Block Python – The Basics

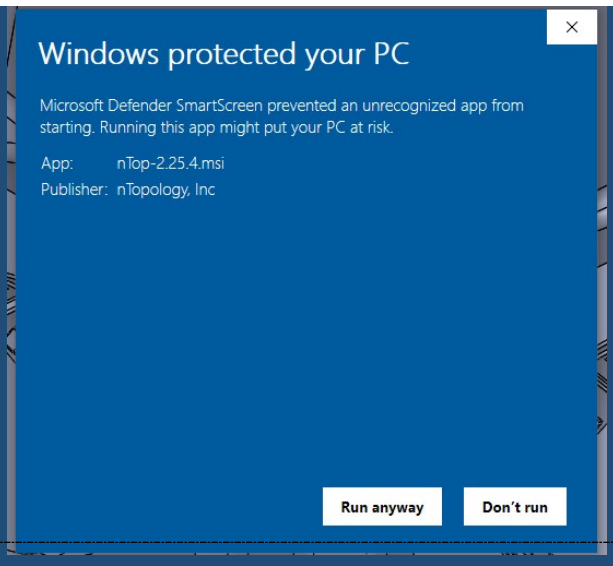
By Hamish BRIGGS

Co-Contributed by:

Josh Hall

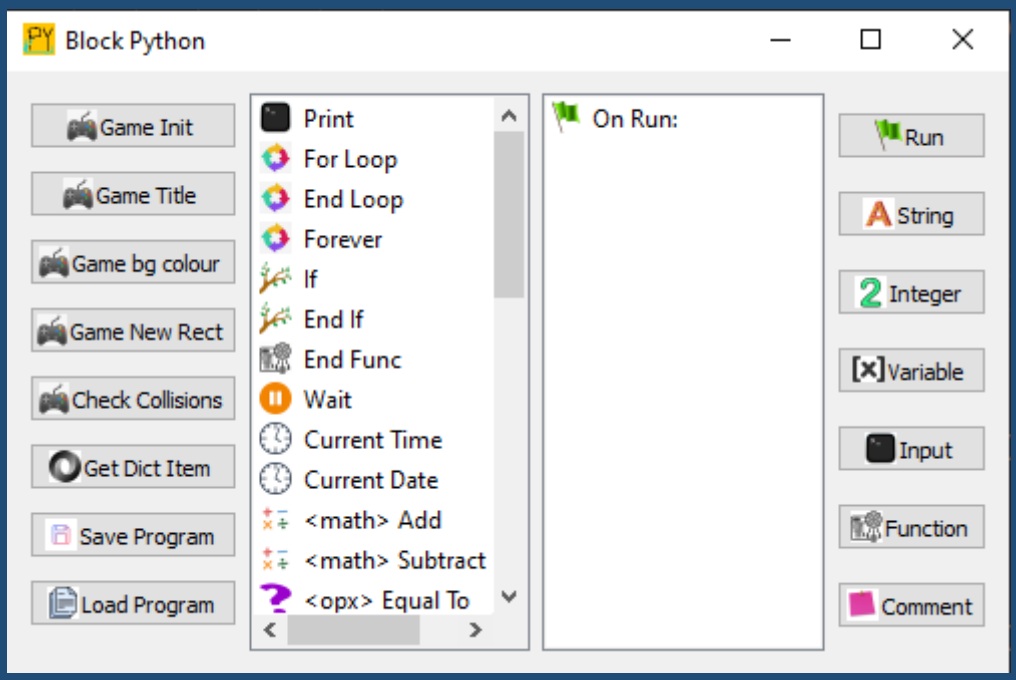
Download And Installation

To download Block Python, please visit https://github.com/Something231/Block-Python/releases and click on the latest release. From there click "Source Code (zip)" and it should download. After this, please navigate to the downloads folder and extract the folder. Once the folder is extracted, please open the folder, and run the file "run.bat". You will likely encounter a message saying Windows protected your PC, just click More info and press Run Anyway. After this, the file should run and install the dependencies for block python and run the program. Just exit the terminal to close the program.



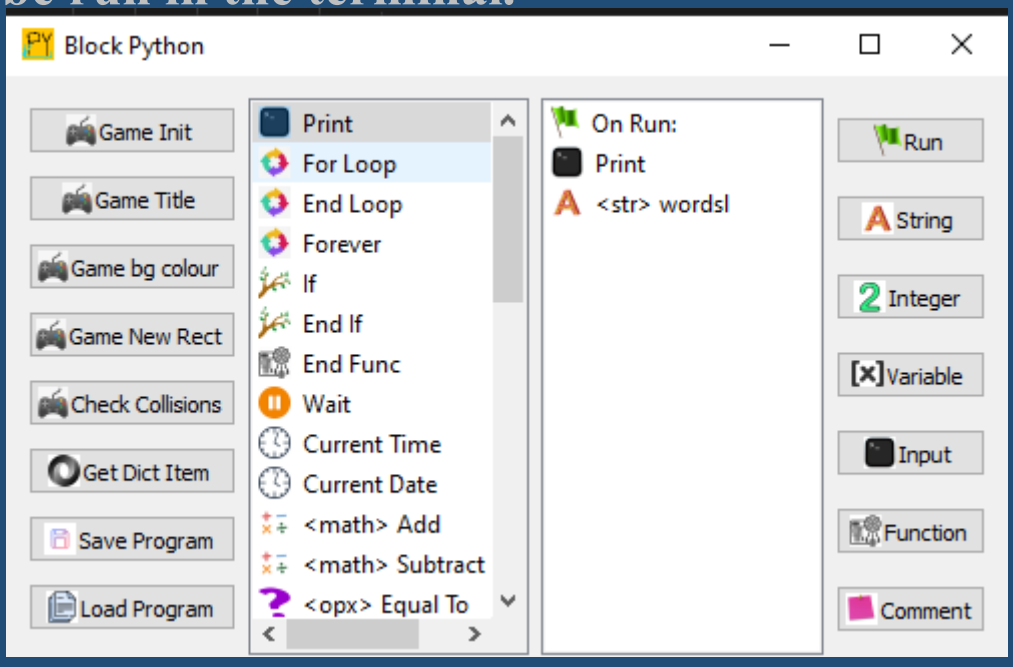
Introducing The Editor

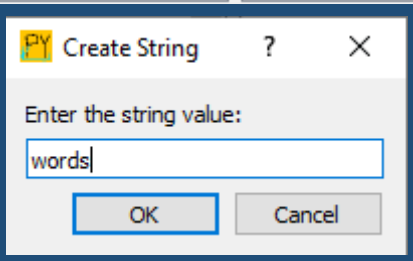
Run the file “run.bat” to start the program and an editor should appear. This is the basic editor for Block Python and most events should happen here. On the Far left is the game creation options, which we will dive into later. To the right of that is the library, a selection of blocks that you might use in your programs. To the right of that is the workspace, a space that blocks can be dragged into. This space is where the actual program that is made by the user can be constructed and edited. On the far right of the editor is the options tab, where the run button is placed, as well as button that allow the creation of certain types of blocks that require customisation.



Creating simple programs

In the editor, drag the “Print” block into the workspace. The Print block allows the program to display text in the terminal, however this block needs to know what to display in the terminal. The Print block only needs to know one thing (one argument) to be able to display text, which is what text should be displayed. To do this, click the string (text) button on the right. This should create a window that allows you to name the string. Once you have named the string press “run” and the program will be run in the terminal.





Basic Block Types

The uses of the following blocks are displayed below:

RUN BUTTON – Runs the program.

STRING BUTTON – Creates a string (text) object that can be used as an argument.

INTEGER BUTTON – Creates an integer (number) object that can be used as an argument.

COMMENT BUTTON – Creates a comment in the program that does nothing apart from helping to create notes in a program.

INPUT BUTTON – Creates a prompt for the user in the program which allows the user to provide input to influence the program.

PRINT – Displays text in the terminal, Requires one string, integer, input, function, or variable as an argument.

FOR LOOP – Repeats everything inside it for the given number of loops, Requires one integer argument (number of loops)

END LOOP – Placed at the end of a loop to indicate that code below is separate from the loop and will not be part of the for loop.

Basic Block Types

FOREVER – A type of loop that will loop forever.

CURRENT TIME – Gets the current time, can be used as a function type argument.

CURRENT DATE – Gets the current date, can be used as a function type argument.

VARIABLE BUTTON – Creates a variable block, you will need to give the variable a name.

VARIABLE – Any block that begins with <var>. This block be assigned something by giving it an argument that is a string, integer, input, function, or variable. Once it is assigned, it can be used as an argument in the rest of the program.

FUNCTION BUTTON – Creates both a <createfunc> block (in the workspace), and a <func> block (in the library).

CREATE FUNC – Everything passed between this and ENDFUNC can be run again in a program easily without the need for rewriting everything again.

END FUNC – Ends the function described by a CREATE FUNC block.

Basic Block Types

FUNC – Runs the code described by <createfunc> with the matching name.

IF – Checks weather something is of the right conditions (denoted by <opx>) to something else, then if the conditions are correct, then will run the code below, requires 3 arguments, a value of any type, an <opx> block to state the conditions, and another value to compare.

ENDIF – Ends an IF block, so that the code below is not conditional.

<OPX> BLOCKS – Work only in the construction of IF statements, are used to describe the conditions needed to run the code inside, such as if something is equal to something else.

Advanced Block Types

Experiment with the other blocks to create more advanced programs. An example of an advanced program is in the “example.txt” file, and can be tested by moving the code into the “saved.bloc” file and used by pressing load.