



Mobile application development

移动应用开发



Lecture 010:

APP Inventor _Example

Review the example and introduce some blocks



Dr Ata Jahangir Moshayedi

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



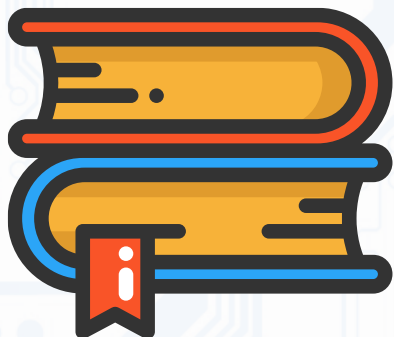
EMAIL: ajm@jxust.edu.cn

Autumn _2021



江西理工大学信息工程学院

JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING



MOBILE APPLICATION DEVELOPMENT

LECTURE 09: **APP Inventor _Example**

Review the example and introduce list blocks

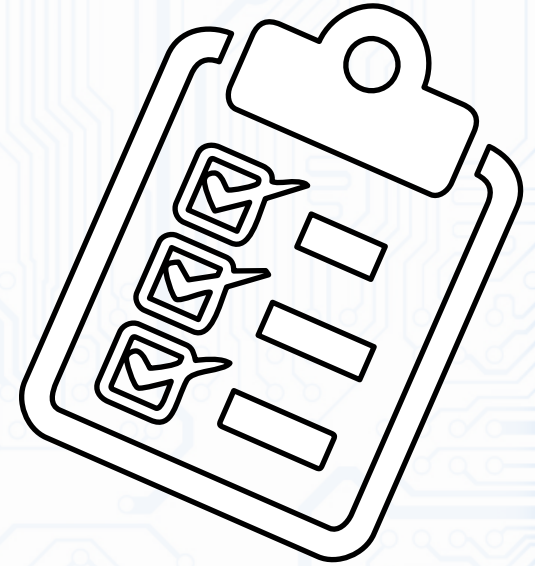




Agenda

– Review the example and introduce LIST blocks

- MIT App Inventor LIST Blocks
- Example 2





APP inventor _ LIST block

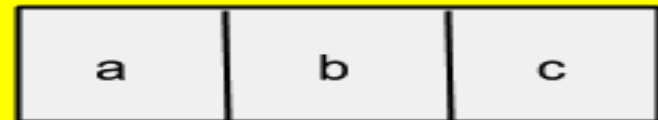


What is a list?

- **Many apps use lists of data. For example, a game may keep a list of high scores and your Facebook app keeps a list of your friends.**
- **Lists are a type of data structure used in every programming language, not just App Inventor.**
- **We use lists to create and manipulate different sets of values/elements.**

What is an index?

- The position of an element in a list is called its index.
- In App Inventor, the first element in a list is index 1.
- In the figure below, a has index 1, b has index 2, and c has index 3.



- What this means is that we can reference a specific element inside our list if we know which index it has and what the list name is.





APP inventor _ LIST block



App Inventor Lists

In App Inventor, it may be useful to have a list to store data such as numbers in your phone, replies for the Magic Eight Ball app, or the names of people you meet at a conference.

This block is a mutator and can be expanded or shrunk by clicking the blue plus sign.



Built-in

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





MIT App Inventor List Blocks

- create empty list
- make a list
- add items to list
- is in list
- length of list
- is list empty
- pick a random item
- index in list
- select list item
- insert list item
- replace list item
- remove list item
- append to list
- copy list
- is a list?
- reverse list
- list to csv row
- list to csv table
- list from csv row
- list from csv table
- lookup in pairs
- join with separator





MIT App Inventor List Blocks

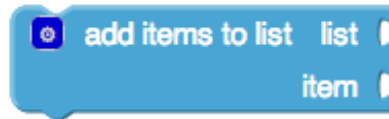


create empty list



- Creates an empty list with no elements.

add items to list



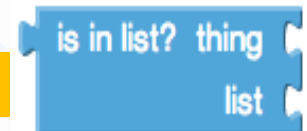
- Adds the given items to the end of the list. The difference between this and append to list is that append to list takes the items to be appended as a single list while add items to list takes the items as individual arguments. This block is a mutator.

make a list



- Creates a list from the given blocks. If you don't supply any arguments, this creates an empty list, which you can add elements to later. This block is a mutator. Clicking the blue plus sign will allow you to add additional items to your list.

is in list



If thing is one of the elements of the list, returns true; otherwise, returns false. Note that if a list contains sublists, the members of the sublists are not themselves members of the list. For example, the members of the list (1 2 (3 4)) are 1, 2, and the list (3 4); 3 and 4 are not themselves members of the list.





MIT App Inventor List Blocks



length of list

length of list list

- Returns the number of items in the list.

is list empty

is list empty? list

- If list has no items, returns true; otherwise, returns false.

pick a random item

pick a random item list

- Picks an item at random from the list.

index in list

index in list thing
list

- Returns the position of the thing in the list. If not in the list, returns 0.





MIT App Inventor List Blocks

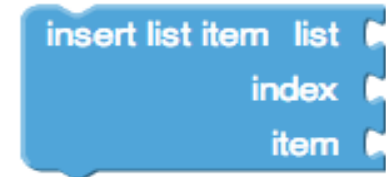


select list item



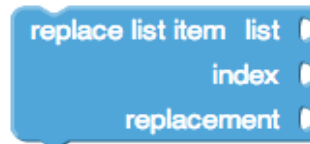
- Selects the item at the given index in the given list. The first list item is at index 1.

insert list item



- Inserts an item into the list at the given position.

replace list item



- Inserts replacement into the given list at position index. The previous item at that position is removed.

remove list item



- Removes the item at the given position.





MIT App Inventor List Blocks



append to list

append to list list1
list2

- Adds the items in the second list to the end of the first list.

copy list

copy list list

- Makes a copy of a list, including copying all sublists.

is a list?

is a list? thing

- If thing is a list, returns true; otherwise, returns false.

reverse list

reverse list list

- Reverses a copy of the list with items in the reverse order.
- For example reverse([1,2,3]) returns [3,2,1]





MIT App Inventor List Blocks



list to csv row

list to csv row list

- Interprets the list as a row of a table and returns a CSV (comma-separated value) text representing the row. Each item in the row list is considered to be a field, and is quoted with double-quotes in the resulting CSV text. Items are separated by commas. For example, converting the list (a b c d) to a CSV row produces ("a", "b", "c", "d"). The returned row text does not have a line separator at the end.

list to csv table

list to csv table list

- Interprets the list as a table in row-major format and returns a CSV (comma-separated value) text representing the table. Each item in the list should itself be a list representing a row of the CSV table. Each item in the row list is considered to be a field, and is quoted with double-quotes in the resulting CSV text. In the returned text, items in rows are separated by commas and rows are separated by CRLF (\r\n).

list from csv row

list from csv row text

Parses a text as a CSV (comma-separated value) formatted row to produce a list of fields. For example, converting ("a", "b", "c", "d") to a list produces (a b c d).

list from csv table

list from csv table text

Parses a text as a CSV (comma-separated value) formatted table to produce a list of rows, each of which is a list of fields. Rows can be separated by newlines (\n) or CRLF (\r\n).





MIT App Inventor List Blocks

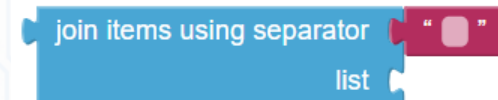


lookup in pairs



- Used for looking up information in a dictionary-like structure represented as a list. This operation takes three inputs, a key, a list pairs, and a notFound result, which by default, is set to “not found”. Here pairs must be a list of pairs, that is, a list where each element is itself a list of two elements. Lookup in pairs finds the first pair in the list whose first element is the key, and returns the second element. For example, if the list is ((a apple) (d dragon) (b boxcar) (cat 100)) then looking up ‘b’ will return ‘boxcar’. If there is no such pair in the list, then the lookup in pairs will return the notFound parameter. If pairs is not a list of pairs, then the operation will signal an error.

join with separator



- Joins all elements in the specified list by the specified separator, producing text as a result.





List Blocks On App Inventor and List activity



- List is a necessity in almost every app regardless of what programming language you use.
 - This is the easiest way to create and manipulate a set of values/items/elements in an ordered fashion.
1. Create a List
 2. Add Items to a List
 3. Display List Items
 4. Remove List Items
 5. Replace an Item
 6. Search for an Item
 7. Sort a List
 8. Shuffle a List





1. Create a List

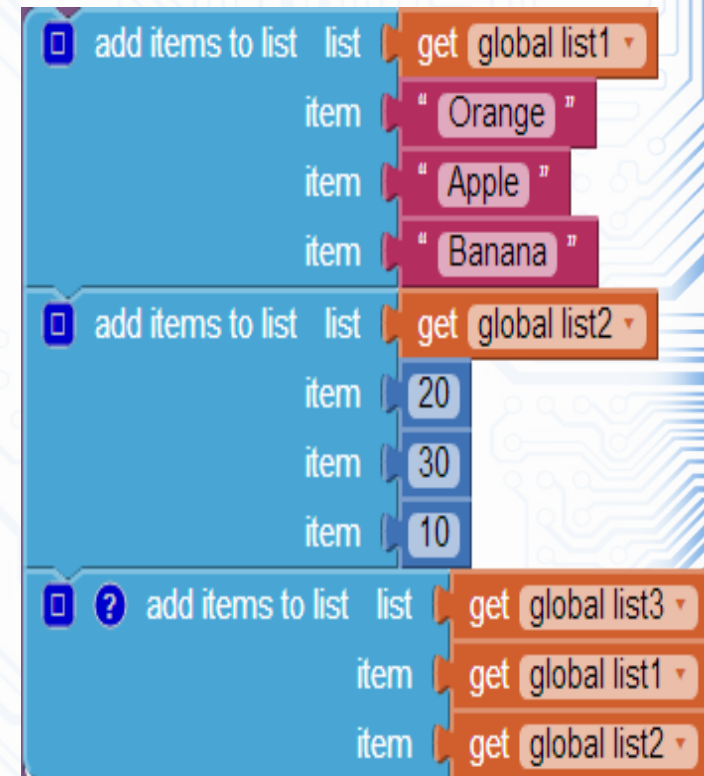
Creating a list is as simple as this-





2. Add Items to a List

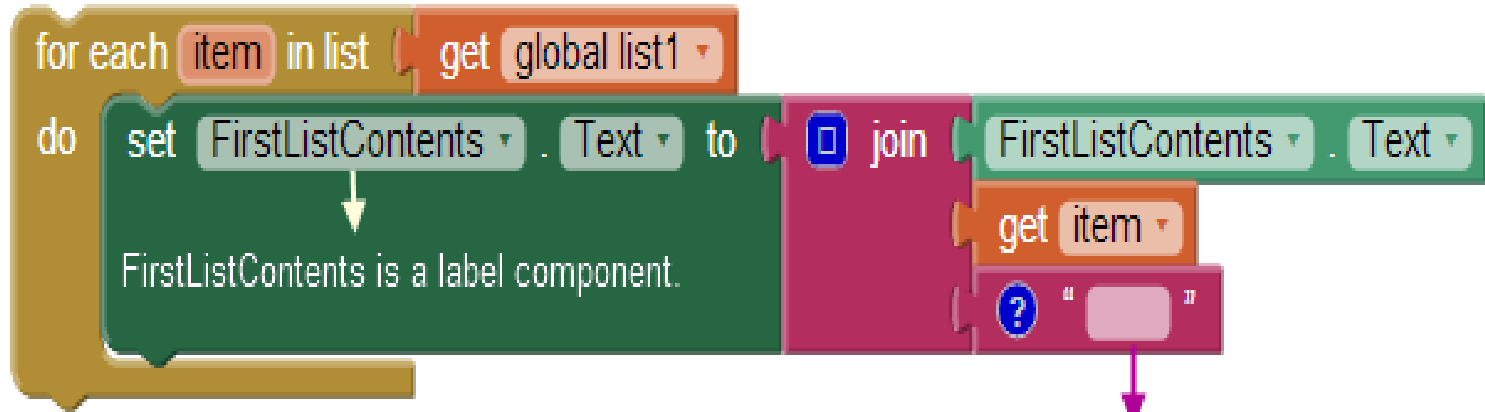
- Adding an item can be done different ways.
- If you look at the image above, we have populated all three lists with different items.
- First one will have a list of three items which are Orange, Apple, and Banana; the second one is consisted of another three items – 20, 30, and 10.
- The last one, list3 is consisted of two items which are actually lists. Yes, you can have a list of lists. If you take the first item from list3 which is also a list, you will get a list of three items which contains Orange, Apple, and Banana.





3. Display List Items

- A fancy way to display list items would be using ListView component.
- In the example below, we used a Label component named FirstListContents to display the contents of list1.



This is a text block that contains some spaces.





4. Remove List Items

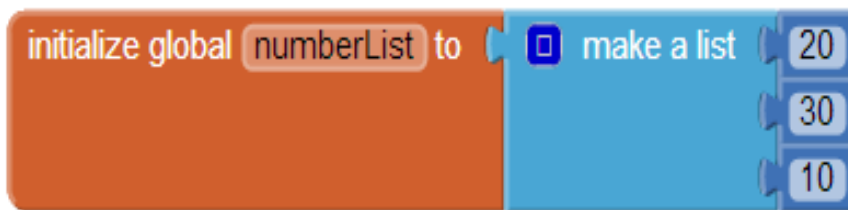
- To remove an item, all you need to know is the index/position of the item you want to remove. If you want to remove number 30 from numberList below, you'll have to pass 2 as the index as you can see 30 is in position/index 2 in the numberList below.





5. Replace an Item

- To replace a list item, you need to provide with three arguments – the list where you want to replace an item, the index of an existing item in the list, and the new item you want to replace an existing item with.



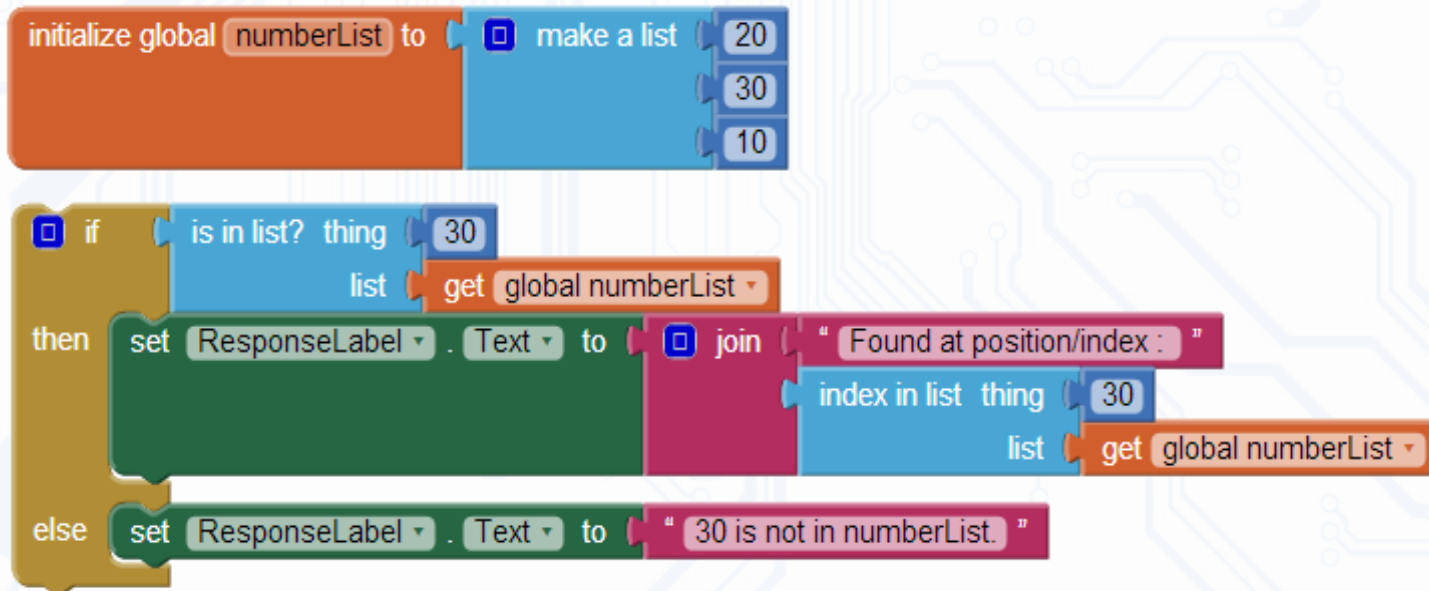
In the image above, initially we had three items 20, 30, and 10 in `numberList`. We used `replace list item` block to replace the item at index 1 which was 20 with a new item 100. Now our `numberList` contains 100, 30, and 10.





6. Search for an Item

- Searching for an item can be a little tricky because sometimes you will need to know if a particular item exists in a list, if it does, then you might be interested in knowing where or in which position that item exists. If you have multiple occurrences of the same item in a list, you may also need to know all the positions where the item exists in the list.





- More study please read the share PDF on MOOC





Reference

- <http://ai2.appinventor.mit.edu/reference/blocks/lists.html#selectlistitem>
- <https://appinventor.mit.edu/explore/content/alertme.html>
- **Teaching with AppInventor** <http://appinventor.mit.edu/explore/teach.html>
- **AppInventor Tutorials:** <http://appinventor.mit.edu/explore/ai2/tutorials.html>
- **Sounds** <http://www.soundbible.com>
- **App Inventor:** <http://appinventor.googlelabs.com/>
- **Appinventor.org:** <http://www.appinventor.org/>
- **Wolber, Abelson et al. text:** <http://www.appinventor.org/text2011>
- **Group:** <http://groups.google.com/group/app-inventor-instructors>
- **Wolber course:** <http://appinventor.org/course-in-a-box>
- **Morelli course:** <http://turing.cs.trincoll.edu/~ram/cpsc110/>



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

STEVEN CHU
Nobel Prize in Physics 1997



江西理工大学

Jiangxi University of Science and Technology

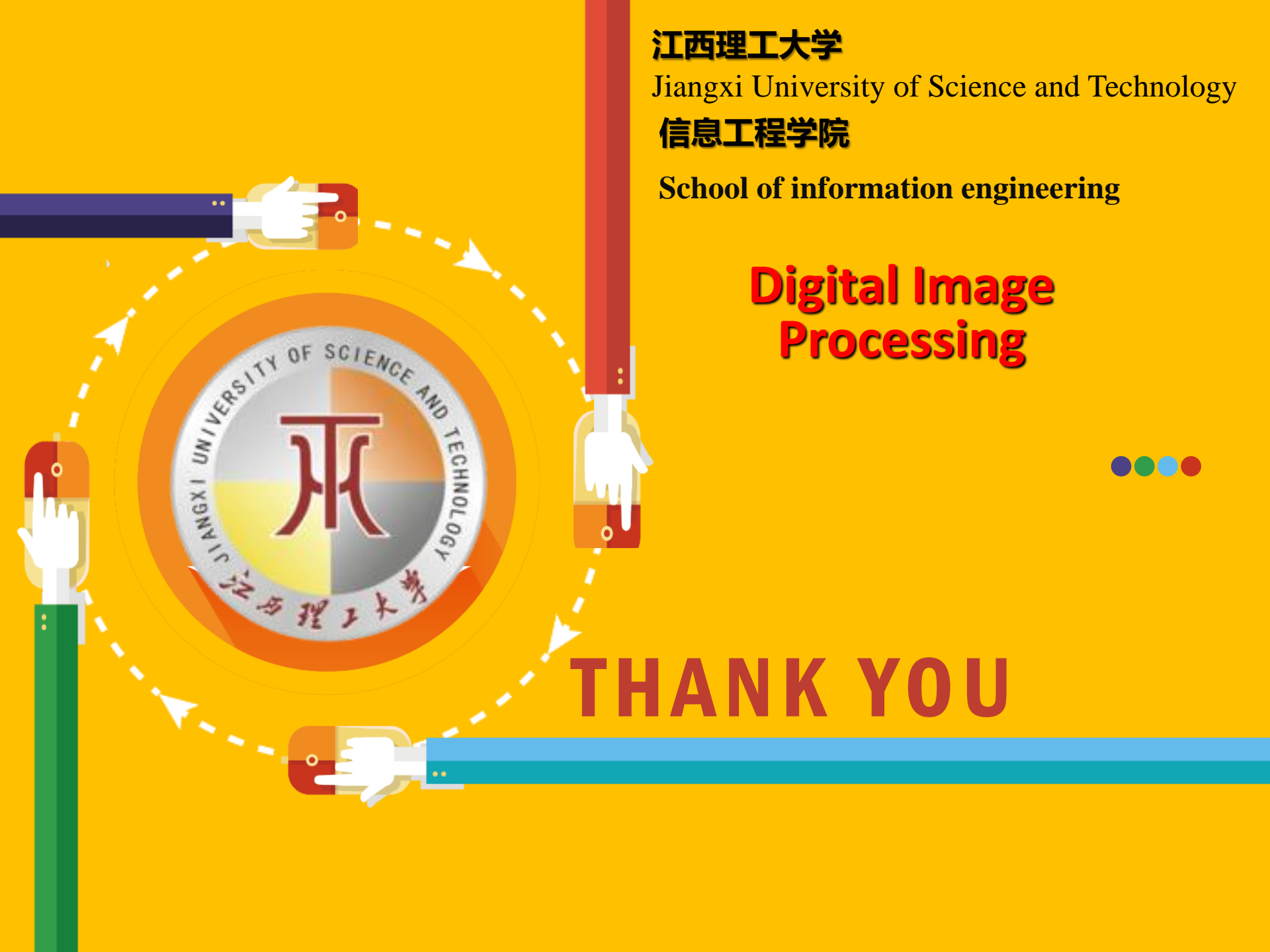
信息工程学院

School of information engineering

Digital Image Processing



THANK YOU





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

