

MICROSOFT

SMALL BASIC INTRODUCTION



Microsoft

Small Basic

Every Kid Can Code

Table of Contents

Practice 1: Displaying Text - <https://aka.ms/SB-Practice01>

Practice 2: Graphics and Coordinates - <https://aka.ms/SB-Practice02>

Practice 3: Variables - <https://aka.ms/SB-Practice03>

Practice 4: More Variables - <https://aka.ms/SB-Practice04>

Practice 5: Conditionals - <https://aka.ms/SB-Practice05>

Practice 6: While Loops - <https://aka.ms/SB-Practice06>

Practice 7: For Loops - <https://aka.ms/SB-Practice07>

Practice 8: Subroutines - <https://aka.ms/SB-Practice08>

Practice 9: Events - <https://aka.ms/SB-Practice09>

Practice 10: Arrays - <https://aka.ms/SB-Practice10>

Practice 11: Math - <https://aka.ms/SB-Practice11>

Practice 12: Advanced Math - <https://aka.ms/SB-Practice12>

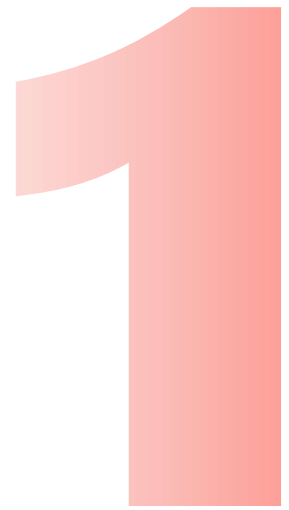
Practice 13: Images - <https://aka.ms/SB-Practice13>

Practice 14: Turtle Graphics - <https://aka.ms/SB-Practice14>



PRACTICE 1: DISPLAYING TEXT





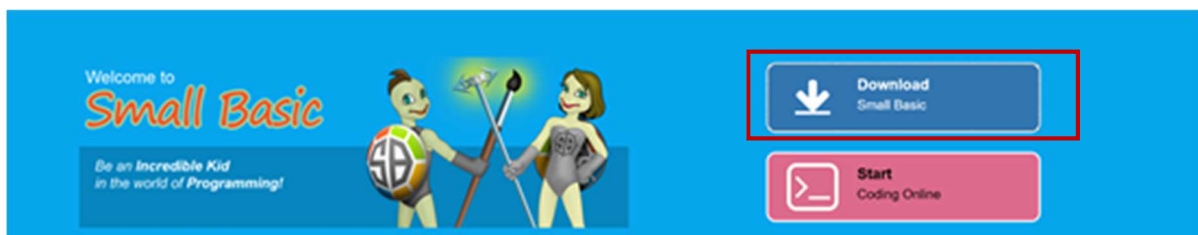
Hello World!

Welcome to your first lesson! In this lesson, you'll get an introduction to a programming language named Small Basic. Just like there are different spoken languages for different countries like English or Spanish, computers have different languages for different tasks as well. You can also think of a language as a tool, like a hammer or a wrench. You'll need to use different tools for different purposes.

There are block-based programming languages, like Scratch and Code.org, where you drag and drop blocks of code. You might use a block-based language to learn programming concepts. Small Basic is a text-based language that exists to make it as easy as possible to do text-based coding. Small Basic is more complex than a block programming language, but it's not as complex as Java or C++, which are used in more advanced classes and by professional programmers.

Downloading Small Basic

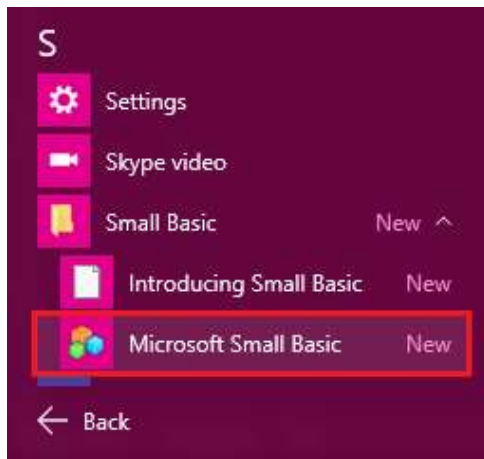
First, download Small Basic from smallbasic.com. Click the **Download** button and run the installer. You'll need a Windows computer or tablet. If you don't have one, click **Start Coding Online** to use the online version of Small Basic, instead of installing it.



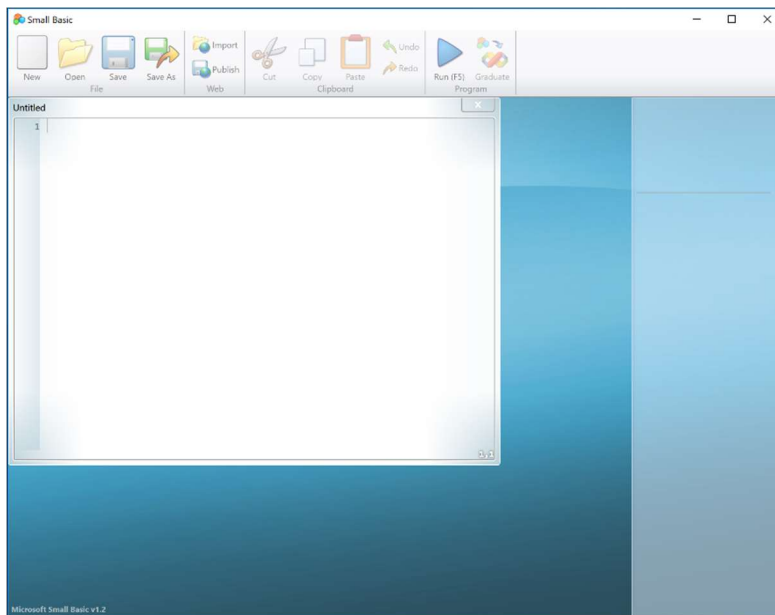
Follow the instructions to install Small Basic. Then, look on your Start Menu for Microsoft Small Basic.



1



Running Small Basic should open a screen like this:



You can type your code into the window (marked Untitled) and press the Run button to run your code. Or press F5 on your keyboard.

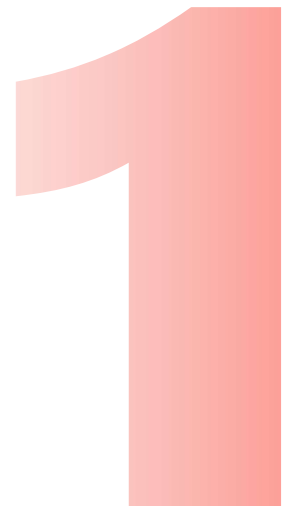
Let's try writing your first program. Type the following into your editor and press the Run button:

```
TextWindow.WriteLine("Hello World!")
```

You should get a window that has the following:

```
Hello World!  
Press any key to continue...
```





If you got an error, look and see if your code looks like this:

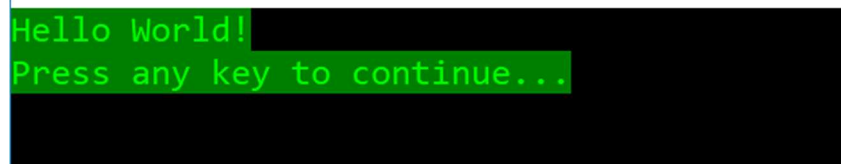
```
TextWindow.WriteLine('Hello World!')
```

The code in the parentheses is green because two single quotes were used instead of a double quote. Text needs to be enclosed in double quotes, which are made by pressing shift and the single quote key at the same time. A single quote is used to add comments to your code and will turn your code green. To Small Basic, it's like that commented out code doesn't exist. That's why you'd get an error. Make sure you use the double quote for strings like this. We'll talk more about using comments in a later lesson.

Now let's make things more colorful:

```
TextWindow.BackgroundColor="darkgreen"  
TextWindow.ForegroundColor="green"  
TextWindow.WriteLine("Hello World!")
```

A program is a series of instructions. The computer first sets the background of text to be dark green, sets the foreground of text to be green, and then prints *Hello World!* to your window. It looks like this:



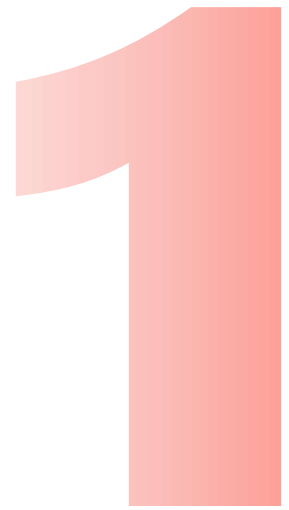
Challenge: Text Window Color

Try changing the code above so that the text is yellow, and the background is dark gray.

Next, try changing the code above so that it uses your favorite colors. There are 16 colors available, as shown in the chart below: Black, DarkBlue, DarkGreen, DarkCyan, DarkRed, DarkMagenta, DarkYellow, DarkGray, Gray, Blue, Green, Cyan, Red, Magenta, Yellow, and White. The colors aren't case sensitive, which means you can type them in all lower case, upper case, or any combination of casing (so it could be BLACK, Black, black, or even BLack). However, you can't include extra spaces.



	0: Black		8: DarkGray
	1: DarkBlue		9: Blue
	2: DarkGreen		10: Green
	3: DarkCyan		11: Cyan
	4: DarkRed		12: Red
	5: DarkMagenta		13: Magenta
	6: DarkYellow		14: Yellow
	7: Gray		15: White



Graphics

Now let's try making some shapes and text:

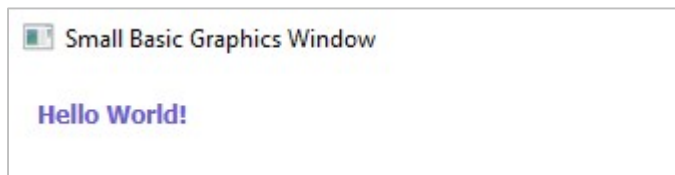
```
GraphicsWindow.DrawText(15,15,"Hello World!")
```

'The two numbers are just an example; you are free to choose where you would like your text to start on the X,Y axis



Note: The green text is a comment. See below for more info.

You should get a graphics window that looks like this:



Let's look a little closer at the code: did you notice the two numbers you added before "Hello World!"? Those numbers indicate where you want the text displayed on your screen. The first number represents the x-axis, or where you want it displayed horizontally. The second number represents the y-axis, or where you want the text to appear vertically. Have you made graphs in math class before? This may be familiar then! If not, we'll talk about this a little more in depth in the next chapter.

Now let's add some color, a shape, and increase the font size. Feel free to add the lines of code one by one to see what each of them does. Type in each line of code, one at a time, and then run your program to see what happens:

```
GraphicsWindow.BackgroundColor="#AFEEEE"
```





Note: Try adding the code line by line

```
GraphicsWindow.BrushColor="#8B0000"  
GraphicsWindow.PenColor="#FF8C00"  
GraphicsWindow.FontSize=30 'This is just a  
sample number  
GraphicsWindow.DrawRectangle(18,25,190,30)  
'These four numbers represent: X axis,Y axis,  
width of the rectangle, and height of the  
rectangle  
GraphicsWindow.DrawText(20,20,"Hello World!")  
'The two numbers are just an example; you are  
free to choose where you would like your text to start on  
the X,Y axis
```



You should get a graphics window that looks like this:

 Small Basic Graphics Window



Hello World!

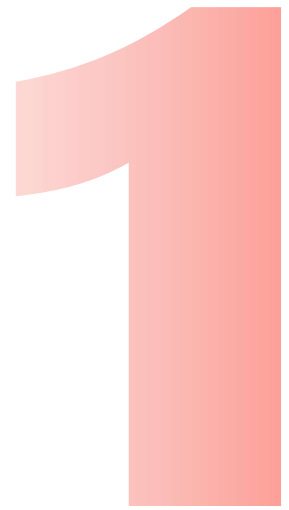
Let's look a little closer at the code: in the first three lines, it looks like there are random characters after a # where you might expect to see color names. Remember how we talked about how you can set colors with names, like "DarkBlue" but there are only 16 options? What if you could get a whole lot more? Using color codes like the above give you many more options! These are called "hex" or "hexadecimal" color codes and are used by many different programming languages and even computer programs to set colors. You can find the table of colors with the respective hex value code in [this](#) article.

There's also a third way to set colors in Small Basic, by using the Red, Green and Blue components (RGB). These values go from 0 to 255 for each

You can set colors in Small Basic in three ways:

- Using pre-defined color names:
`GraphicsWindow.BrushColor = "Blue"`
- Using a Hex value code:
`GraphicsWindow.BrushColor = "#FF0000"`
- Setting the Red, Green, Blue components (RGB):
`GraphicsWindow.BrushColor =
GraphicsWindow.GetColorFromRGB(255,0,0)`





The computer first sets the background to light blue, the brush color sets the color of the text, and the pen color sets the color of the rectangle lines. The rectangle is set by the measures given by you. Lastly, the text starts on the X, Y axis that you specified. Try playing with these numbers to see the effects.

Comments

Did you notice that none of the green text showed up on the graphics window? That's because it's a comment. Comments are made by putting a single quote (') in front of text. The computer then knows to skip that and continue to the next line of code!

So what do you use comments for? Professional programmers use them all the time to explain their code to one another. Just like people write differently, people also code differently, and this is one way to explain what you're doing. It's also a good reminder when you come back to some code you haven't worked on in a while!

You can also use comments to build a framework for your code, even before you write it! This helps break down a complex task into smaller tasks and helps you make a plan, so that you're solving lots of little problems instead of one big one—it may be easy to say "I want a blue screen with an orange rectangle around the words "Hello World" in maroon," but hard to do once you sit down to write the code. Here's an example of how you can use comments to break down and explain the work you want the computer to do, using the code above:

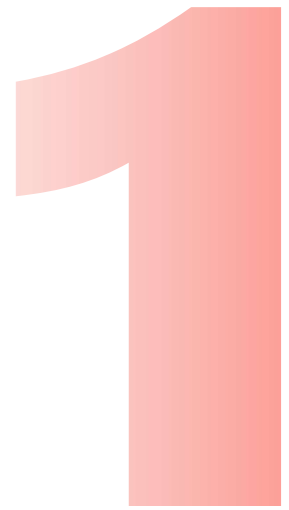
```
'Set the background color of the graphics window
GraphicsWindow.BackgroundColor="#AFEEEE"
'Set the brush color for the text
GraphicsWindow.BrushColor="#8B0000"
'Set the text color
GraphicsWindow.PenColor="#FF8C00"
'Set how big the text will be
GraphicsWindow.FontSize=30 'This is just a sample number
'Code to draw the rectangle around the text
GraphicsWindow.DrawRectangle(18,25,190,30) 'These four
numbers represent: X axis,Y axis, width of the rectangle,
and height of the rectangle
'Code to write the text within the rectangle
```



Note: Try adding the code line by line



```
GraphicsWindow.DrawText(20,20,"Hello World!")  
'The two numbers are just an example; you are  
free to choose where you would like your text  
to start on the X,Y axis
```



Challenge: Graphics Window Color

- Try changing the code above so that the text is green, the background is dark green, and the rectangle lines are blue.
- Try changing the code font and rectangle size.
- Try changing the shape to use an ellipse instead of a rectangle.



Hint: Use the 'DrawEllipse' object

Discussion Questions

Now that you have written your first lines of code, what was your favorite part?

- Did writing comments help you write your code?
- Did you expect to use math? Does finding out there's a use for math make a difference in how you feel about your math classes? Why or why not?
- How might you use different colors to help convey meaning for your text and shapes displayed on the graphics window?
- Language syntax is important for each language (computer and human languages alike). Just like you saw how using a double quote has a different meaning than using a single quote in Small Basic, what are some other examples of language syntax for Small Basic that you have learned so far? Can you think of any examples of specific syntax that you use in the language you speak and write in every day?

Additional Resources

- [Introduction to Small Basic](https://aka.ms/SmallBasicIntro)
 - <https://aka.ms/SmallBasicIntro>
- [Small Basic Color](https://aka.ms/SBColors)
 - <https://aka.ms/SBColors>
- [Small Basic: Supported Colors](https://aka.ms/SBSupportedColors)
 - <https://aka.ms/SBSupportedColors>



1

