Try to read Piazza frequently (very helpful Q&A so far)
  - IDLE issues: rename random.py
  - Week 1 files will not appear on Moodle due to the merge

Lab 3: more problem solving and integrating concepts

No Lab 4 due to Rally Day

Start Homework 3 early!

Office hours tomorrow (Thursday)
11am-1pm (Ford 015)
Outline: 2/15

- Feedback from Week 1 (Lab/HW)
- Random colors exercise
- Nested for-loops
- Preview Homework 3
Week 1 Feedback
Lab 1 Feedback

- Pay close attention to variables names, make them descriptive and informative
- Make sure to comment your code
- Duplicate function names (i.e. two “pretty_print(name)” functions)
- Using spaces (not a major issue for now, but always good to develop good habits)
These examples are taken directly from the Python style guide. They are listed under “pet peeves”, so not major issues!

Spaces between function and arguments:

Yes: spam(1)
No: spam (1)

https://www.python.org/dev/peps/pep-0008/
Python style guide on spaces

These examples are taken directly from the Python style guide. They are listed under “pet peeves”, so not major issues!

Spaces between function and arguments:
- **Yes:** spam(1)
- **No:** spam (1)

Spaces during assignment:
- **Yes:**
  - $i = i + 1$
- **No:**
  - $i = i + 1$

https://www.python.org/dev/peps pep-0008/
These examples are taken directly from the Python style guide. They are listed under “pet peeves”, so not major issues!

Spaces between function and arguments:

Yes: spam(1)
No: spam (1)

Spaces during assignment:

Yes: i = i + 1
No: i=i+1

Spaces during indexing:

Yes: spam ham[1], {eggs: 2})
No: spam( ham[ 1 ], { eggs: 2 } )

[https://www.python.org/dev/peps/pep-0008/](https://www.python.org/dev/peps/pep-0008/)
# Python style guide on spaces

These examples are taken directly from the Python style guide. They are listed under “pet peeves”, so not major issues!

## Spaces between function and arguments:

- **Yes:** spam(1)
- **No:** spam (1)

## Spaces during assignment:

<table>
<thead>
<tr>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>i = i + 1</td>
<td>i=i+1</td>
</tr>
</tbody>
</table>

## Spaces during indexing:

- **Yes:** spam(ham[1], {eggs: 2})
- **No:** spam(ham[ 1 ], { eggs: 2 } )

## Spaces with multiple operations:

- **Yes:** c = (a+b) * (a-b)
- **No:** c = (a + b) * (a - b)

[https://www.python.org/dev/peps/pep-0008/](https://www.python.org/dev/peps/pep-0008/)
# CSC 111, Lab 1, Part C
# Author: Eve Xu and Macarena Rojas

def main():
    print("This program will ask you questions :)")
    name = input("Enter your name: ")
    college = input("Enter your college: ")
    year = eval(input("Enter your class year: "))
    major = input("Enter your major: ")

    print("\n" + name + ", class of", year, "goes to", college, "and is studying", major + ".")

main()
# CSC 111 Lab1D
# Authors: Eve Xu and Macarena Rojas

# This program will print 3 names with a pretty border

def pretty_print(name):
    len_name = len(name)
    border = ("+-"+ ("-" * len_name)+"-+)
    print( border, 

    border, "\n" + "|", name, "|", "\n" + border)


def main():
    print("This program will print 3 names with a pretty border :) ")

    name_1 = input("Enter name 1: ")
    name_2 = input("Enter name 2: ")
    name_3 = input("Enter name 3: ")

    # call the pretty print function in a loop
    for name in [name_1, name_2, name_3]:
        pretty_print(name)

main()
Homework 1 Feedback

- More comments and more descriptive variable name choices
  - Grading is based on style as well as correctness

- Make sure your transcript file is plain text

- A few issues with computation and/or lack of testing, but not many
Putting things together from last time (split & random):

random colors
Pick a random color program

1) Ask the user for a list of their favorite colors
2) Colors should be separated by commas (no space)
3) Split colors based on commas (using the split function)
4) Choose a color from the resulting list at random and print it

(Bonus) Allow arbitrary spaces and remove (replace?) before splitting
Pick a random color program

1) Ask the user for a list of their favorite colors
2) Colors should be separated by commas (no space)
3) Split colors based on commas (using the split function)
4) Choose a color from the resulting list at random and print it

(Bonus) Allow arbitrary spaces and remove (replace?) before splitting

Examples:

```python
>>> Enter your favorite colors, separated by commas: pink,red,aqua,blue
pink
>>> Enter your favorite colors, separated by commas: pink,red,aqua,blue
blue
```
Nested for-loops: **INSIDE-OUT** technique
Nested control statements

- Idea: for-loops and conditionals (both types of control statements) can be used like building blocks
- For-loops inside other for-loops, conditionals inside for-loops, for-loops inside conditionals, etc
Nested control statements

- Idea: for-loops and conditionals (both types of control statements) can be used like building blocks.
- For-loops inside other for-loops, conditionals inside for-loops, for-loops inside conditionals, etc.

```python
for <variable1> in <sequence1>:
    <code>
    <code>
    <code>
    <code>
```
Nested control statements

+ Idea: for-loops and conditionals (both types of control statements) can be used like building blocks
+ For-loops inside other for-loops, conditionals inside for-loops, for-loops inside conditionals, etc

for <variable1> in <sequence1>:
  <code>
  for <variable2> in <sequence2>:
    <code>
    <code>
Nested control statements

+ Idea: for-loops and conditionals (both types of control statements) can be used like building blocks
+ For-loops inside other for-loops, conditionals inside for-loops, for-loops inside conditionals, etc

```
for <variable1> in <sequence1>:
  <code>
  for <variable2> in <sequence2>:
    <code>
```

Inside *outer* for-loop

Inside *inner* for-loop
Inside-Out technique

1) Develop and test the inner loop so it works on one example
2) Develop the outer for loop and make variables general
3) Indent the inner loop to the right
Inside-Out technique

1) Develop and test the inner loop so it works on one example
2) Develop the outer for loop and make variables general
3) Indent the inner loop to the right

```python
for <variable2> in <sequence2>:
    <code>
```

step 1
Inside-Out technique

1) Develop and test the inner loop so it works on one example
2) Develop the outer for loop and make variables general
3) Indent the inner loop to the right
   for <variable1> in <sequence1>:
     <code>
     for <variable2> in <sequence2>:
       <code>
     <code>
   <code>
Inside-Out technique

1) Develop and test the inner loop so it works on one example
2) Develop the outer for loop and make variables general
3) Indent the inner loop to the right

```
for <variable1> in <sequence1>:
  for <variable2> in <sequence2>:
    <code>
  <code>
```
Example: guess my number

+ Goal: the user picks a number and the computer tries to guess it

+ The computer gets 10 tries

+ After that, onto the next round

>>> How many times to do you want to play? 3

Enter a number between 0 and 9 inclusive: 3
2 is wrong, guess again.
8 is wrong, guess again.
9 is wrong, guess again.
8 is wrong, guess again.
You guessed my number!
8 is wrong, guess again.
8 is wrong, guess again.
8 is wrong, guess again.
1 is wrong, guess again.
7 is wrong, guess again.

Enter a number between 0 and 9 inclusive: 9
1 is wrong, guess again.
7 is wrong, guess again.
5 is wrong, guess again.
6 is wrong, guess again.
1 is wrong, guess again.
7 is wrong, guess again.
4 is wrong, guess again.
4 is wrong, guess again.
8 is wrong, guess again.
7 is wrong, guess again.

Enter a number between 0 and 9 inclusive: 1
You guessed my number!
7 is wrong, guess again.
2 is wrong, guess again.
4 is wrong, guess again.
8 is wrong, guess again.
4 is wrong, guess again.
3 is wrong, guess again.
3 is wrong, guess again.
8 is wrong, guess again.
4 is wrong, guess again.

>>>
Step 1: just do one round

Goal output:

Enter a number between 0 and 9 inclusive: 3
2 is wrong, guess again.
8 is wrong, guess again.
9 is wrong, guess again.
8 is wrong, guess again.
You guessed my number!
8 is wrong, guess again.
8 is wrong, guess again.
8 is wrong, guess again.
1 is wrong, guess again.
7 is wrong, guess again.
Step 2: write the outer for-loop

Add the code:

```python
rounds = eval(input("How many times to do you want to play? "))
for j in range(rounds):
```
Step 3: indent!

```python
# TODO: this needs comments!

import random

def main():
    rounds = eval(input("How many times to do you want to play? "))

    for j in range(rounds):
        print()
        secret_number = eval(input("Enter a number between 0 and 9 inclusive: "))

        for i in range(10):
            guess = random.randint(0,9)

            if guess == secret_number:
                print("You guessed my number!")
            else:
                print(guess, "is wrong, guess again.")

main()
```
Homework 3 Preview: ELIZA

http://www.masswerk.at/elizabot/