SOUND OFF !!

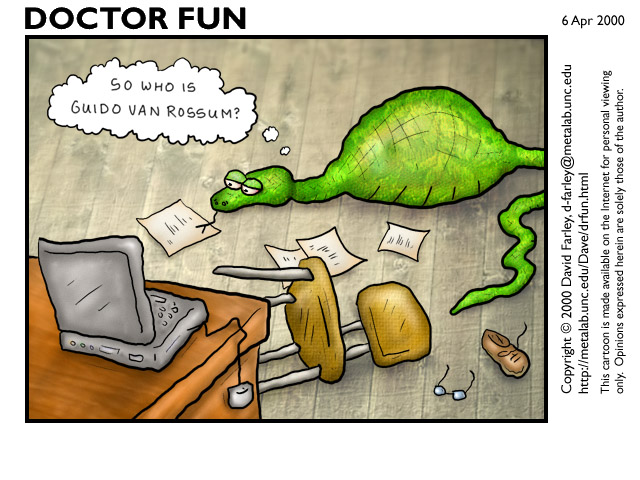
Lab 6

**What You’ll Learn in this Lab:**

* A better understanding of how sound works
* A more complete understanding of how sound is stored in the computer
* How to splice sounds
* To get more experience with “for” loops, “if” statements, and arrays in Python

**Getting Ready:**

* Read Chapter 7 in the Python textbook
* Bring the following to lab:
  + Lab 5 Worksheet, ready to turn in at beginning of lab!!
  + CSIS-110 Lab Manual
  + Python textbook
  + Something to write with
  + The Lab 6 Worksheet



**Details**

*This section of each lab will involve practicing and experimenting with various aspects of our computer systems – editing, printing, mail, techniques for making your life easier, and saving your projects.*



***Who is Guido van Rossum (hint: see cartoon on page 1)?***

As you saw with images, there are many sounds available for you to “play with” in the Media Sources folder. But it’s also sometimes fun to create your own sounds. In this section of the lab, both of you will record a short sentence that you will use later in the lab.

On the PC in front of you, double-click on the icon labeled “Audacity”. This is a sound recording program that produces “.wav” sound files. At this point, see your instructor to get a USB microphone and plug the microphone into your computer and turn it on (a green light will show when the microphone is on).

Sound recording tips:

* **Make sure you speak directly into the mic**
* **Enunciate clearly**
* **Don’t run your words together**

Using Audacity, each of you should record yourselves saying **the same** 4-word sentence (keep it clean, please!). It’s best if you make up your own, but if you’re not sure what sentence to use, here’s a suggestion: “Python is so cool!” Click the red button to record and the square tan button to stop.

**IF THE SOUND DOESN’T SEEM TO BE RECORDING: Go to Control Panel, Hardware & Sound, Sound, Manage Audio Devices, Recording, and set the USB microphone as the default recording device.**

If you don’t like the first try, just try again (you can use the recorder to play back the most recent version) – once you’re satisfied, save the sentences as “sentence1” and “sentence2” in your CSIS110 folder (not on your desktop). To save a sound use the “File” menu’s “Export” option.

Now it’s time to get those sounds into Python and explore them a bit. Start JES, and type the following commands (choose the file you called “sentence1”):

**s1 = makeSound (pickAFile())**

**explore (s1)**

***Using the sound explorer, complete the first row of the table on the worksheet with the proper “Current Index” numbers, then repeat the above two commands for “sentence2” and fill in the 2nd row of the table. You will use this information later. Make sure you use the actual starting/ending points for each word; don’t include any “gaps” between the words.***



***What is the sampling rate for “sentence2.wav”? How many samples are in it? How long is it, in seconds?***

**Enhancing your Computer Science Knowledge**

*This section of each lab will involve problems and software that will augment your understanding of concepts that are fundamental to Computer Science.*

**Copy** the file “lab6Enhancing.py” from the proper folder in the common area into your own CSIS110 lab area. **Open** the file in JES. Type the following commands to test the first function in the file:

**increaseVolume (s1)**

**play(s1)**

You should hear a louder version of one of your sentences than you heard before. Note that this actually changed s1 to be louder (but it did not change the file on disk). Type the following command, choosing “sentence1.wav” again, to restore s1 to the original sound:

**s1 = makeSound (pickAFile())**

You will now modify “increaseVolume” as follows: for positive samples (sample value is larger than 0), double the sample value. For negative sample values, set the value to the most negative possible value *(Hint: see page 230 in your textbook for sample value range)*. Edit, save, load, and test until working, and don’t forget to “restore” your s1 variable (using the above command) after each test so you’re working with the original version of the sound!



***Describe the difference(s) you hear in the sound now. Can you still make out the words?***



***Use “explore” to examine s1. How does the graph look different from the original?***



***Call your instructor over to hear the results of your new “increaseVolume”, and make sure he/she signs on the line on the worksheet.***



***Print a copy of “lab6Enhancing.py” and attach it to your worksheet before you turn it in.***

**Extending and Expanding:**

*This section of each lab will involve using Python to answer questions and solve problems.*

Now we will work with splicing sounds. Copy the file “lab6Expanding.py” from the common area to your CSIS110 folder. Open your copy of this file in JES.

For your next task in this lab, you will create a new sound that is built from the words in your recorded sentences. The final sound should be the 4 words of your sentence, with the first and third words coming from “sentence1.wav” and the second and fourth words coming from “sentence2.wav”. Here are a few things that will make this much easier: (1) There are 2 useful functions, “clip” and “copy”, in “lab6Expanding.py”. Check ‘em out! (2) Program 107 on page 246 of the Python text has some examples of using those 2 functions. (3) The table of numbers you created in the “Details” section.

Write a function called “sentence()” that will create and return the merged sentence as described in the last paragraph. It should use “clip” and “copy” four times each. If you think your words are too close together (or if you think that your lab instructor will think so), you should fix things to add some space in between the words. The easiest way to include silence is to simply start the next sound later (i.e., “skip” some samples in the target sound by not copying anything into them).



***Call your instructor over to hear the results of your new sentence, and make sure he/she signs on the line on the worksheet.***

You will now add another function to the “lab6Expanding.py” file. This function will take one parameter, a sound, and will lower and raise the volume in portions of the sound. Specifically, it should alter the sound’s volume as follows: first quarter of sound a tenth the original volume, second quarter double original volume, third quarter a tenth the original volume, and last quarter double original volume.



***Call your instructor over to hear the results of your new function applied to the sentence you created, and make sure he/she signs on the line on the worksheet.***



***Print a copy of “lab6Expanding.py” and attach it to your worksheet before you turn it in.***

One more task in today’s lab! **In the command area only**, add ½ second of one of the “bassoon” notes (from the Media Sources folder) starting in the middle of another, longer sound. (Don’t make the sound any longer, just copy the ½ second of bassoon over ½ second of the longer sound.) You will need to use the “makeSound (pickAFile())”, “clip”, “copy”, and “play” functions.

***In the space on the worksheet, write the sequence of commands you used in the command area to accomplish this.***



***Call your instructor over to hear your new sound, and make sure he/she signs on the line on the worksheet.***



**Reflection:**

*The discussion questions in this section of each lab are meant to make you think critically and creatively about some of the things you did earlier in the lab. Your answers to these questions must not be written on the lab worksheet, but on separate sheets of paper attached to the end of your lab worksheet. Your answers must be typed (not handwritten), and you will be graded on all aspects of your answer (correctness, use of proper terminology, readability, use of complete sentences only, etc.). In general you are expected to write at least one or two paragraphs in answer to each question.*

Given what you’ve been able to do so far in this course with both images and sounds, how hard do you think it would be to take an audio recording of someone speaking and create new sentences that make it seem like they are saying something entirely different? What parts of this task would be easy? Which parts would be difficult? How realistic do you think you could make it sound? How ethical would this be?



Find an example of a speech like this on the Internet and cite your source. Why was this speech created? How do you know it’s fake? Is there anything about the speech that makes it sound fake? If so, describe it, relating it to what you know about sound processing in Python.