

Homework 0

1 Introduction

This assignment gets you started with the basic tools you will need to complete all of your homework projects. This project will

- ensure that you have correctly installed Python 3
- give you practice using a text editor to write Python programs, and
- give you practice running Python programs and using command line features.

2 Problem Description

You are a CS 2316 student who needs to install Python, configure it for command line use, and learn how to use a programmer's text editor to create and edit Python source code.

3 Solution Description

1. Download and install Python 3 on your computer and configure your system so you can run Python from the command line. See the instructions on the class web site's Resources page.
2. Download and install a programmer's text editor. You may end up trying out several over the course of the semester before you settle on one. See the Text Editors guide on the class web site's Resources page.
3. Create a directory for your CS 2316 coursework somewhere on your hard disk. We suggest `cs2316`. Note: avoid putting spaces in file and directory names, since doing so complicates the use of some command line tools.
4. Create a `hw0` subdirectory of your CS 2316 coursework directory for your HW0 solution.

On Unix/BASH you can create both of these directories at once with

```
$ mkdir -p cs2316/hw0
```

Note: the `$` is the command prompt (would be something like `C:\>` on Windows), the text after it is what you enter.

5. On the command line, go to the `hw0` directory you just created and enter these commands:

```
$ python3 --version > hw0-output.txt
```

`>` redirects the output of a program, in this case to the `hw0-output.txt` file. Important note: if the line above doesn't write your Python version to the `hw0-output.txt` file then replace the `>` with `2>` and try again. Some versions of Python, such as the one installed by Anaconda and miniconda, write the Python version to `stderr` instead of `stdout`. `>`

redirects `stdout` and `2>` redirects `stderr`. For more information, [this blog post](#) has a nice discussion of the file descriptors `stdin`, `stdout` and `stderr`.

6. Open your text editor, create a file in your newly created `hw0` directory named `nimbly_bimbly.py` and save the following Python program in the file:

```
print("\u004D\u0065\u006F\u0077 " * 9)
print("...")
print("\u004D\u0065\u006F\u0077\u0021")
```

7. In your OS command shell, `cd` to your `hw0` directory and enter `python3 nimbly_bimbly.py` to run the program and see its output on the command line.
8. Add the output of your program to `hw0-output.txt` by running `python3 nimbly_bimbly.py >> hw0-output.txt`. Don't forget the extra `>` in `>>`. `>>` appends to an existing file, a single `>` overwrites an existing file.

4 Turn-in Procedure

Submit your `hw0-output.txt` file on T-Square as an attachment. When you're ready, double-check that you have submitted and not just saved a draft.

5 Verify the Success of Your Submission to T-Square

Practice safe submission! Verify that your HW files were truly submitted correctly, the upload was successful, and that your program runs with no syntax or runtime errors. It is solely your responsibility to turn in your homework and practice this safe submission safeguard.

1. After uploading the files to T-Square you should receive an email from T-Square listing the names of the files that were uploaded and received. If you do not get the confirmation email almost immediately, something is wrong with your HW submission and/or your email. Even receiving the email does not guarantee that you turned in exactly what you intended.
2. After submitting the files to T-Square, return to the Assignment menu option and this homework. It should show the submitted files.
3. Download copies of your submitted files from the T-Square Assignment page placing them in a new folder.
4. Re-run and test the files you downloaded from T-Square to make sure it's what you expect.
5. This procedure helps guard against a few things.
 - (a) It helps insure that you turn in the correct files.
 - (b) It helps you realize if you omit a file or files.¹ (If you do discover that you omitted a file, submit all of your files again, not just the missing one.)
 - (c) Helps find syntax errors or runtime errors that you may have added after you last tested your code.

¹Missing files will not be given any credit, and non-compiling homework solutions will receive few to zero points. Also recall that late homework will not be accepted regardless of excuse. Treat the due date with respect. Do not wait until the last minute!