KUAS
KYOTO UNIVERSITY of ADVANCED SCIENCE
京都発世界人財

京都先端科学大学

Introduction to Design (Track 3)

4.1 Procedure

Zilu Liang

www.zilu-liang.net/id3

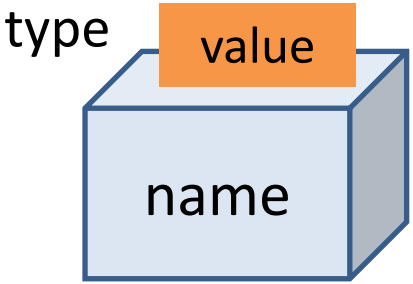
Important!!

Please disinfectize your hands before entering the classroom!
入室前にアルコールを使用して手指消毒を行ってください。

Please disinfectize your chair and table!
①ペーパーにアルコールを噴霧してください。
②アルコールが噴霧されたペーパーで、使用箇所（テーブル、椅子など）を拭き取ってください。
③使用済のペーパーは廊下のごみ箱に捨ててください。

1. Recap of Week 3

Variable (変数)



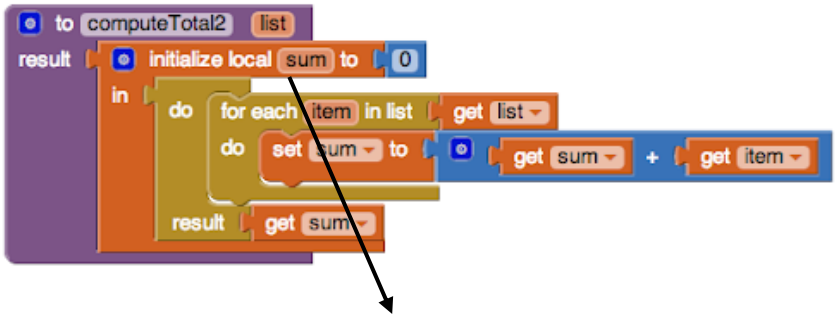
- A variable is a storage location in the memory
- A variable has three elements: name (変数名), value (数値) and type (変数の型)
- Variable type could be integer, double, string, booleen, etc.

- Variable names should be descriptive
- Avoid using space in variable names

Variable Scope and Initialization

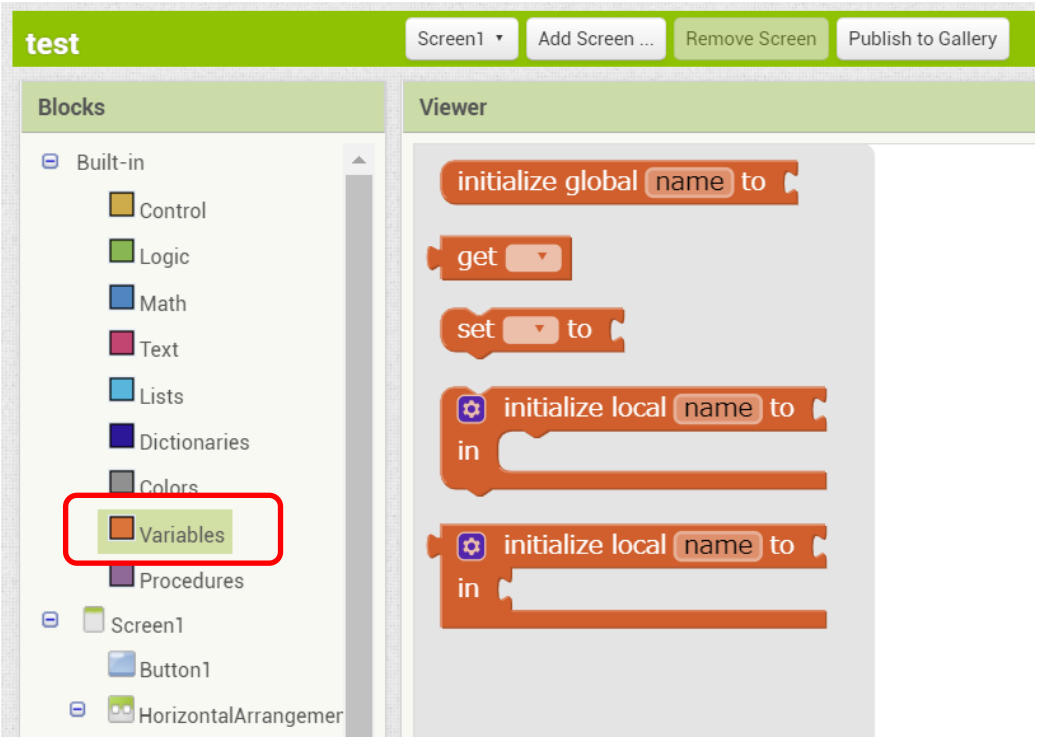
- Global variables are valid for the whole program; local variables are only valid for one place (e.g., in one block)
- Both global and local variables need to be initialized first

Initialize a global variable called ‘score’ and set it to 0



Initialize a local variable called ‘sum’ and set it to 0; this variable is only valid inside this block

Five Main Types of Variable Blocks



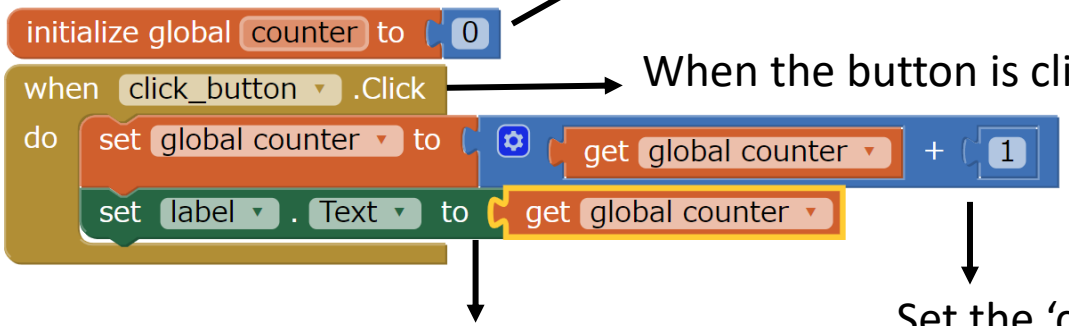
日本語版　https://appinventor2.wiki.fc2.com/wiki/Variables

Counter App

In the hands-on tutorial last week, we made a counter app. Every time you click the button, the counter will increase by 1.

Code Anatomy

Let’s review the blocks code of the Counter App we made last week.



Initialize the global variable ‘counter’ and set it to 0

When the button is clicked, do these things

Set the ‘counter’ variable to its current value + 1

Set the label text to the current value of the ‘counter’ variable

2. Procedure

How to make 肉じゃが

Procedure (手順)

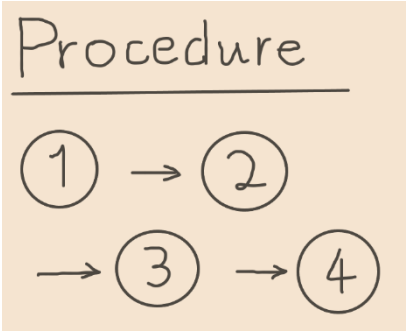
1. Start by cutting up the meat and onion into thin slices. Peel the carrots and potatoes and then chop into bite-size pieces.
2. Warm some vegetable oil in a frying pan, then add the onions cook over a medium heat for a few moments before adding the meat and letting it cook. After that, add in the potatoes and carrots.
3. …

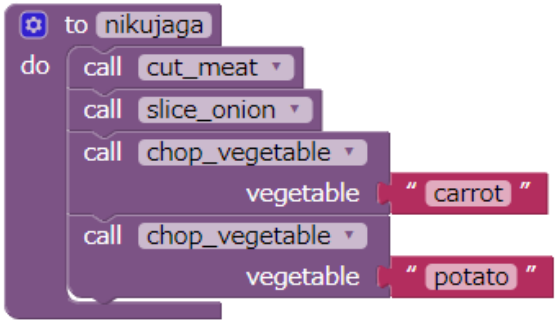
Procedure

A procedure is **a set of instructions** that is grouped together, given a name, and can perform specific tasks. It may also be called a function in programming.

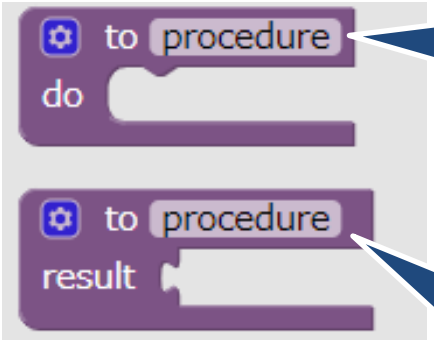
For example, a recipe for 肉じゃが　is an example of a procedure. The cook must follow the instructions step-by-step to produce the dish.

We use ‘Procedures’ in the blocks pane to create new procedures that we can use repeatedly





Two Types of Procedure in App Inventor

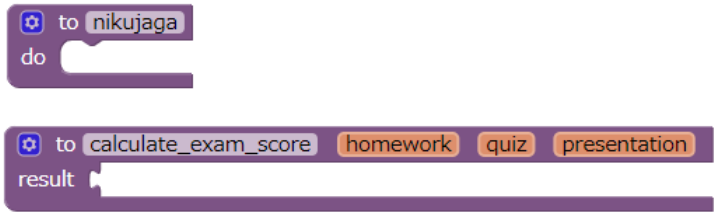


This type of procedure only do something; when this procedure is called, a set of actions will be performed.

This type of procedure returns some value. When this procedure is called, a set of actions will be performed, and a result will be generated and returned.

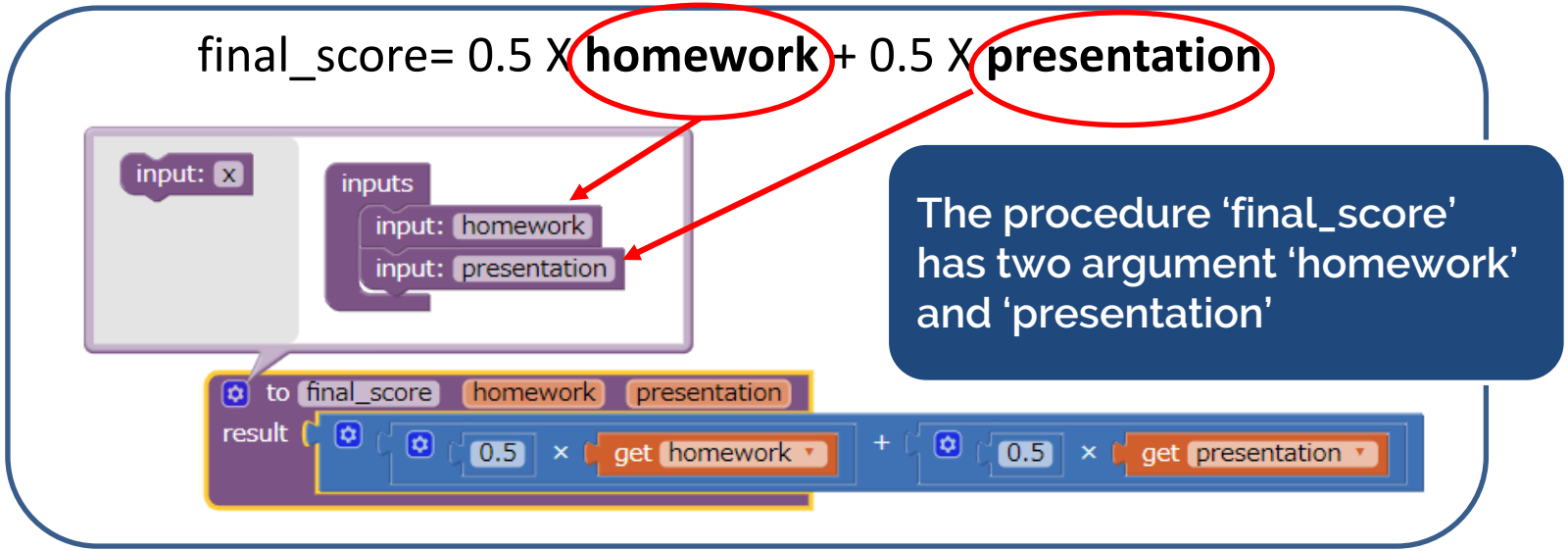
Procedure Name

- When you create a new procedure block, App Inventor chooses a unique name automatically. You can change the name.
- Procedure names in an app must be unique. App Inventor will not let you define two procedures in the same app with the same name.
- You can rename a procedure at any time while you are building the app. App Inventor will automatically rename the associated call blocks to match.

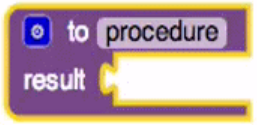


Procedure Argument

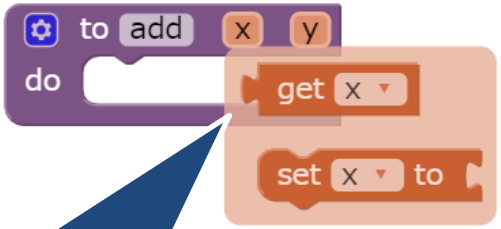
- An argument is an **input** to a procedure. It's similar to the concept of 'independent variable' in mathematical functions.
- Some procedures require knowing some bits of information that change how the procedure is run.



Add Argument in Procedure



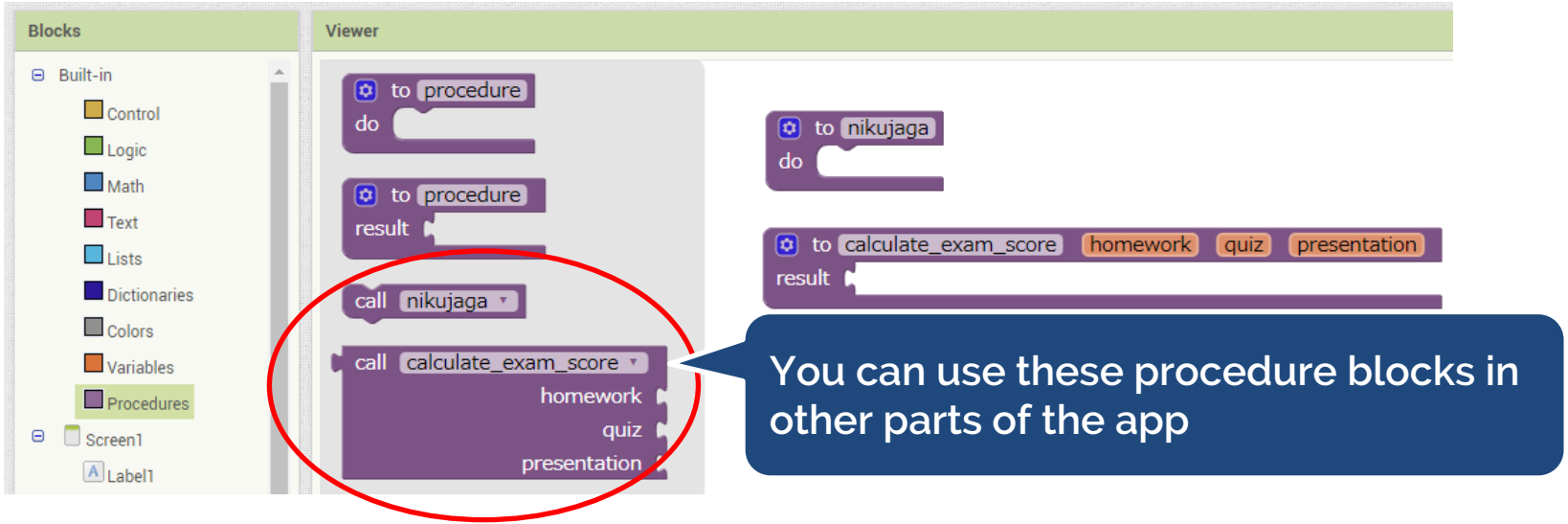
When you create a procedure, you can use the mutator button to add arguments.



By hovering your mouse over an argument, you will see a get and set block appear.

Procedure in App Inventor

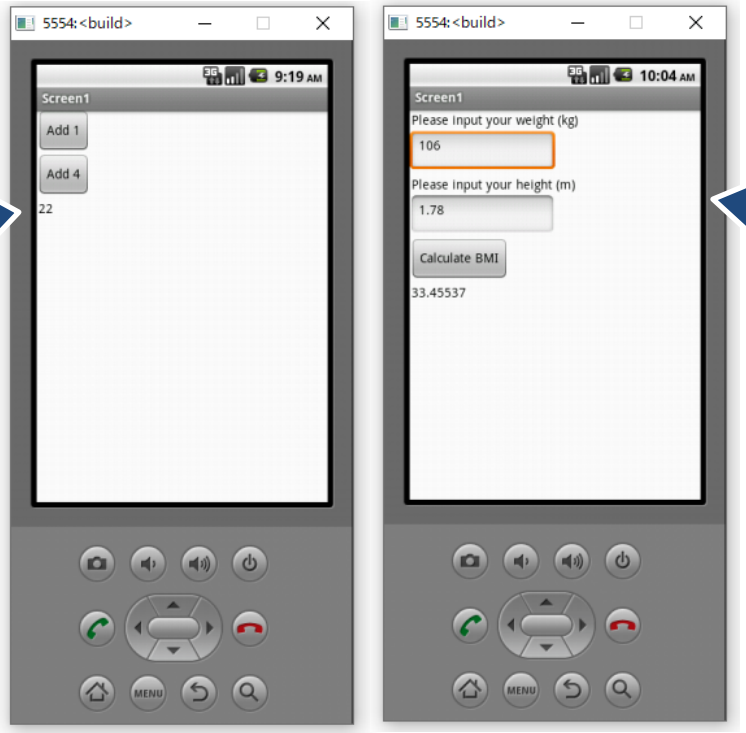
When you create a procedure, App Inventor automatically generates a call block and places it in the ‘Procedures’ drawer. You use the call block to invoke the procedure.



Preview of Hands-on Tasks

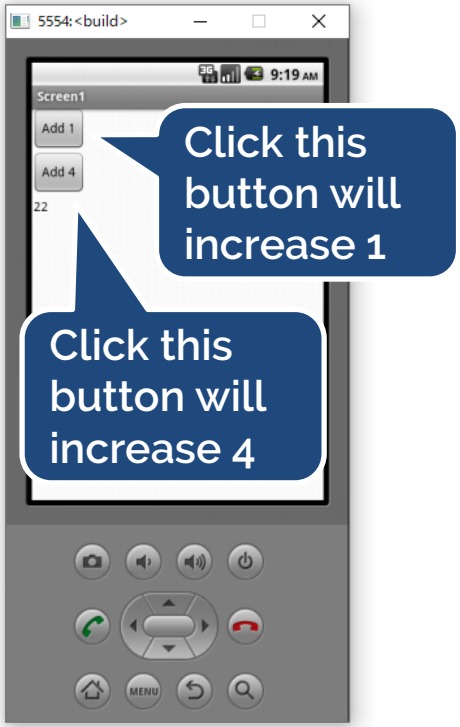
We'll make two apps today

The counter app will use the 'to-do' procedure



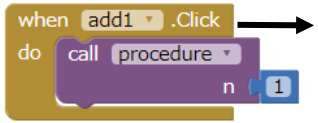
The BMI calculator app will use the 'to-result' procedure

Code Anatomy: Counter App

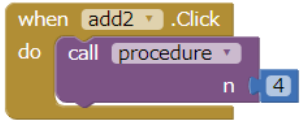


Initialize the global variable ‘counter’ and set it to 0

Define a ‘procedure’ that takes in the argument ‘n’, increase the counter by ‘n’, and show the result in the label

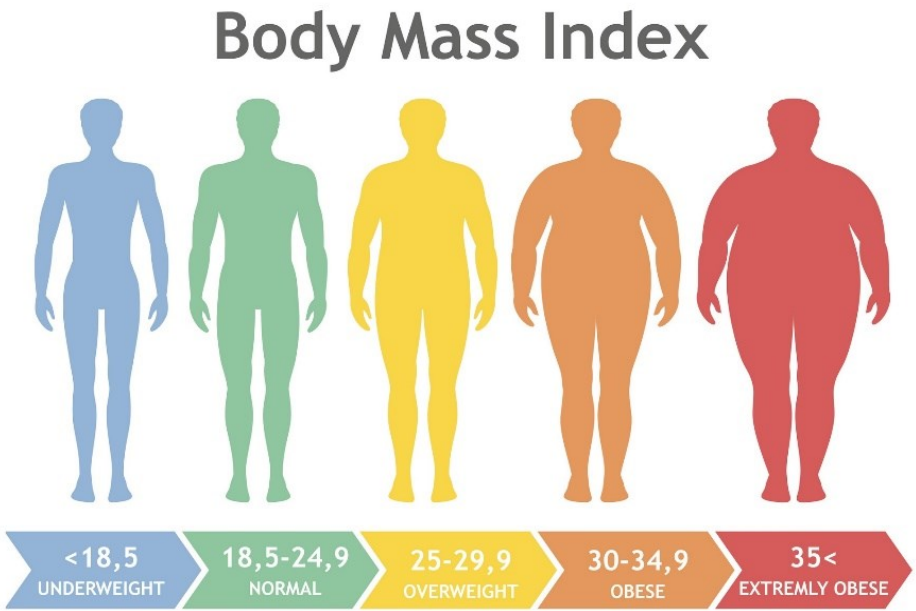


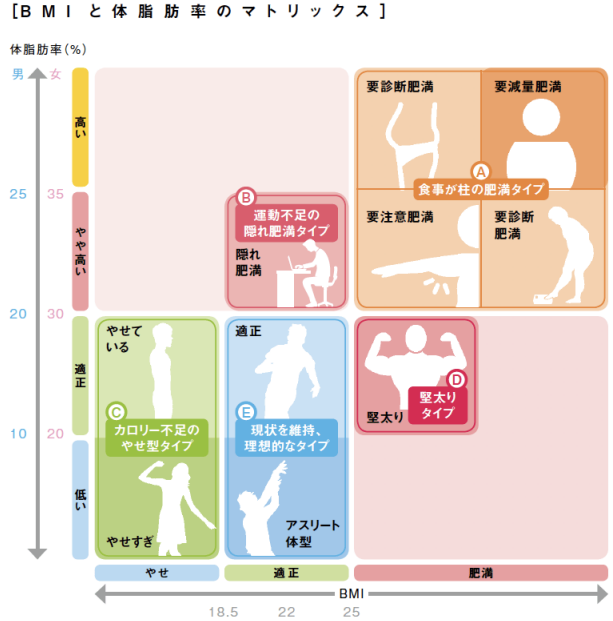
When ‘add1’ button is clicked, call ‘procedure’ and pass the argument ‘n=1’



When ‘add2’ button is clicked, call ‘procedure’ and pass the argument ‘n=4’

What is BMI?


Body Mass Index
<18,5
UNDERWEIGHT
18,5-24,9
NORMAL
25-29,9
OVERWEIGHT
30-34,9
OBESE
35<
EXTREMLY OBESE


[BMIと体脂肪率のマトリックス]
体脂肪率(%)
男
女
高い
やや高い
適正
低い
25
35
20
30
10
20
要診断肥満
要減量肥満
A
食事が柱の肥満タイプ
B
運動不足の
隠れ肥満タイプ
隠れ
肥満
要注意肥満
要診断
肥満
やせて
いる
C
カロリー不足の
やせ型タイプ
やせすぎ
適正
E
現状を維持、
理想的なタイプ
アスリート
体型
D
堅太り
タイプ
堅太り
やせ
適正
肥満
BMI
18.5
22
25

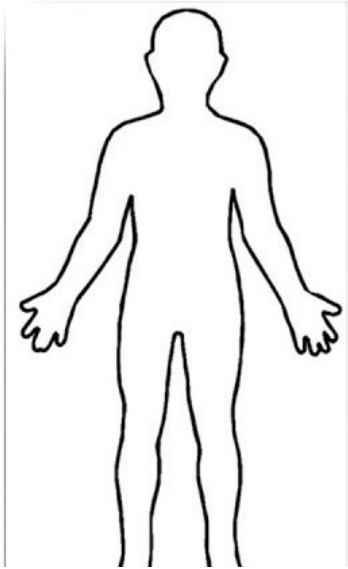
How to calculate BMI?

How to calculate your

Body Mass Index

(BMI) value

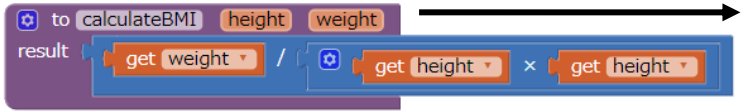
$$\mathrm{BMI} = \frac{\mathrm{mass_{kg}}}{\mathrm{height_{m}^{2}}} = \frac{\mathrm{mass_{lb}}}{\mathrm{height_{in}^{2}}} \times 703$$



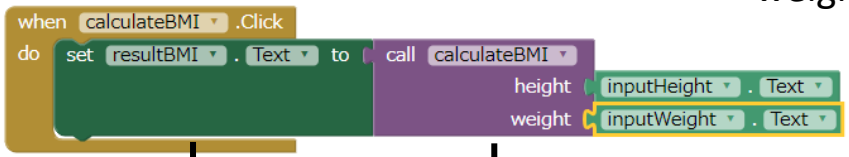
Code Anatomy: BMI Calculator App

A user needs to input weight and height

Click this button will calculate BMI



Define a ‘calculateBMI’ procedure that takes in the argument ‘height’ and ‘weight’ to calculate BMI



1. Set the ‘weight’ and ‘height’ arguments to the values that a user input in the textboxes

2. Call the ‘calculateBMI’ procedure and pass the two arguments ‘weight’ and ‘height’, and return the result value to the block on the left

3. Set the ‘resultBMI’ text to the result returned from the ‘calculateBMI’ procedure

Any questions?


KUAS
KYOTO UNIVERSITY of ADVANCED SCIENCE
京都先端科学大学