
- TinyWebDB1
- Notifier1

**Properties**

TinyWebDB1

ServiceURL  
<http://appinvtinywebdb.appspot.com>

**Media**

Upload File ...

**Non-visible components**

- TinyWebDB1
- Notifier1



**Important:**  
In the **ServiceURL** property of TinyWebDB1, should you set the website where the information is stored, in our example:  
**<http://appinvtinywebdb.appspot.com>**



App Inventor



江西理工大学  
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY



# Example 02:

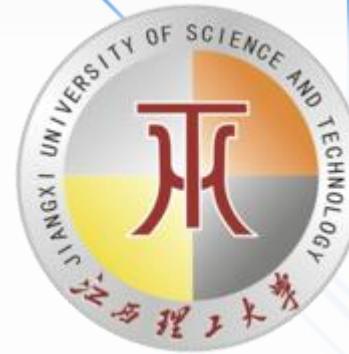
## MiniTinyDB with several fields.



```
initialize global persons to create empty list
initialize global register_csv to " "
when Botón1 Click
do set global persons to create empty list
  add items to list list get global persons
    item CampoDeTexto2 . Text
    item CampoDeTexto3 . Text
    item CampoDeTexto4 . Text
  set Etiqueta5 . Text to get global persons
  set global register_csv to list to csv row list get global persons
  set Etiqueta6 . Text to get global register_csv
  call TinyWebDB1 . StoreValue
    tag CampoDeTexto1 . Text
    valueToStore get global register_csv
when TinyWebDB1 ValueStored
do call Notifier1 . ShowAlert
  notice "Datas saved."
```

```
when Botón2 Click
do call TinyWebDB1 . GetValue
  tag CampoDeTexto1 . Text
when TinyWebDB1 GotValue
tagFromWebDB valueFromWebDB
do set global persons to list from csv row text get valueFromWebDB
  set CampoDeTexto1 . Text to get tagFromWebDB
  set CampoDeTexto2 . Text to select list item list get global persons
    index 1
  set CampoDeTexto3 . Text to select list item list get global persons
    index 2
  set CampoDeTexto4 . Text to select list item list get global persons
    index 3
```





Jiangxi University of Science and Technology



# MOBILE APPLICATION DEVELOPMENT

Example 03:

TinyWebDB data.





# Example 03: TinyWebDB data.

- Another way to store data using several different Labels.

The screenshot shows the App Inventor interface with the following components:

- Components Panel:** Shows the hierarchical structure of the app's components. The root component is **Screen1**, which contains three **HorizontalArrangement** components. Each arrangement has a **Label** and a **TextBox**. Additionally, there are three more labels (**Label2**, **Label3**, **Label4**) and three buttons (**btn\_Save**, **btn\_Read**, **Label5**, **Label6**). A **TinyWebDB1** component is also listed under **Screen1**.
- Properties Panel:** Shows the properties for the **TinyWebDB1** component, including the **ServiceURL** set to <http://appinvtinywebdb.appspot.com>.
- Viewer Panel:** Displays the mobile application interface. It features three text input fields labeled **Name:**, **City:**, and **Age:**. Below these are two buttons: **Guardar** and **Read**. To the right of the buttons are three labels: **Text to Label4**, **Text to Label5**, and **Text to Label6**. At the bottom of the screen, there is a watermark for **KIO4.COM**.



## Example 03:

# TinyWebDB data.



initialize global name to " "

initialize global city to " "

when btn\_Save .Click

do set global name to TextBox1 .Text  
set global city to TextBox2 .Text  
set global age to TextBox3 .Text

call TinyWebDB1 .StoreValue  
tag " Name "  
valueToStore get global name  
call TinyWebDB1 .StoreValue  
tag " City "  
valueToStore get global city  
call TinyWebDB1 .StoreValue  
tag " Age "  
valueToStore get global age

initialize global age to " "

App Inventor

when btn\_Read .Click

do set Label4 .Text to " "  
set Label5 .Text to " "  
set Label6 .Text to " "  
call TinyWebDB1 .GetValue  
tag " Name "  
call TinyWebDB1 .GetValue  
tag " City "  
call TinyWebDB1 .GetValue  
tag " Age "

when TinyWebDB1 .GetValue

tagFromWebDB valueFromWebDB  
do if get tagFromWebDB = " Nombre "  
then set Label4 .Text to get valueFromWebDB  
if get tagFromWebDB = " Ciudad "  
then set Label5 .Text to get valueFromWebDB  
if get tagFromWebDB = " Edad "  
then set Label6 .Text to get valueFromWebDB



# Student Task\_12



**Repeat this examples and make based on our task format**

- We need the video of processing for example 00 and GPS ( full format) rest of example we need demo**
- Add this part to example GPS( example 04):**
  - Solve the location problem**
  - After push the clear remove all data and make data base empty**

**The last 3 example of Tiny DB is optional but you can get more mark by sending the them**

**Next lecture**

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



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.”**





# Reference

- **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/>
- **CloudDB Chat App:** <http://appinventor.mit.edu/explore/ai2/clouddb-chat>
- <https://community.appinventor.mit.edu/t/working-with-lists-and-tinydb/11568>
- <https://learn2c.org/2014/08/26/using-tinydb-in-app-inventor/>
- [http://kio4.com/appinventori/8file\\_database.htm](http://kio4.com/appinventori/8file_database.htm)



# Student Task\_Group



- Each student is responsible to send two task and should not copy from others.
- Copy task from other will cut both student mark.
- Both task has the 25 mark and counted as midterm exam
- The student should send all files with their name and task name
  1. Complete the report in word file
  2. Send the viso file based on completed task (separate file)
  3. Send aia and apk file ( separate file)
  4. Send the process video and working demo (separate file)

The task I will send for you based on table

Your file should have this format of name

<Task number><student name><Student ID>.ppt

**2 week time  
10 DEC**



JAMES PEEBLES  
Nobel Prize in Physics 2019

“Students learn from  
teachers and teachers  
learn from students.”

