

## Things to Know (Info) and Errata

Introduction to Computing and Programming in Python, A Multimedia Approach, Second Edition

### Chapter 5

Info	Pg 108	<p>The general form of the if statement is not as clear as it could be. The comment goes into the second line and “is true” is not a block of commands but a continuation of the comment. (See the symbol in Program 42 used to show continuation on the next line and the “Making It Work Tip” on page 118 in the text for an explanation – it should have been noted here, too.) The code might be more clear if it was something like the following:</p> <pre>if (some test):     # The Python statements in the following block are executed if test is true     statement1     statement2     statement3 #The if statement block has completed above and the following line is # executed whether the test was true or not print “This will print anyway.”</pre>
Error/Info	Pg 110 & 111	<p>In both Programs 36 and 37, the blueness and greenness has been reversed in the setColor function and there is no need to do this to increase the redness. The last bullet under How It Works has blue before green – the explanation is correct (blue and green not changing) but mirrors this reversal of colors. Note that the second to last bullet has the green between red and blue.</p> <p>The use of int() in the calculation of redness is unnecessary as getRed(px) *2 is already an integer value.</p> <p>By now, there is no need to make the picture inside the function and then return it – we will usually make the picture outside the function and pass it in as a parameter of the function unless we are making several pictures and creating a collage.</p>
Error/Info	Pg 111 & 112	<p>Program 38 has the last parameter on the second line – it needs to be with the other parameters in the “def” line. (See the symbol in Program 42 used to show this and the “Making It Work Tip” on page 118 in the text for an explanation – it should have been noted here, too.)</p> <p>There is no need to make the color red in the third line of Program 38 – red is already defined in JES as (255, 0, 0) for its component colors (see page 72). This line of code and the second to last bullet on page 112 under How It Works are unnecessary.</p>
Info/Error	Pg 112	<p>In the function call of reduceRedEye (3<sup>rd</sup> line of &gt;&gt;&gt;), there is no need to use makeColor(0, 0, 0) – black is already defined – see page 72.</p> <p>The last paragraph above How It Works says that the eye has all black pixels – in reality, the eye’s pupil now has all black pixels.</p>

These errata and informational notes were identified at the United States Military Academy, West Point, NY. Any corrections should be forwarded to the IT105 (Introduction to Computing and Information Technology) Course Director. (Last updated: 25 Aug 2013)

## Things to Know (Info) and Errata

### Introduction to Computing and Programming in Python, A Multimedia Approach, Second Edition Chapter 5

		First bullet, last word in the second line should be “pictures” not programs.
Info	Pg 114	Note that the author uses <code>grayScaleNew</code> in the 3 <sup>rd</sup> line of Program 39 which is Program 18 on pg 70 and not <code>grayScale</code> which is Program 17 on pg 69.  In Program 39, there is no explanation why the author chose to use <code>red &lt; 63</code> (which includes 62 and not 63) and <code>red &gt; 62</code> (includes 63 and not 62) instead of <code>&lt;= 62</code> and <code>&gt;=63</code> . (This comment also applies to Program 40 on pg 116.)  There is also no explanation why <code>green</code> is not changed at all but as with many of the examples in this text, the programs reflect the experimentation of the author to achieve a desired effect (see last line of last paragraph on pg 114).
Info	Pg 116	In the second through fourth if statements for each color in Program 40, the “and” operator is used in the test condition. It requires that both inequalities are true for the test to be true.
Info	Pg 117	Assume that the choice of 64 for luminance (to determine which pixels become black or white) in Program 41 is based on the author’s testing to determine the best choice. (Try changing it to see what happens!)
Info	Pg 118	Note that in Program 42 that the parameter is the filename (and not a picture) – this facilitates making two picture objects from the same file. This is also done in Program 43 on pg 120. In both cases, the program requires the use of the return statement to get back the created picture. (The show of the picture is not necessary.)  Also, this is the first time the author uses the symbol to show a line of code in the program that only goes to the next line due to the formatting of the textbook. (See the Making It Work Tip.)
Info	Pg 118-119 (not covered in IT105)	In Program 42, the two <i>for</i> loops have the wrong range. The intent is to start and end one pixel from the edges of the picture in both the x and y directions to accommodate the averaging of pixel color component values (see figure 5.9). The ranges should be modified to subtract 1 and not 2 in each <i>for</i> loop. The loops should read:  <pre>for x in range(1, getWidth(source)-1):     for y in range(1, getWidth(source)-1):</pre> The ranges end at 1 pixel away from the edge and allows calculations of the edge pixels. At the top of pg 119, the text says the “x index values goes from 1 to the width-1.”
Info	Pg 120 (not	Note that the values for the components of the “center” pixel could be from either the current source or the target pixel because neither has been modified

These errata and informational notes were identified at the United States Military Academy, West Point, NY. Any corrections should be forwarded to the IT105 (Introduction to Computing and Information Technology) Course Director. (Last updated: 25 Aug 2013)

## Things to Know (Info) and Errata

Introduction to Computing and Programming in Python, A Multimedia Approach, Second Edition  
Chapter 5

	covered in IT105)	yet and can be used in the calculation. Using the target pixel facilitates the assignment of the calculated new color to the center pixel in the target picture.
Info	Pg 121 (not covered in IT105)	In the second if statements in Program 43, the “or” operator is used in the test condition. It requires that at least one of the inequalities is true for the test to be true. It is possible that neither will be true.
Info	Pg 121 (not covered in IT105)	In the first bullet, the call of the function should be on one line: <code style="text-align: center;">bwVersion=lineDectect(pickAFile())</code>
Info	Pg 122 (not covered in IT105)	Program 44 should be called lineDetect2 since the code is different than Program 43 on pg 120.  Program 44 introduces the “not” operator which negates an expression changing a true test to a false or a false to a true. In going from Program 43 to Program 44, it uses the mathematical concept: <b>not(A and B) = not A or not B</b> Program 43 checks for <code>(abs(...)&gt;10 and abs(...)&gt;10)</code> in the first if. That is the same test used in the isLine function in Program 44. Taking the <b>not</b> of this test (as in the second if in Program 44), changes the <b>and</b> to an <b>or</b> and negates the two <code>(abs(...)&gt; 10)</code> inequalities. The <b>not</b> of an expression using <code>&gt;</code> changes it to use <code>&lt;=</code> which is the same as the second if statement in Program 43. (This is a little difficult to follow but check it out!)
Error	Pg 125 (not covered in IT105)	The if statement in Program 46 that includes the distance function should all be on one line. (See the symbol in Program 42 used to show this and the “Making It Work Tip” on page 118 in the text for an explanation – it should have been noted here, too.) The return statement is also not needed.
Error	Pg 128 (not covered in IT105)	The three comments in Program 47 should all be on one line unless a second # is used on the second line of each comment. There is also no need to return source since the source picture is used as a parameter.
Error/Info	Pg 130	In Program 48, the chromakey2 example has program as a parameter – it should be source: <code style="text-align: center;">def chromakey2(source, bg):</code> There is also no need to return source since the source picture is used as a parameter.

These errata and informational notes were identified at the United States Military Academy, West Point, NY. Any corrections should be forwarded to the IT105 (Introduction to Computing and Information Technology) Course Director. (Last updated: 25 Aug 2013)

## Things to Know (Info) and Errata

Introduction to Computing and Programming in Python, A Multimedia Approach, Second Edition  
Chapter 5

Info	Pg 132	In the function <code>horizontalLines</code> at the top of the page, the code works correctly. However, the <code>x</code> and <code>y</code> variables are used to represent what are normally the <code>y</code> and <code>x</code> values of the pixel location – note that <code>x</code> is used with height in the first for loop, <code>y</code> is used with width in the second for loop, and <code>setColor</code> uses <code>(y,x)</code> which corresponds to the loops but goes against our intuition.
Error/Info	Pg 132	The first paragraph talks about using the predefined colors here – this is not done throughout the text and ought to be.  Program 50 talks about using “beach-smaller.jpg” but in the provided folder of images from the author (called <code>mediaSources</code> by the author and IT105 calls it <code>textbookMedia\PictureFiles</code> folder), there is only a <code>beach.jpg</code> and the coordinates for placing the rectangle on the beach do not do so – it ends up in the sky. Use <code>(350, 350, 50, 50)</code> instead.
Info	Pg 134	Note that Program 51 takes no parameters but creates the canvas, draws on it, and then returns the canvas – the canvas must be returned because it was created inside the function. In this program, the author also chooses to not display the canvas using <code>show</code> or <code>explore</code> . The returned picture can be shown or explored at the Command Area prompt.
Info	Pg 136	Program 52 should be <code>grayPosterize2</code> to distinguish it from Program 41 on pg 117 because the code has changed.  This rewrite of Program 41 to use “else” is very powerful and used extensively by programmers. See the two paragraphs above the start of section 5.9.  The explanation of <code>elif</code> may be a little confusing. What the author is trying to say is: <b>if</b> the condition is true, then choose the first option, else if ( <b>elif</b> ) a different condition is true, choose the second option, <b>else</b> choose the third option.  The concepts of using “if”, “else”, and “elif” are not listed at the end of Chapter 5 summary. See the bottom of Appendix A.3 on pg 409, Problem 9.1 on pg 239, and under Functional Programming on page 373 for more examples.
Info	Pg 137	The author uses comments that talk about increasing grayness. In reality, more white is being added to the color (higher numbers for color components) in the second and fourth parts of Program 53. Also, in the first and fourth comments, <code>100-gray</code> is not used as subtraction but the author is saying that the gray used has all three component values of 100.

These errata and informational notes were identified at the United States Military Academy, West Point, NY. Any corrections should be forwarded to the IT105 (Introduction to Computing and Information Technology) Course Director. (Last updated: 25 Aug 2013)