江西理工大學信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

Mobile application development

移动应用开发

Lecture 015:

APP Inventor_enviroment

Storage and Connectivity

Dr Ata Jahangir Moshayedi

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

EMAIL: ajm@jxust.edu.cn

Autumn _2021

MOBILE APPLICATION DEVELOPMENT

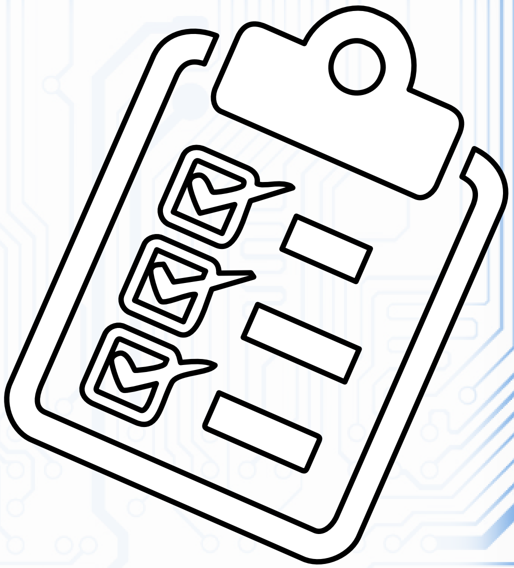
LECTURE 015: APP Inventor_enviroment

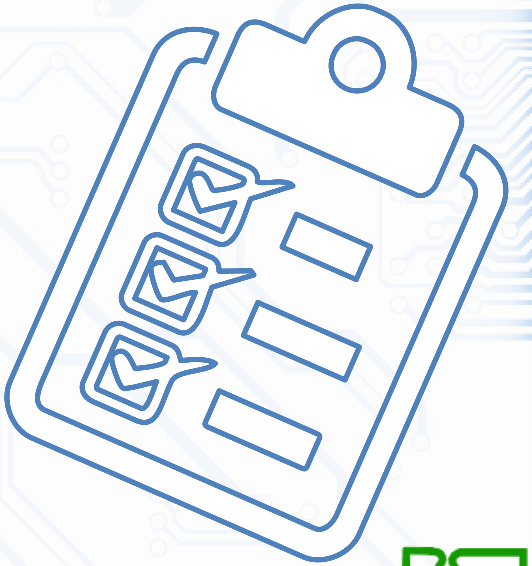
Storage and Connectivity

App Inventor

Agenda

- – **Storage and Connectivity**

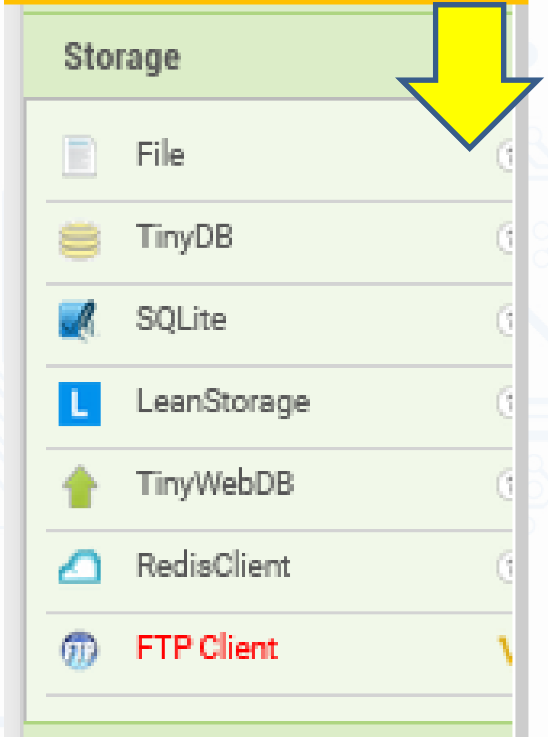




Storage

- Table of Contents:
  - **CloudDB**
  - **File**
  - **TinyDB**
  - **TinyWebDB**

App Inventor 2
WxBit 汉化增强版



Storage

File

TinyDB

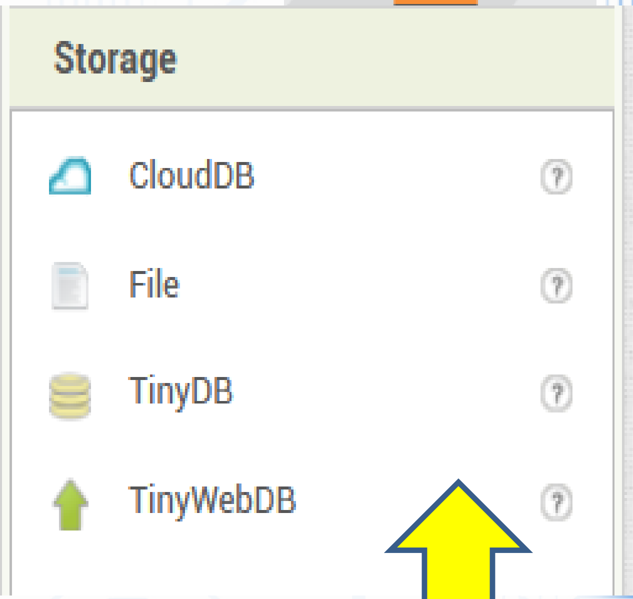
SQLite

LeanStorage

TinyWebDB

RedisClient

FTP Client



Storage

CloudDB

File

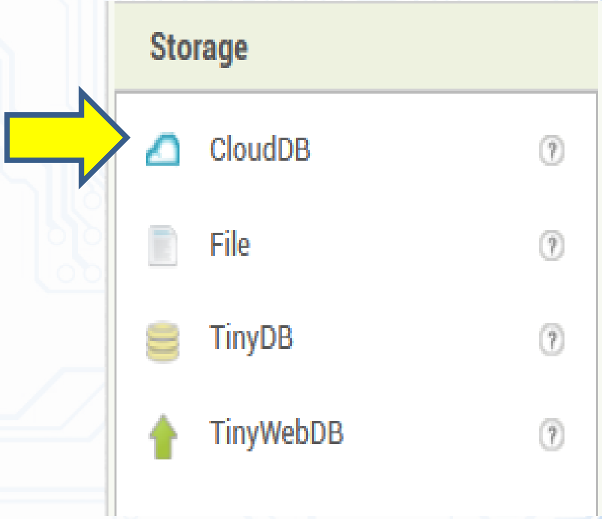
TinyDB

TinyWebDB

App Inventor 2
http://ai2.appinventor.mit.edu

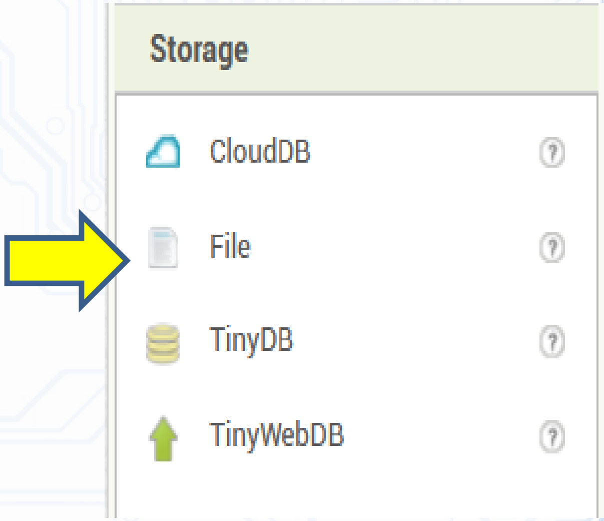
Storage :CloudDB

- The CloudDB component is a Non-visible component that allows you to store data on a Internet connected database server (using Redis software).
- This allows the users of your App to share data with each other.
- By default data will be stored in a server maintained by MIT, however you can setup and run your own server. Set the RedisServer property and RedisPort property to access your own server.



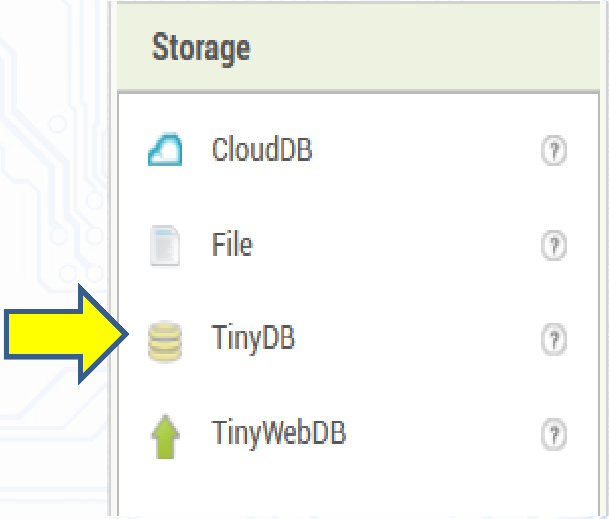
Storage :File

- Non-visible component for storing and retrieving files. Use this component to write or read files on the device.
- The default behavior is to write files to the private data directory associated with the app.
- The Companion writes files to /sdcard/AppInventor/data for easy debugging. If the file path starts with a slash (/), then the file is created relative to /sdcard. For example, writing a file to /myFile.txt will write the file in /sdcard/myFile.txt.



Storage :TinyDB

- TinyDB is a non-visible component that stores data for an app.
- Apps created with App Inventor are initialized each time they run.
- This means that if an app sets the value of a variable and the user then quits the app, the value of that variable will not be remembered the next time the app is run.
- In contrast, TinyDB is a persistent data store for the app. The data stored in a TinyDB will be available each time the app is run.
- **An example might be a game that saves the high score and retrieves it each time the game is played.**



Storage :TinyDB

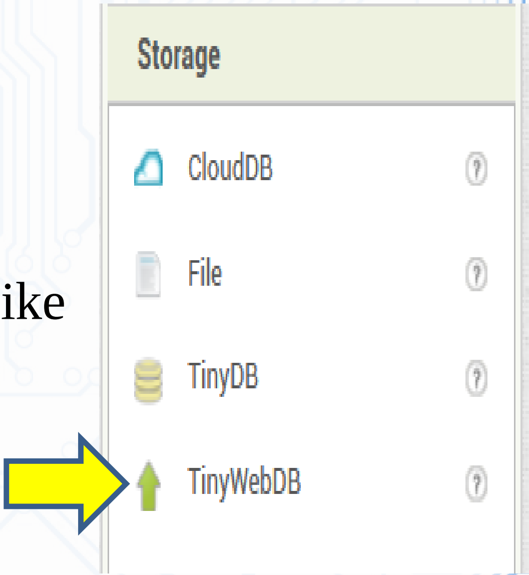
- Data items consist of tags and values. To store a data item, you specify the tag it should be stored under.
- The tag must be a text block, giving the data a name. Subsequently, you can retrieve the data that was stored under a given tag.
- You cannot use the TinyDB to pass data between two different apps on the phone, although you can use the TinyDB to share data between the different screens of a multi-screen app.
- When you are developing apps using the AI Companion, all the apps using that Companion will share the same TinyDB. That sharing will disappear once the apps are packaged and installed on the phone.
- During development you should be careful to clear the Companion app’s data each time you start working on a new app.

Storage :TinyWebDB

- The TinyWebDB component communicates with a Web service to store and retrieve information.
- Although this component is usable, it is very limited and meant primarily as a demonstration for people who would like to create their own components that talk to the Web.
- The accompanying Web service is at

  (http://tinywebdb.appinventor.mit.edu).

- The component has methods to store a value under a tag and to retrieve the value associated with the tag.
- The interpretation of what “store” and “retrieve” means is up to the Web service.
- In this implementation, all tags and values are strings (text). This restriction may be relaxed in future versions.



What is TinyDB?

- TinyDB is a simple “database” that stores data on your phone or tablet.
- Unlike program variables that go away when your app is finished running or your phone is re-set, values stored in TinyDB remain on your phone for use the next time your app is run.

TINY DB

About Memory on your Phone or Tablet

***Your smart phone or tablet typically has two primary types of memory:* RAM and FLASH memory.**

- RAM stands for “random access memory” – but today we mostly think of RAM as memory that can be accessed very fast (as compared to Flash or hard drive memory storage).

- RAM retains values as long as power is applied to the RAM circuity. Once we turn off the power, the values stored in RAM are lost. (In some applications, extra batteries are used to continuously provide power to RAM even when the “normal” power is turned off.)

- Flash memory retains values when the power is turned off. But access to Flash RAM is not as fast as access to conventional RAM memory.

Why is it called "Flash"?

- There was an early version of memory technology where the memory was erased by literally flashing it with ultraviolet light.
- However the inventor of Flash RAM chose the name "Flash" for different reasons. Modern Flash RAM is read, written and erased electronically.
- App Inventor variables are stored in RAM memory – and the content of RAM is erased or reset whenever the power is turned off. TinyDB, on the other hand, stores values in FLASH RAM, where the values remain even when the power is turned off.

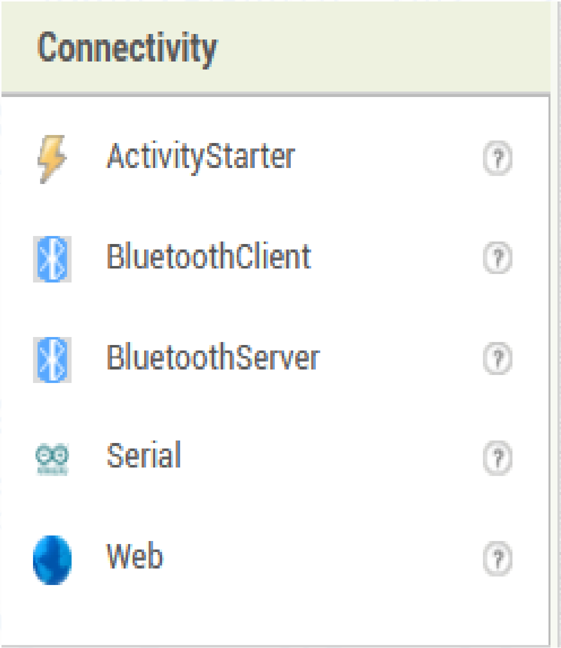
Using TinyDB

- TinyDB provides a simple way to store and retrieve data efficiently and to store the data in long-term storage.
- TinyDB is based on the concept of a “tag” to identify the stored data, and the data value.
- Think of a “tag” as like using your name as your identification to look up your address:
  - Tag value: Martin
  - Value: 123 Main St, Anytown, CHNA
  - or
  - Tag value: Alexa
  - Value: 321 Other St, Someplace, CHAINA

- TinyDB uses the “tag” (such as Alexa) to quickly locate the corresponding value.
- Even if you have 100 names and addresses stored in TinyDB, TinyDB can look up the “tag” quickly and use the tag to find the corresponding value.

Connectivity

- ActivityStarter
- BluetoothClient
- BluetoothServer
- Serial
- Web

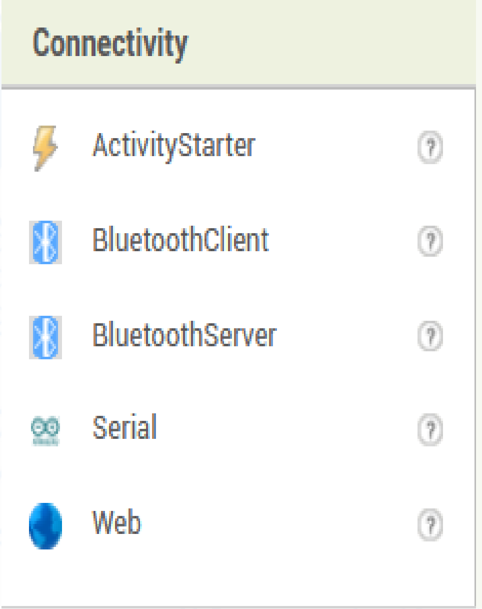


Connectivity

- ActivityStarter
- BluetoothClient
- BluetoothServer
- Serial
- Web

Connectivity : ActivityStarter

- A component that can launch an activity using the StartActivity method.
- Activities that can be launched include:
  - **Starting another App Inventor for Android app**.
  - To do so, first find out the class of the other application by downloading the source code and using a file explorer or unzip utility to find a file named “youngandroidproject/project.properties”.
  - The first line of the file will start with “main=” and be followed by the class name;
  - **for example, main=com.gmail.Bitdiddle.**
  - **Ben.HelloPurr.Screen1. (The first components indicate that it was created by Ben.Bitdiddle@gmail.com.)**

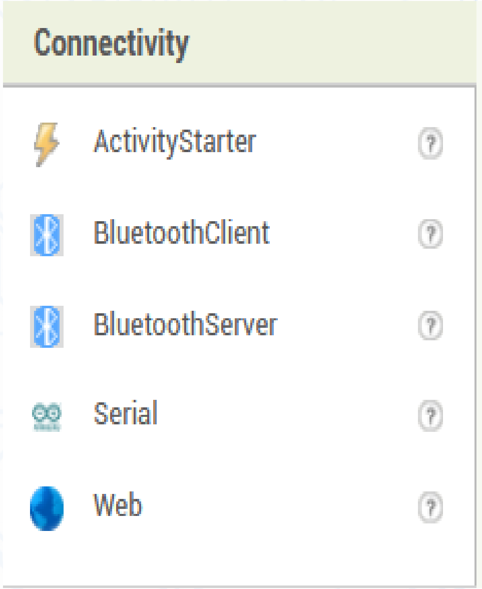


Connectivity :ActivityStarter

To make your ActivityStarter launch this application,

set the following properties:

- ActivityPackage to the class name, dropping the last component
- (for example, com.gmail.Bitdiddle.Ben.HelloPurr)
- ActivityClass to the entire class name (for example, com.gmail.Bitdiddle.Ben.HelloPurr.Screen1)



Starting the camera application by setting the following properties:

Action: android.intent.action.MAIN

ActivityPackage: com.android.camera

ActivityClass: com.android.camera.Camera

Connectivity :ActivityStarter

- Performing web search. Assuming the term you want to search for is “vampire” (feel free to substitute your own choice), set the properties to:

  Action: android.intent.action.WEB_SEARCH

  ExtraKey: query

  ExtraValue: vampire

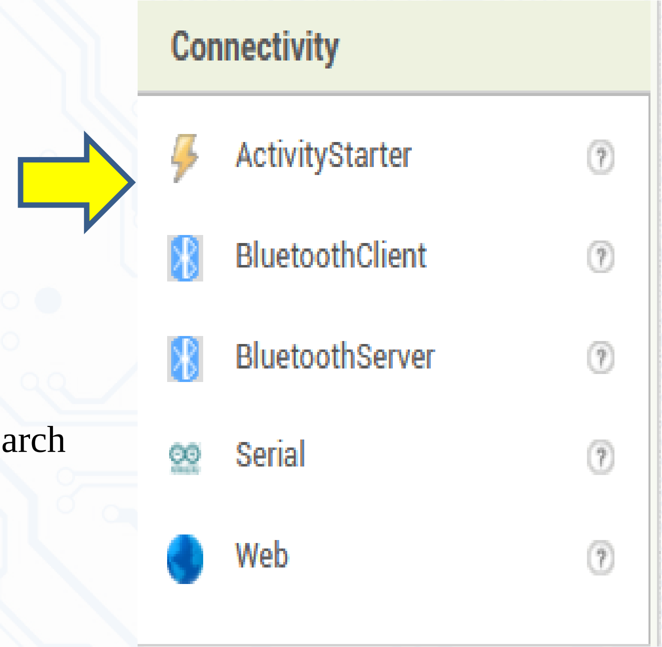
  ActivityPackage: com.google.android.providers.enhancedgooglesearch

  ActivityClass: com.google.android.providers.enhancedgooglesearch.Launcher

  Opening a browser to a specified web page. Assuming the page you want to go to is “www.facebook.com” (feel free to substitute your own choice), set the properties to:

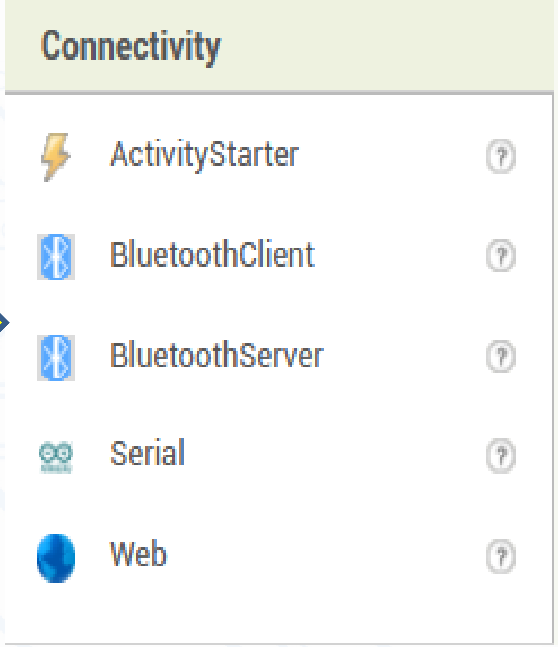
  Action: android.intent.action.VIEW

  DataUri: http://www.facebook.com



Connectivity :Bluetooth

- **BluetoothClient**
  - Use BluetoothClient to connect your device to other devices using Bluetooth.
  - This component uses the Serial Port Profile (SPP) for communication.
  - If you are interested in using Bluetooth low energy, please see the BluetoothLE extension.
- **BluetoothServer**
  - Use the BluetoothServer component to turn your device into a server that receive connections from other apps that use the BluetoothClient component.

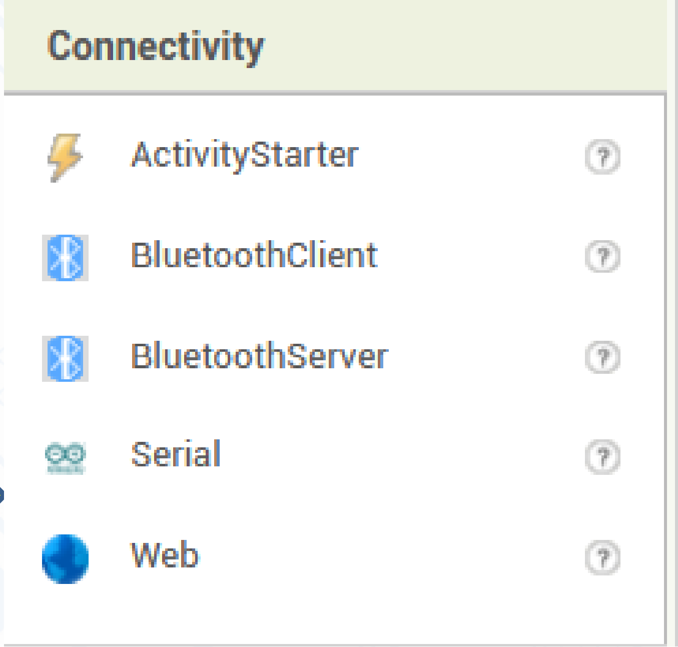


Connectivity

- ActivityStarter ?
- BluetoothClient ?
- BluetoothServer ?
- Serial ?
- Web ?

Connectivity

- **Serial**
  - Component for Serial
- **Web**
  - Non-visible component that provides functions for HTTP GET, POST, PUT, and DELETE requests.



Connectivity

ActivityStarter

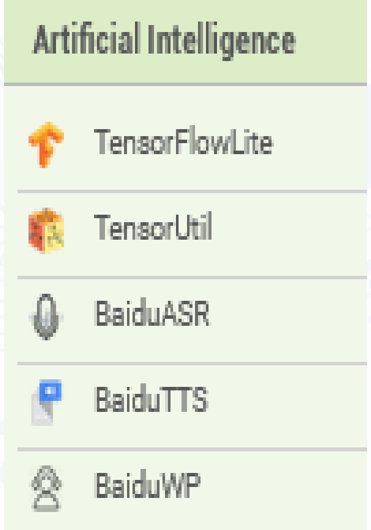
BluetoothClient

BluetoothServer

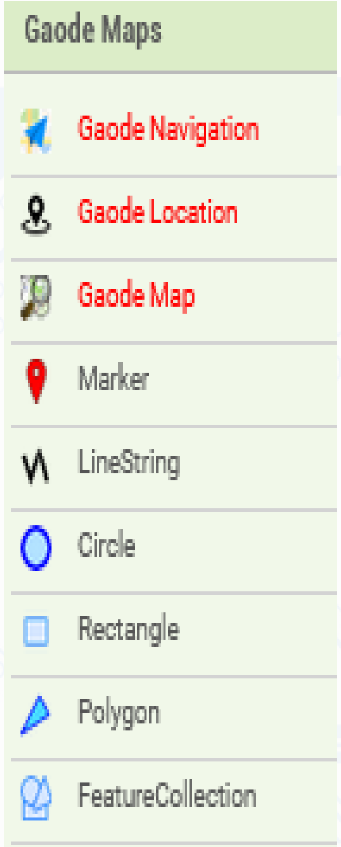
Serial

Web

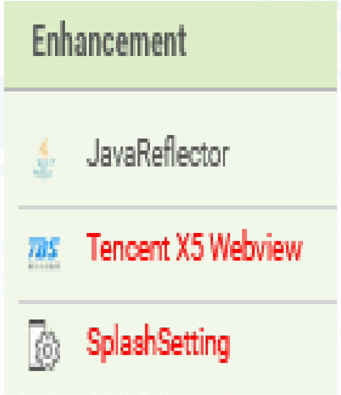
Other APP inventor part


Artificial Intelligence

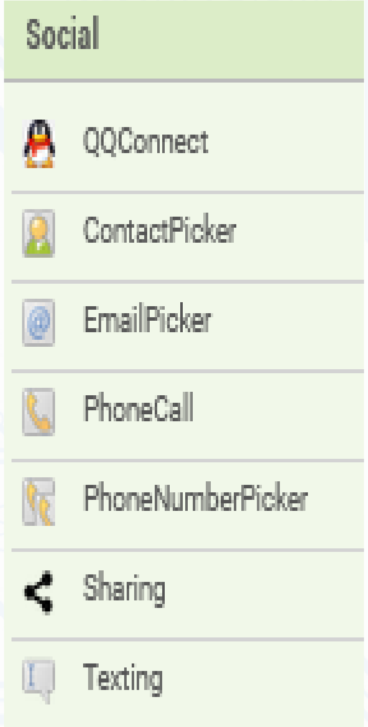
- TensorFlowLite
- TensorUtil
- BaiduASR
- BaiduTTS
- BaiduWP


Gaode Maps

- Gaode Navigation
- Gaode Location
- Gaode Map
- Marker
- LineString
- Circle
- Rectangle
- Polygon
- FeatureCollection

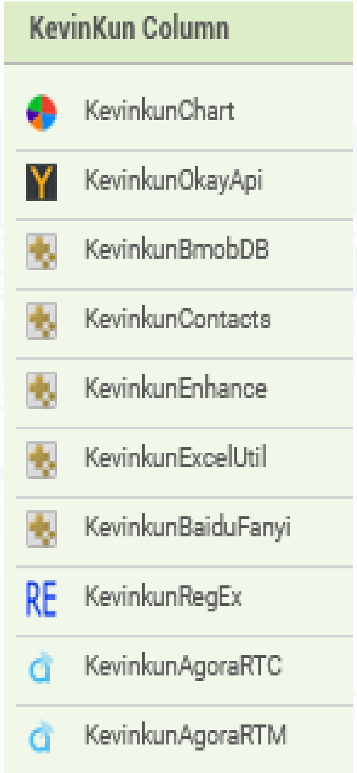

Enhancement

- JavaReflector
- Tencent X5 Webview
- SplashSetting

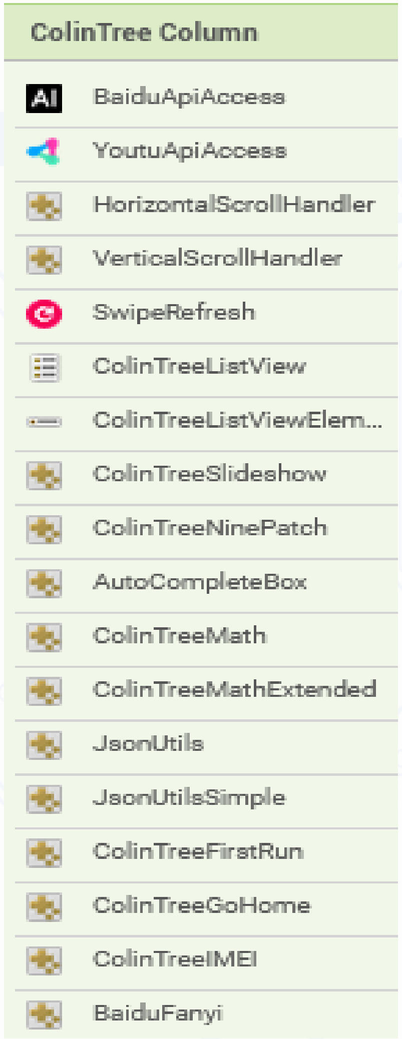

Social

- QQConnect
- ContactPicker
- EmailPicker
- PhoneCall
- PhoneNumberPicker
- Sharing
- Texting

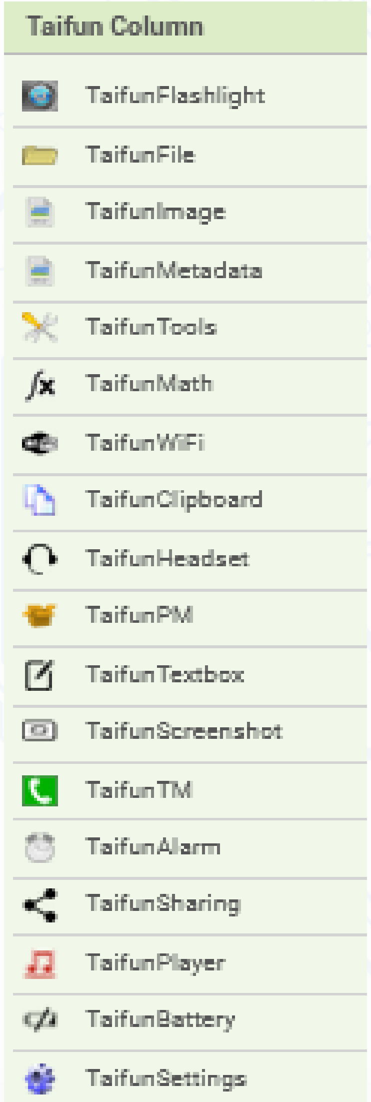
Other APP inventor part


KevinKun Column

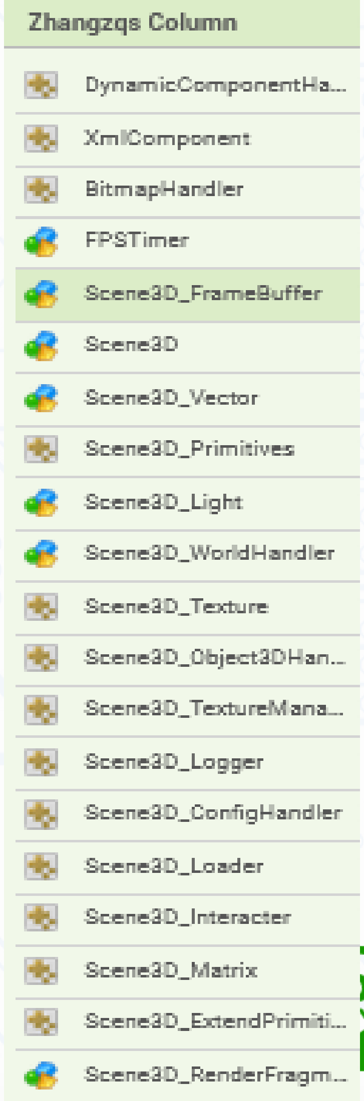
- KevinkunChart
- KevinkunOkayApi
- KevinkunBmobDB
- KevinkunContacts
- KevinkunEnhance
- KevinkunExcelUtil
- KevinkunBaiduFanyi
- KevinkunRegEx
- KevinkunAgoraRTC
- KevinkunAgoraRTM


ColinTree Column

- BaiduApiAccess
- YoutuApiAccess
- HorizontalScrollHandler
- VerticalScrollHandler
- SwipeRefresh
- ColinTreeListView
- ColinTreeListViewElem...
- ColinTreeSlideshow
- ColinTreeNinePatch
- AutoCompleteBox
- ColinTreeMath
- ColinTreeMathExtended
- JsonUtils
- JsonUtilsSimple
- ColinTreeFirstRun
- ColinTreeGoHome
- ColinTreeIMEI
- BaiduFanyi


Taifun Column

- TaifunFlashlight
- TaifunFile
- TaifunImage
- TaifunMetadata
- TaifunTools
- TaifunMath
- TaifunWiFi
- TaifunClipboard
- TaifunHeadset
- TaifunPM
- TaifunTextbox
- TaifunScreenshot
- TaifunTM
- TaifunAlarm
- TaifunSharing
- TaifunPlayer
- TaifunBattery
- TaifunSettings


Zhangzqs Column

- DynamicComponentHa...
- XmlComponent
- BitmapHandler
- FPSTimer
- Scene3D_FrameBuffer
- Scene3D
- Scene3D_Vector
- Scene3D_Primitives
- Scene3D_Light
- Scene3D_WorldHandler
- Scene3D_Texture
- Scene3D_Object3DHan...
- Scene3D_TextureMana...
- Scene3D_Logger
- Scene3D_ConfigHandler
- Scene3D_Loader
- Scene3D_Interacter
- Scene3D_Matrix
- Scene3D_ExtendPrimiti...
- Scene3D_RenderFragm...

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

JAMES PEEBLES
Nobel Prize in Physics 2019

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

Photo: A Mahmoud

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/

Example source

- **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

江西理工大学

Jiangxi University of Science and Technology

信息工程学院

School of information engineering

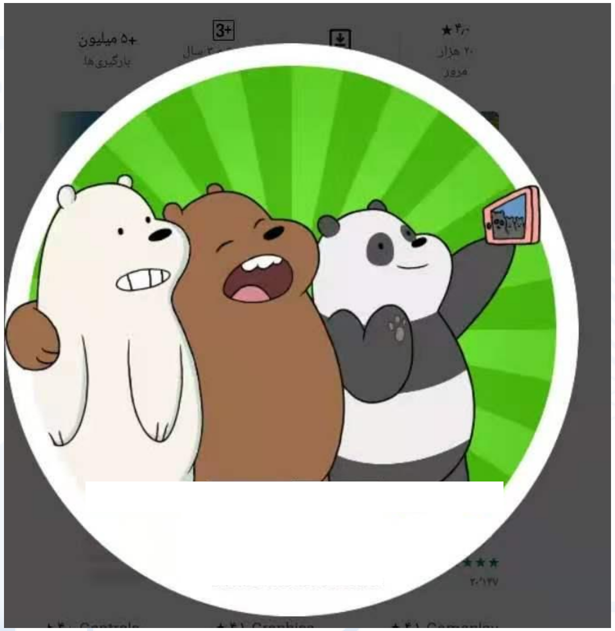
Digital Image Processing

THANK YOU

Gōng fū bú fù yǒu xīn rén

功夫不负有心人

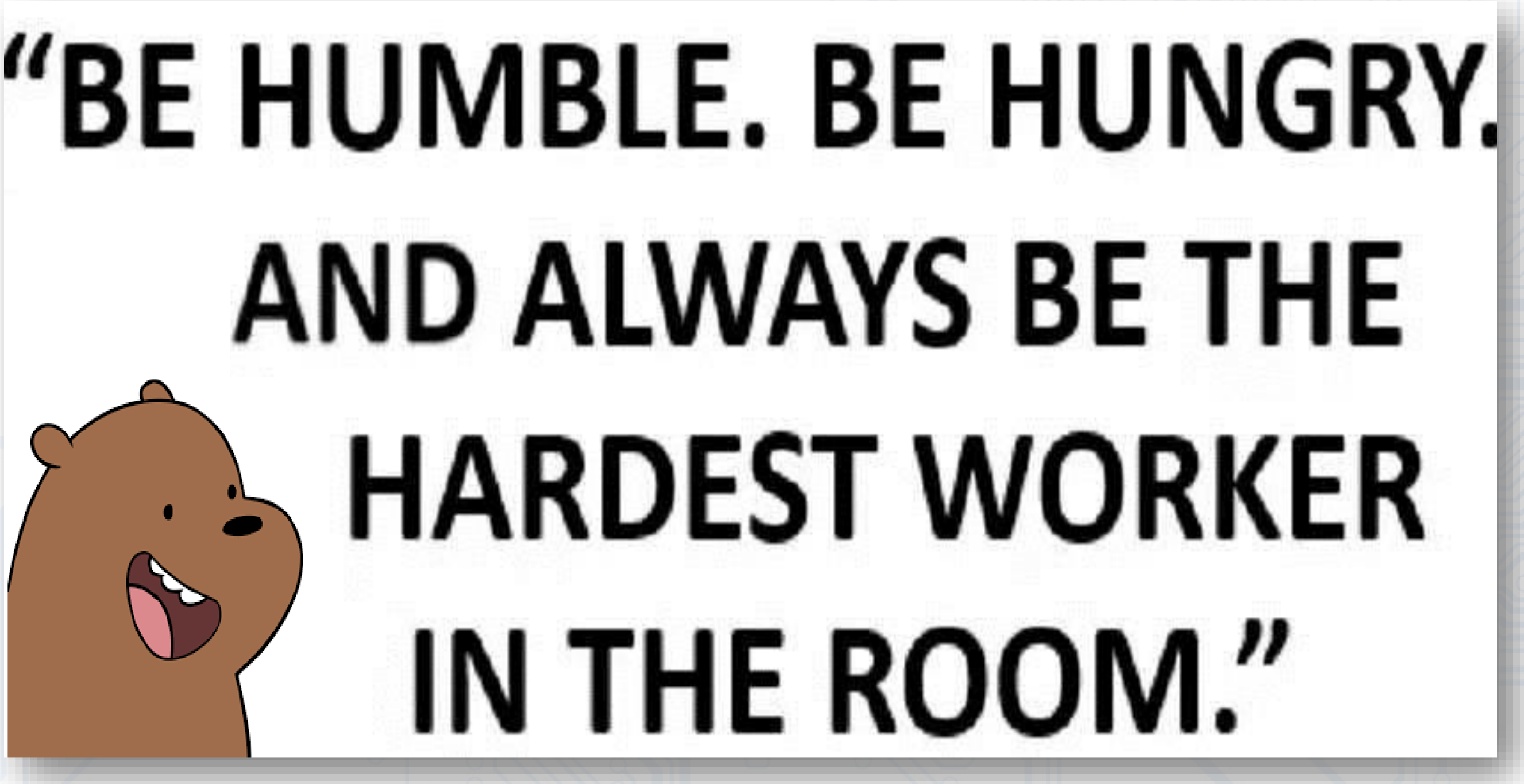
Efforts will be rewarded. No pains no gains.



過去就是過去

The past is just the past

App Inventor



"BE HUMBLE. BE HUNGRY. AND ALWAYS BE THE HARDEST WORKER IN THE ROOM."