

1.4 Editing Programs with IDLE

IDLE will format programs for you in ways that make them easy for both you and the Python system to understand, but you have to let it do its work. Here are a few tips:

- Be careful with horizontal spaces. As we will see in the next few chapters, Python is very particular about indentation; the way a program is indented affects the way it runs. If two lines are supposed to have the same indentation, as in the last two lines of

```

if x < 10:
    result = "small"
    print( "That number is small." )

```

it is not enough to make them look the same on the screen – their indentations need to consist of the same characters (either spaces or tabs). I recommend using the `tab` key for all indentation.

- Be generous with vertical spaces. Python ignores blank lines in your code. It helps the reader of your programs (especially you) if you use blank lines to separate the functional parts of your programs. As our programs become more complex, they will consist of more and more short functions, each of which is independent of the others. Use an extra blank line to separate functions. That helps to make the structure of the program easier to understand, and the overall program easier to read. As we will say many times, a program that is easy to read is much more likely to be correct than one that is hard to read.
- Beware of long lines. Because Python uses indentation as a functional part of the language, complex portions of code tend to be shifted to the right in the IDLE window. For some reason the designers of IDLE did not give it a horizontal scrollbar. If you have a line of code that extends to the right of your IDLE window, you must put the cursor on this line and use the right arrow key to move the cursor to see the rightmost portions of the line. This is both awkward and mistake-prone. You can split the line, but you must be careful how you do it because white space is meaningful in Python. If you end a line, Python will probably think you are ending the statement it contains and, if this isn't a complete statement, give you an error message. To avoid this, put a `\` character at the end of the line you will continue. To the system this says "Treat the next line as a continuation of the current one." For example,

```

print( "Dear %s. Please send $%d" to me soon." \
      %( "Mom" , 20) )

```

Without the `\` character at the end of the first line this code would not work.

- Write programs non-linearly. Only the very simplest programs are written from top to bottom. Usually I write a small portion of code, then see what supporting code it needs and write that above the original chunk of code. Over the course of writing a complete program I tend to move all over the file. IDLE gives you a nice editor that makes it easy to move around the file; make use of it.
- Write small chunks of code at a time. Beginners want to write a complete program and then have a horrible time trying to debug it. More experienced programmers write a bit of a program, test it out, then write another piece of the program and test it out and repeat this until the program is finished. In an integrated system like IDLE and the Python shell, it takes only a few seconds to test a program and if the new code is only a few lines it can't take long to debug. When you are in a hurry to be done it always feels like it will be faster to just write all of the code, but in the end it will be slower and more frustrating to do it this way. Programming is one area where taking your time and being careful and methodical actually save you time.

IDLE colors

The IDLE editor makes programs colorful. This is not just for the artistic value; the colors are meaningful. Here is how the colors are used by default; if you wish you can change the color assignments in the **Configure IDLE** option of the **Options** menu of IDLE.

orange Keywords: words that are meaningful in Python, such as "if", "def", and "while".

green Strings: text in quotes, such as "bob".

red Comments: text for the human reader of the program that ignored by the computer.

blue Names of things being defined.