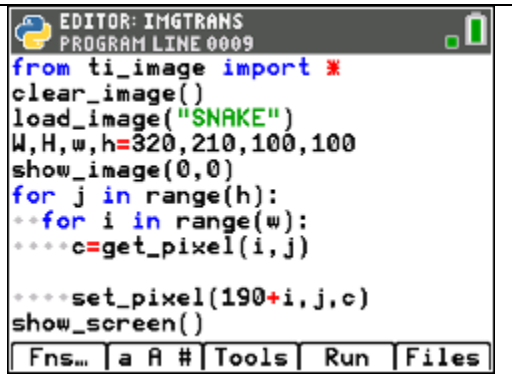
TI Image: Image Processing **Transformations**

This activity builds on the grayscale activity from Getting Started with Image Processing. You will perform image transformations like flipping, mirroring and rotating an image.

---

0. First make a copy of your **grayscale** activity that you made in the "Getting Started…" activity. To copy a program, select **<Files>**, point to the desired program, select **<Manage> Replicate program** and give the new program a name. We named it **IMGTRANS.**

   If you did not complete that activity, this screen shows most of the code that was created. Enter this code into a python program.

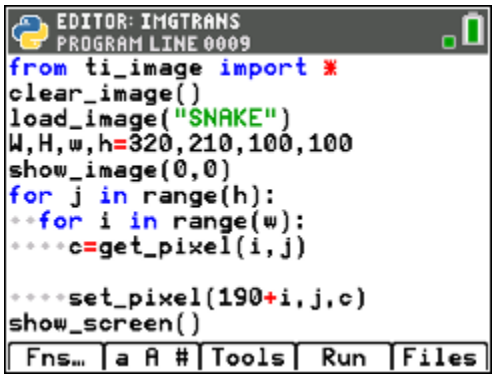


1. This image of a snake (python) is used in this activity again. Recall that the image is **100x100** pixels. If you do not have this image file on your calculator you can download it from SNAKE.8xv. This special AppVar contains an image and can be transferred to a TI-84 Plus CE Python calculator using your TI-Connect CE software (this *free* software is available here).

2. Here's the code again. We will keep almost all the GRAYSCAL code but notice that the average **a=** … statement has been removed and **set_pixel( )** uses the tuple variable **c** for the color.

   Our **goal** in this part of the activity is to *move* the pixels, not change their color. We will flip the image vertically, so that the snake is upside down and the tongue is still pointing to the left.

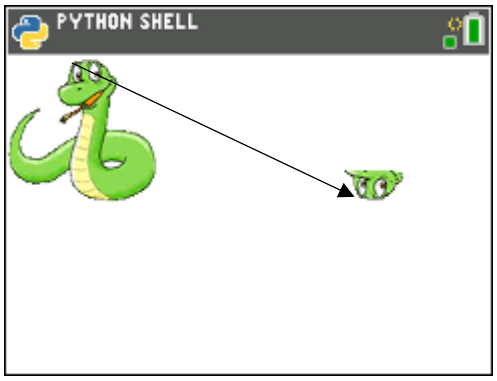
   Just modify the **set_pixel( )** statement… Try it yourself!



3. The pixels in the top row move to the bottom row. As the row numbers increase in the original image, they will *decrease* in the flipped image:

   Row j → Row (h-1) – j

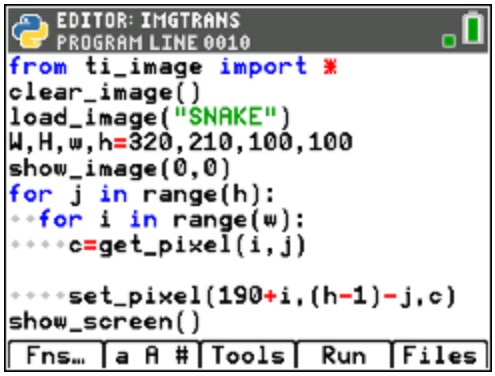
   Why (h – 1)? Remember that since the top row is row # 0 then the bottom row is row # 99.



4. Change the **set_pixel( )** statement:

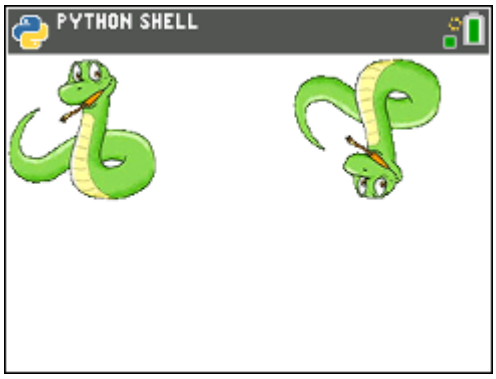
   set_pixel(190 + i, (h - 1) – j, c)

   190 + i makes the image appear on the right side of the screen.
   c is the color *tuple* from the original image.



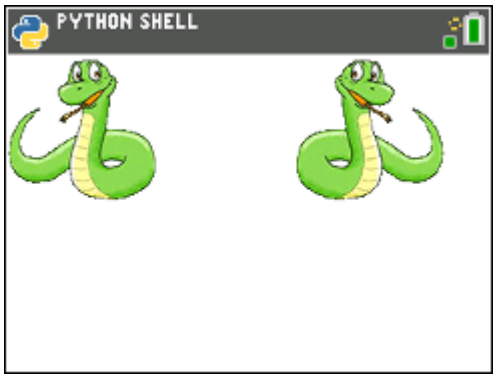
5. **<Run>** the program. Soon your flipped image is complete! Other transformations include mirroring (flipping left to right) and rotating (90, 180, and 270 degrees).

   The next part of this activity demonstrates mirroring…



6. **Mirroring** the image is similar to flipping. We just need to determine the proper position of each new pixel, so the only change to the code will be in the **set_pixel( )** coordinate expressions.

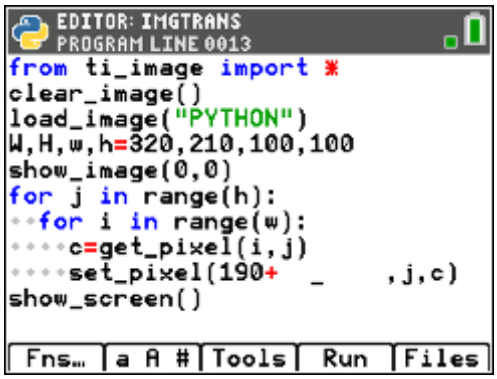
   Try it yourself now and remember to keep the new image on the top right side of the screen.



7. Leave **190+** in the x-coordinate but change the **i**.

   Row **j** in the original goes to row **j** in the copy.
   In each row, pixel **i** from the left moves to pixel i *from the right* and the right edge is **(w - 1)**, since the left edge is 0.

   Try again!

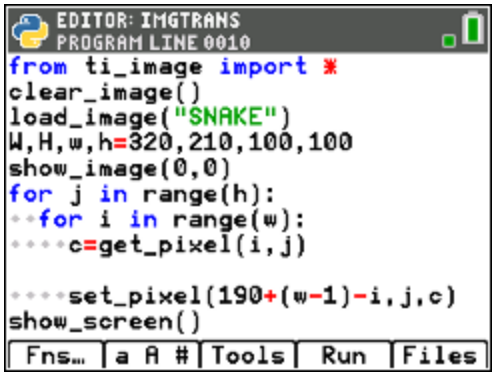


8. The statement is:

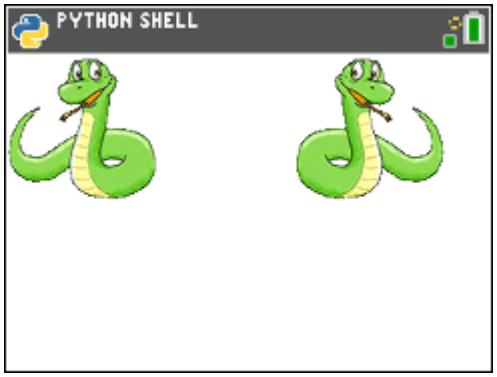
set_pixel(190+ (w – 1) - i), j, c)

   190+ is used to place the image on the right side of the screen
   (w – 1) is the right edge of the new image
   - **i** is i pixels from the left edge of the *original* image
   - **j** is the row number from the *original* image
   - **c** is the color tuple from the original image

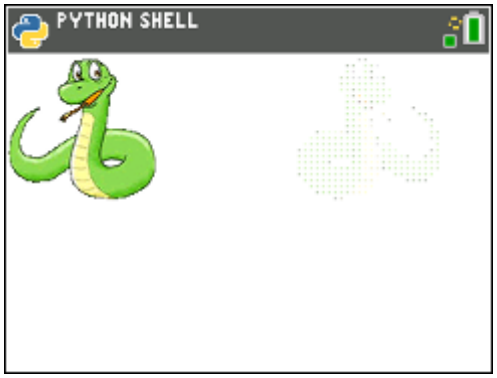
so, **190+ (w – 1) – i** is the **i**th pixel *from the right edge* of the new image!



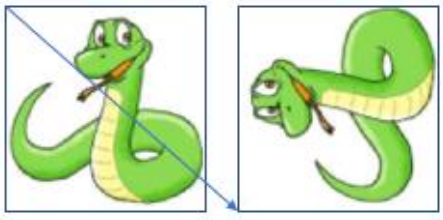
9. **<Run>** the program to see the mirrored image eventually appear! Please be patient.



Challenge 1: You can get an idea that your algorithm is working properly more quickly by plotting fewer pixels. Modify your program to plot every second, third or fourth pixel in the image instead of all of them.



Challenge 2: Rotate



Use the same image processing technique to *rotate* the image.
Hint: row **i** from the top of the original image moves to **column i** (from the left) in the new image working from *bottom* to *top*.

row col row col
Pixel **(i, j) → (j, (h – 1) - i)**.

This will rotate the image **90 degrees counter-clockwise**. You can also rotate 180 degrees (that's not the same as flipping!) or 90 degrees clockwise.

Teacher Tip:

SNAKE.8xv can also be loaded into the teacher software TI_SmartView CE using the **Explorer** workspace.

Be sure to keep your software up-to-date!