

CHAPTER 6

Procedures and Functions

starting out with >>> **APP INVENTOR**
FOR ANDROID



TONY GADDIS · REBECCA HALSEY

Topics

- Modularizing Your Code with Procedures
- Passing Arguments to Procedures
- Returning Values from Procedures

Modularizing Your Code with Procedures

- Procedures can be used to break up a complex program into small, manageable pieces.
- Modularization tends to simplify code.
- The benefit of using procedures is known as *code reuse*.

Modularizing Your Code with Procedures

So far you have experienced procedures in the following ways:

- Creating event handlers.
- Executing built-in procedures and functions, such as the `Sound` component's `Play` procedure.

Modularizing Your Code with Procedures

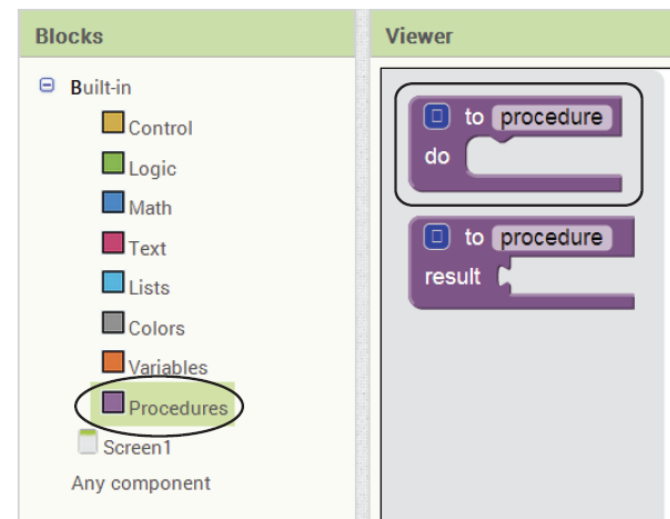
Procedures and Procedures with Results (Functions)

- When you call a procedure, it simply executes the blocks it contains and then terminates.
- When you call a *procedure with results*, it executes the blocks that it contains and then it returns a value back to the block that called it.
- Procedures with the results are also known as *functions*.

Procedures

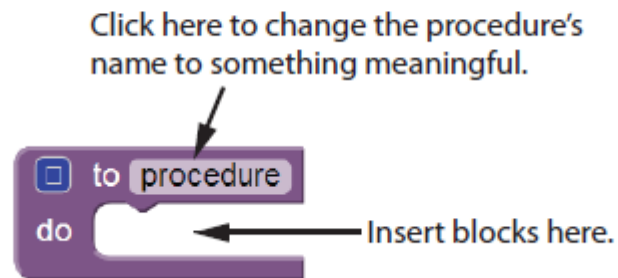
- A procedure is a block that contains other blocks.
- In the Block's Editor, go to the *Built-in* section of the Blocks Column, then select *Procedures*.
- In Figure 6-1, select the `to procedure do` block from the drawer. Usually this is referred to as the procedure block.

Figure 6-1 The procedure Block (Source: MIT App Inventor 2)



Procedures

Figure 6-2 A procedure Block (Source: MIT App Inventor 2)



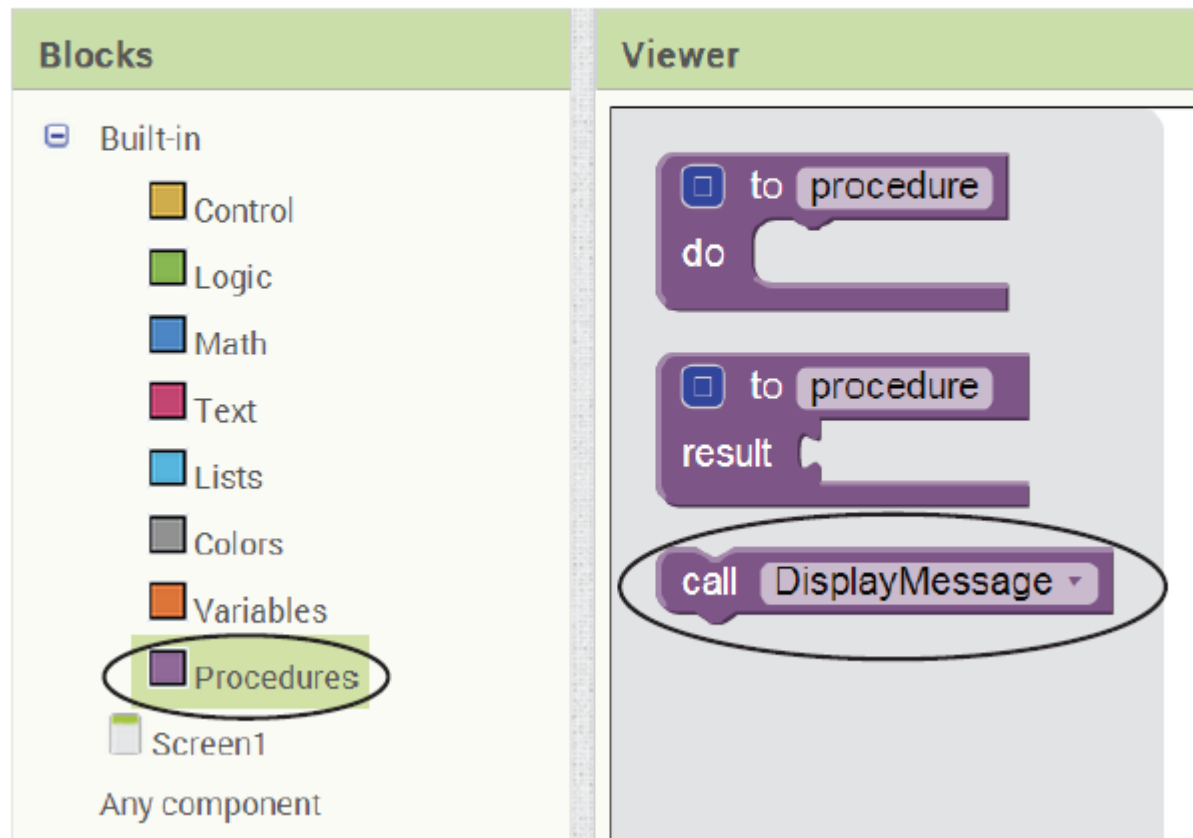
- Figure 6-2 shows an empty procedure block.
- The word that appears at the top of the block is the procedure's default name. Change the name to something more meaningful.

Procedures

- You execute a procedure with a `call` block.
- When you create a `procedure` block, App Inventor automatically creates a `call` block for the procedure, which you will find by opening the Procedures drawer of the Blocks Column.
- Figure 6-4 shows a `call` block for a procedure named `DisplayMessage`.

Procedures

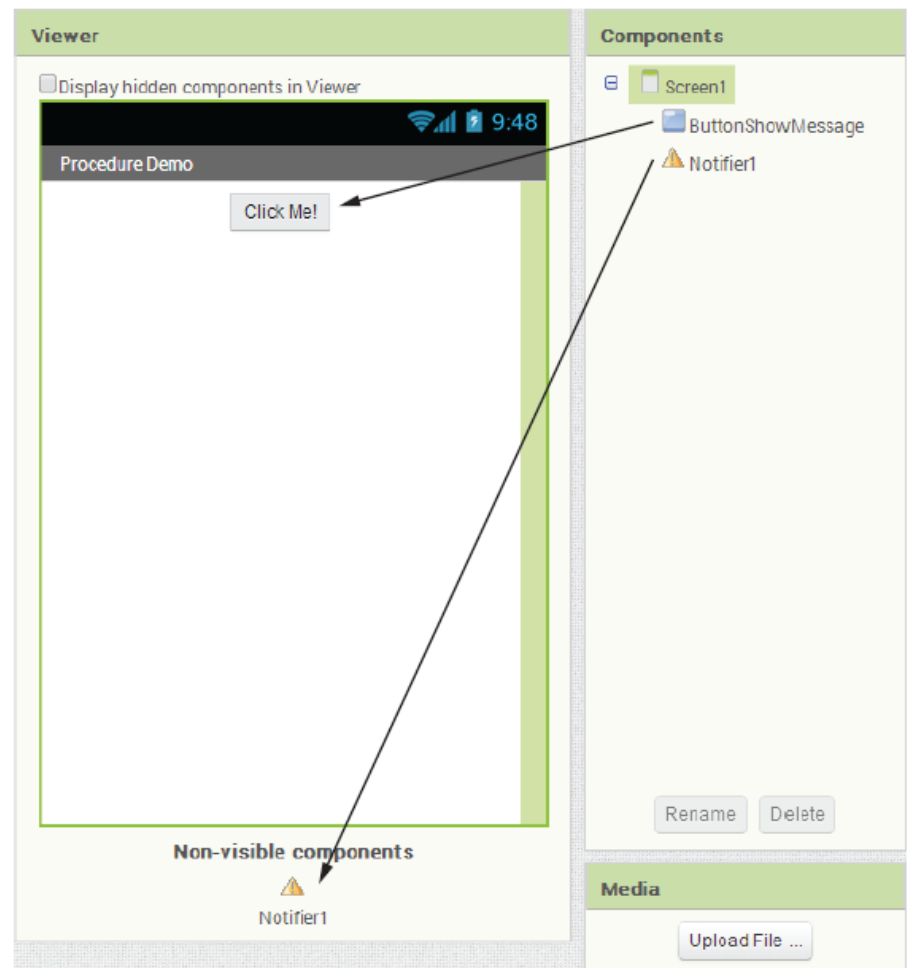
Figure 6-4 The call Block for a Procedure Named DisplayMessage
(Source: MIT App Inventor 2)



Procedures

- Figure 6-5 shows the ProcedureDemo app in the Designer.
- The app has a `Button` component on its screen and a `Notifier` component for displaying messages.

Figure 6-5 The ProcedureDemo App in the Designer (Source: MIT App Inventor 2)

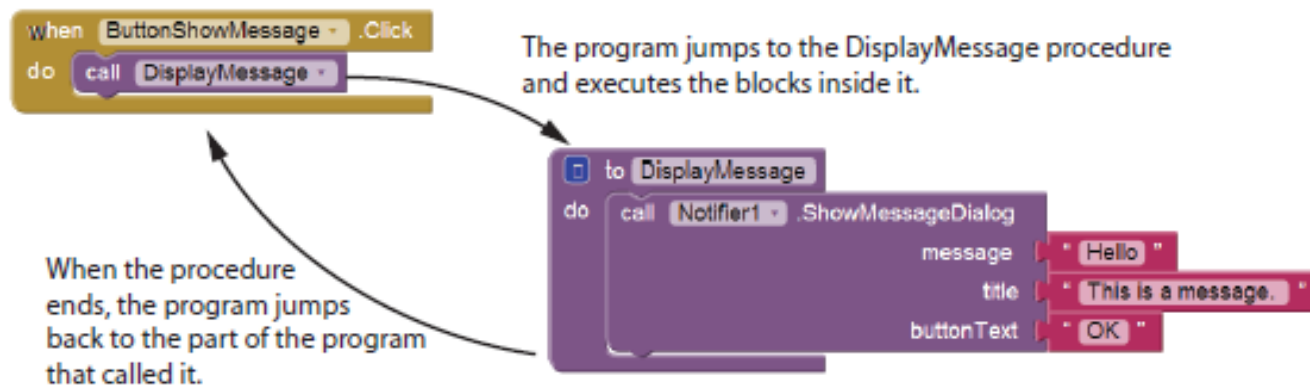


Procedures

- The app has a `Click` event handler for the `ButtonShowMessage` component and a procedure named `DisplayMessage`.
- Inside the `ButtonShowMessage.Click` event handler, we have a `call` block that calls the `DisplayMessage` procedure.
- When the procedure ends as shown in Figure 6-7, the program jumps back to the part of the program that called the `DisplayMessage` procedure.

Procedures

Figure 6-7 Calling a Procedure (Source: MIT App Inventor 2)



Procedures

Top-Down Design

Programmers commonly use a technique known as *top-down design* to break down a program into procedures. The process of top-down design is performed in the following manner:

- The overall task that the program is to perform is broken down into a series of subtasks.
- Each of these subtasks is examined to determine whether it can be further broken down into more subtasks.
- Once all of these subtasks have been identified, they are written in code.