### CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

イロト イポト イヨト イヨト

CSci 127 (Hunter)

Lecture 4

Э 3 October 2018 1 / 30

#### Announcements



• CS Survey:

Today: Bernard Desert & Elise Harris, CUNY 2X & Tech Talent Pipeline

< 口 > < 同

-

Э

Sac

From lecture slips & recitation sections.

From lecture slips & recitation sections.

• I missed or did terrible on a quiz. What does that do to my grade?

From lecture slips & recitation sections.

 I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade.

From lecture slips & recitation sections.

 I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors?

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors? Yes! In today's lecture and the next couple of labs.

E SQC

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors? Yes! In today's lecture and the next couple of labs.
- Why hexadecimal? Why can't we just use decimal?

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors? Yes! In today's lecture and the next couple of labs.
- Why hexadecimal? Why can't we just use decimal? Standard way of representing colors.

200

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors? Yes! In today's lecture and the next couple of labs.
- Why hexadecimal? Why can't we just use decimal? Standard way of representing colors. And more! More in later classes.

200

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors? Yes! In today's lecture and the next couple of labs.
- Why hexadecimal? Why can't we just use decimal? Standard way of representing colors. And more! More in later classes.
- What does len() mean?

E SQC

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors? Yes! In today's lecture and the next couple of labs.
- Why hexadecimal? Why can't we just use decimal? Standard way of representing colors. And more! More in later classes.
- What does len() mean?
  len(s) gives the length (# of items or chars.). Ex: len("hi!!") is 4.

≡ ∽ar

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors? Yes! In today's lecture and the next couple of labs.
- Why hexadecimal? Why can't we just use decimal? Standard way of representing colors. And more! More in later classes.
- What does len() mean?
  len(s) gives the length (# of items or chars.). Ex: len("hi!!") is 4.
- When do you use :? What's a slice?

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors? Yes! In today's lecture and the next couple of labs.
- Why hexadecimal? Why can't we just use decimal? Standard way of representing colors. And more! More in later classes.
- What does len() mean?
  len(s) gives the length (# of items or chars.). Ex: len("hi!!") is 4.
- When do you use :? What's a slice? The colon, :, gives a slice, substring or sublist, ex: myString[3:5].

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors? Yes! In today's lecture and the next couple of labs.
- Why hexadecimal? Why can't we just use decimal? Standard way of representing colors. And more! More in later classes.
- What does len() mean?
  len(s) gives the length (# of items or chars.). Ex: len("hi!!") is 4.
- When do you use :? What's a slice? The colon, :, gives a slice, substring or sublist, ex: myString[3:5]. More today!

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors? Yes! In today's lecture and the next couple of labs.
- Why hexadecimal? Why can't we just use decimal? Standard way of representing colors. And more! More in later classes.
- What does len() mean?
  len(s) gives the length (# of items or chars.). Ex: len("hi!!") is 4.
- When do you use :? What's a slice? The colon, :, gives a slice, substring or sublist, ex: myString[3:5]. More today!
- What is numpy really? And matplotlib & pyplot?

CSci 127 (Hunter)

Lecture 4

3 October 2018 3 / 30

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors? Yes! In today's lecture and the next couple of labs.
- Why hexadecimal? Why can't we just use decimal? Standard way of representing colors. And more! More in later classes.
- What does len() mean?
  len(s) gives the length (# of items or chars.). Ex: len("hi!!") is 4.
- When do you use :? What's a slice? The colon, :, gives a slice, substring or sublist, ex: myString[3:5]. More today!
- What is numpy really? And matplotlib & pyplot? They are Python files that includes useful functions, definitions, etc.

CSci 127 (Hunter)

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors? Yes! In today's lecture and the next couple of labs.
- Why hexadecimal? Why can't we just use decimal? Standard way of representing colors. And more! More in later classes.
- What does len() mean?
  len(s) gives the length (# of items or chars.). Ex: len("hi!!") is 4.
- When do you use :? What's a slice? The colon, :, gives a slice, substring or sublist, ex: myString[3:5]. More today!
- What is numpy really? And matplotlib & pyplot? They are Python files that includes useful functions, definitions, etc.
- Could you spend more time on problem solving & algorithms?

CSci 127 (Hunter)

From lecture slips & recitation sections.

- I missed or did terrible on a quiz. What does that do to my grade? Nothing. The in-class quizzes (and lecture slips) only help your grade. If you do better than on the final; we use it. Otherwise, we use the final grade.
- Could you spend more time on colors? Yes! In today's lecture and the next couple of labs.
- Why hexadecimal? Why can't we just use decimal? Standard way of representing colors. And more! More in later classes.
- What does len() mean?
  len(s) gives the length (# of items or chars.). Ex: len("hi!!") is 4.
- When do you use :? What's a slice? The colon, :, gives a slice, substring or sublist, ex: myString[3:5]. More today!
- What is numpy really? And matplotlib & pyplot? They are Python files that includes useful functions, definitions, etc.
- Could you spend more time on problem solving & algorithms? Yes! More in upcoming lectures & labs.

CSci 127 (Hunter)

3 October 2018 3 / 30

# Today's Topics



- Recap: Colors
- Indexing and Slicing
- Design Question: Cropping Images
- Decisions

Э

999

Color Name	HEX	Color
Black	<u>#000000</u>	
Navy	<u>#000080</u>	
DarkBlue	<u>#00008B</u>	
MediumBlue	#0000CD	
Blue	#0000FF	

• Can specify by name.

999

<ロト <回ト < 回ト < 回ト

Color Name	HEX	Color
Black	<u>#000000</u>	
Navy	<u>#000080</u>	
DarkBlue	<u>#00008B</u>	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by name.
- Can specify by numbers:

990

Color Name	HEX	Color
Black	<u>#000000</u>	
Navy	<u>#000080</u>	
DarkBlue	<u>#00008B</u>	
MediumBlue	#0000CD	
Blue	<u>#0000FF</u>	

- Can specify by name.
- Can specify by numbers:
  - Amount of Red, Green, and Blue (RGB).

CSci 127 (Hunter)

996

Color Name	HEX	Color
Black	<u>#000000</u>	
Navy	<u>#000080</u>	
DarkBlue	<u>#00008B</u>	
MediumBlue	#0000CD	
Blue	<u>#0000FF</u>	

- Can specify by name.
- Can specify by numbers:
  - ► Amount of Red, Green, and Blue (RGB).
  - Adding light, not paint:

CSci 127 (Hunter)

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

Color Name	HEX	Color
Black	<u>#000000</u>	
Navy	<u>#000080</u>	
DarkBlue	<u>#00008B</u>	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by name.
- Can specify by numbers:
  - Amount of Red, Green, and Blue (RGB).
  - Adding light, not paint:
    - ★ Black: 0% red, 0% green, 0% blue

CSci 127 (Hunter)

200

Color Name	HEX	Color
Black	<u>#000000</u>	
Navy	<u>#000080</u>	
DarkBlue	<u>#00008B</u>	
MediumBlue	#0000CD	
Blue	#0000FF	

- Can specify by name.
- Can specify by numbers:
  - ► Amount of Red, Green, and Blue (RGB).
  - Adding light, not paint:
    - ★ Black: 0% red, 0% green, 0% blue
    - ★ White: 100% red, 100% green, 100% blue

CSci 127 (Hunter)

E Sac

Color Name	HEX	Color
Black	<u>#000000</u>	
Navy	<u>#000080</u>	
DarkBlue	<u>#00008B</u>	
MediumBlue	#0000CD	
Blue	#0000FF	

#### • Can specify by numbers (RGB):

9990

Color Name	HEX	Color
Black	#000000	
Navy	#000080	
DarkBlue	<u>#00008B</u>	
MediumBlue	#0000CD	
Blue	#0000FF	

#### • Can specify by numbers (RGB):

► Fractions of each:

e.g. (1.0, 0, 0) is 100% red, no green, and no blue.

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

Color Name	HEX	Color
Black	<u>#000000</u>	
Navy	<u>#000080</u>	
DarkBlue	<u>#00008B</u>	
MediumBlue	#0000CD	
Blue	#0000FF	

#### • Can specify by numbers (RGB):

- Fractions of each:
  - e.g. (1.0, 0, 0) is 100% red, no green, and no blue.
- ▶ 8-bit colors: numbers from 0 to 255:
  e.g. (0, 255, 0) is no red, 100% green, and no blue.

CSci 127 (Hunter)

Color Name	HEX	Color
Black	<u>#000000</u>	
Navy	<u>#000080</u>	
DarkBlue	<u>#00008B</u>	
MediumBlue	#0000CD	
Blue	#0000FF	

#### • Can specify by numbers (RGB):

Fractions of each:

e.g. (1.0, 0, 0) is 100% red, no green, and no blue.

- ▶ 8-bit colors: numbers from 0 to 255: e.g. (0, 255, 0) is no red, 100% green, and no blue.
- Hexcodes (base-16 numbers)...

CSci 127 (Hunter)

#### Recap: Hexadecimal



00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F 60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F 90 91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F AO A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF CO C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF DO D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF EO E1 E2 E3 E4 E5 E6 E7 E8 E9 EA EB EC ED EE EF F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF

Lecture 4

3 October 2018 7 / 30

イロト イポト イヨト イヨ

# Colors

Color Name	HEX	Color
Black	<u>#000000</u>	
Navy	<u>#000080</u>	
DarkBlue	<u>#00008B</u>	
MediumBlue	#0000CD	
Blue	#0000FF	

#### • Can specify by numbers (RGB):

- ► Fractions of each:
  - e.g. (1.0, 0, 0) is 100% red, no green, and no blue.
- ▶ 8-bit colors: numbers from 0 to 255: e.g. (0, 255, 0) is no red, 100% green, and no blue.
- Hexcodes (base-16 numbers):

CSci 127 (Hunter)

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 ろの⊙

# Colors

Color Name	HEX	Color
Black	<u>#000000</u>	
Navy	<u>#000080</u>	
DarkBlue	<u>#00008B</u>	
MediumBlue	#0000CD	
Blue	#0000FF	

#### • Can specify by numbers (RGB):

- ► Fractions of each:
  - e.g. (1.0, 0, 0) is 100% red, no green, and no blue.
- ▶ 8-bit colors: numbers from 0 to 255: e.g. (0, 255, 0) is no red, 100% green, and no blue.
- Hexcodes (base-16 numbers):
  e.g. #0000FF is no red, no green, and 100% blue.

CSci 127 (Hunter)

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 ろの⊙

Images



◆□▶ ◆□▶ ◆三▶ ◆三▶ ・三 ・ ��や

Images



This image has 287 rows, 573 columns, and 4 color channels (for red, green, blue, and a 4th for how transparent).

CSci 127 (Hunter)

Lecture 4

3 3 October 2018 9 / 30

Sac
In Pairs or Triples...

Let's start with loops & slices:

```
word = "Hunter"
for i in range(2,10,3):
    for c in word:
        print(i.c. end = "")
    print()
pali = "a man a plan a canal Panama"
print(pali[0], pali[-1])
print(pali[2:5], pali[-4:-1])
qPop = [152999, 284041, 469042, 1079129, 1297634]
    1550849, 1809578, 1986473, 1891325, 1951598,
    2229379,2230722]
print("Queens population in 1900:", aPop[0])
print("Since 2000:", qPop[-3:len(qPop)])
```

CSci 127 (Hunter)

3 October 2018 10 / 30

▲□▶ ▲□▶ ▲三▶ ▲三▶ 三三 ろの⊙

### Python Tutor

```
word = "Hunter"
for i in range(2,10,3):
    for c in word:
        print(i,c, end = "")
    print()
pali = "a man a plan a canal Panama"
    print(pali[3], pali[-1])
    print(pali[2], pali[-4:-1])
    qPop = [152999,284041,469042,1079129,1297634,
    1550849,1809578,1986473,1881325,1951598,
    222379,2230722]
    print("Queens population in 1900:", qPop[3])
    print("Queens population in 1900:", qPop[3])
    print("Queens population in 1900:", qPop[3])
```

#### (Demo with pythonTutor)

200

イロト 不得 トイヨト イヨト 二日



990











CSci 127 (Hunter)

Lecture 4

3 3 October 2018 12 / 30

990





Design Question: Design an algorithm that will crop an image.

CSci 127 (Hunter)

Lecture 4

3 October 2018 12 / 30

= nar





Design Question: Design an algorithm that will crop an image.First: specify what the inputs & outputs for the algorithm.

CSci 127 (Hunter)

Lecture 4

3 October 2018 12 / 30

イロト イロト イヨト イ





Design Question: Design an algorithm that will crop an image.

- First: specify what the inputs & outputs for the algorithm.
- Next: write pseudocode.

< D > < P > < P > < P >





Design Question: Design an algorithm that will crop an image.

- First: specify what the inputs & outputs for the algorithm.
- Next: write pseudocode.
- If time: translate to Python.

∃ >





Design Question: Design an algorithm that will crop an image.

- First: specify what the inputs & outputs for the algorithm.
- Next: write pseudocode.
- If time: translate to Python.

(Write your design as if you are the manager listing out tasks for your employees to do.)

CSci 127 (Hunter)

3 October 2018 12 / 30

### In Pairs or Triples: Cropping Images





Design Question: Design an algorithm that will crop an image.

- First: specify inputs/outputs.
- Next: write pseudocode.
- If time: translate to Python

CSci 127 (Hunter)

Lecture 4

3 October 2018 13 / 30

3





• First: specify inputs/outputs.

CSci 127 (Hunter)

Lecture 4

3 October 2018 14 / 30

3

990





• First: specify inputs/outputs. Inputs: name of file to be read in,

イロト イポト イヨト イヨト

200





• First: specify inputs/outputs. Inputs: name of file to be read in, name of file to saved, and





Inputs: name of file to be read in, name of file to saved, and the upper, lower, left, right coordinates ("bounding box") Outputs: cropped file.

CSci 127 (Hunter)





- First: specify inputs/outputs. Input file name, output file name, upper, lower, left, right ("bounding box") & output cropped file.
- Next: write pseudocode.

イロト イロト イヨト イ





- First: specify inputs/outputs. Input file name, output file name, upper, lower, left, right ("bounding box") & output cropped file.
- Next: write pseudocode.
  - Import numpy and pyplot.

イロト イロト イヨト イ





- First: specify inputs/outputs. Input file name, output file name, upper, lower, left, right ("bounding box") & output cropped file.
- Next: write pseudocode.
  - Import numpy and pyplot.
  - 2 Ask user for file names and dimensions for cropping.





- First: specify inputs/outputs. Input file name, output file name, upper, lower, left, right ("bounding box") & output cropped file.
- Next: write pseudocode.
  - Import numpy and pyplot.
  - 2 Ask user for file names and dimensions for cropping.
  - Save input file to an array.





- First: specify inputs/outputs. Input file name, output file name, upper, lower, left, right ("bounding box") & output cropped file.
- Next: write pseudocode.
  - Import numpy and pyplot.
  - 2 Ask user for file names and dimensions for cropping.
  - ③ Save input file to an array.
  - ④ Copy the cropped portion to a new array.

イロト イボト イヨト イヨ





- First: specify inputs/outputs. Input file name, output file name, upper, lower, left, right ("bounding box") & output cropped file.
- Next: write pseudocode.
  - Import numpy and pyplot.
  - 2 Ask user for file names and dimensions for cropping.
  - ③ Save input file to an array.
  - ④ Copy the cropped portion to a new array.
  - 5 Save the new array to the output file.







- First: specify inputs/outputs. Input file name, output file name, upper, lower, left, right ("bounding box") & output cropped file.
- Next: write pseudocode.
  - Import numpy and pyplot.
  - 2 Ask user for file names and dimensions for cropping.
  - ③ Save input file to an array.
  - ④ Copy the cropped portion to a new array.
  - 5 Save the new array to the output file.

#### • If time: translate to Python.

```
#Name: CSci 127 Teaching Staff
#Date: Fall 2017
#This program loads an image, displays it, and then creates, displays,
               and saves a new image that has only the red channel displayed.
#
#Import the packages for images and arrays:
import matplotlib.pyplot as plt
import numpy as np
inImg = input('Enter input image: ')
img = plt.imread(inImg) #Read in image from csBridge.png
plt.imshow(ima) #Load image into pyplot
plt.show()
                                                                    #Show the image (waits until closed to continue)
outImg = input('Enter out image: ')
t = int(input('Enter top:'))
b = int(input('Enter bottom:'))
l = int(input('Enter left: '))
r = int(input('Enter right: '))
ima2 = ima[t:b,l:r] #Slice the original array by dimensions entered
plt.imshow(img2)
                                                                            #Load our new image into pyplot
plt.show()
                                                                             #Show the image (waits until closed to continue)
plt.imsave(outIma, ima2) #Save the image we created to the out file.
                                                                                                                                                                                   <ロト < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < 回 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □
             CSci 127 (Hunter)
                                                                                                                            Lecture 4
                                                                                                                                                                                                                          3 October 2018
```

17 / 30

### In Pairs or Triples...

Predict what these will do (novel concepts):

```
yearBorn = int(input('Enter year born: '))
if vearBorn < 1946:
    print("Greatest Generation")
elif vearBorn <= 1964:
    print("Baby Boomer")
elif yearBorn <= 1984:
    print("Generation X")
elif vearBorn <= 2004:
    print("Millennial")
else:
    print("TBD")
x = int(input('Enter number: '))
if x % 2 == 0:
    print('Even number')
else:
    print('Odd number')
```

import turtle

```
tess = turtle.Turtle()
myWin = turtle.Screen() #The graphics window
commands = input("Please enter a command string: ")
```

for ch in commands:
 #perform action indicated by the character

```
if ch == 'E':
                         #move forward
   tess.forward(50)
elif ch == 'l':
                         #turn left
   tess.left(90)
elif ch == 'R'
                         #turn right
   tess.right(90)
elif ch == '^':
                         #lift pen
   tess.penup()
elif ch == 'v':
                         #lower pen
   tess.pendown()
elif ch == 'B':
                         #ao backwards
   tess.backward(50)
elif ch == 'r':
                         #turn red
   tess.color("red")
elif ch == 'q':
                         #turn green
   tess.color("green")
elif ch == 'b':
                         #turn blue
   tess.color("blue")
else:
                        #for any other character
   print("Error: do not know the command:", c)
```

イロト イポト イヨト イヨト

3 October 2018 18 / 30

Sac

### Python Tutor

```
yearBorn = int(input('Enter year born: '))
if yearBorn < 1946:
   print("Greatest Generation")
elif yearBorn <= 1964:
    print("Baby Boomer")
elif yearBorn <= 1984:
   print("Generation X")
elif yearBorn <= 2004:
    print("Millennial")
else:
   print("TBD")
x = int(input('Enter number: '))
if x % 2 == 0:
   print('Even number')
else:
   print('Odd number')
```

#### (Demo with pythonTutor)

```
import turtle
tess = turtle.Turtle()
myWin = turtle.Screen()
                            #The anaphics window
commands = input("Please enter a command string: ")
for ch in commands:
    #perform action indicated by the character
    if ch == 'F':
                              #move forward
        tess.forward(50)
    elif ch == 'L':
                             #turn left
        tess.left(90)
    elif ch --- 'R':
                             #turn right
        tess.right(90)
    elif ch --- '^':
                             #lift pen
        tess.penup()
    elif ch == 'v':
                             #lower pen
        tess.pendown()
    elif ch == 'B':
                             #go backwards
        tess.backward(50)
    elif ch --- 'r':
                             #turn red
        tess.color("red")
    elif ch == 'a':
                             #turn areen
        tess.color("green")
    elif ch == 'b':
                             #turn blue
        tess.color("blue")
    else:
                            #for any other character
        print("Error: do not know the command:", c)
```

#### (Demo with IDLE)

Sac

### Decisions

```
if x < y:
    print("x is less than y")
elif x > y:
    print("x is greater than y")
else:
    print("x and y must be equal")
```

< □ > < □ > < 三 > < 三 > < 三 > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □

### Decisions

```
if x < y:
    print("x is less than y")
elif x > y:
    print("x is greater than y")
else:
    print("x and y must be equal")
```





(This was just a first glance, will do much more on decisions over the next several weeks.)

CSci 127 (Hunter)

3 October 2018 21 / 30

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □



#### Bernard Desert & Elise Harris

Sac



#### Bernard Desert & Elise Harris

Brief overview of CUNY 2X & Tech Talent Pipeline

CSci 127 (Hunter)

Lecture 4

3 October 2018 22 / 30

◆ロト ◆信 ト ◆注 ト ◆注 ト



#### Bernard Desert & Elise Harris

- Brief overview of CUNY 2X & Tech Talent Pipeline
- What Bernard & Elise love about their jobs.

CSci 127 (Hunter)

Lecture 4

3 October 2018 22 / 30



#### Bernard Desert & Elise Harris

- Brief overview of CUNY 2X & Tech Talent Pipeline
- What Bernard & Elise love about their jobs.
- Design challenge: classic tech interview question.

CSci 127 (Hunter)

Lecture 4

3 October 2018 22 / 30

### CS Survey Talk: Hunter Tech Calendar



To sign up:

- http://bit.ly/cuny2xcontactinfo
- Does not have to be a Hunter email- prefer one that you access most.

CSci 127 (Hunter)

Lecture 4

3 3 October 2018 23 / 30

E 1 4 E

### Tech Interview Classic

 Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

### Tech Interview Classic

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- Write down the output to see the pattern:
- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- Write down the output to see the pattern:

1

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- Write down the output to see the pattern:

1 2

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- Write down the output to see the pattern:

1 2 Fizz

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- Write down the output to see the pattern:

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- Write down the output to see the pattern:

1 2 Fizz 4 Buzz

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- Write down the output to see the pattern:

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- Write down the output to see the pattern:

1 2 Fizz 4 Buzz 5 Fizz

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- Write down the output to see the pattern:

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- Write down the output to see the pattern:

1 2 Fizz 4 Buzz 5 Fizz 7 ... 14

CSci 127 (Hunter)

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- Write down the output to see the pattern:

2 Fizz 4 Buzz 5 Fizz . . . 14 FizzBuzz

CSci 127 (Hunter)

 Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- To Do List:

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- To Do List:
  - Create a loop that goes from 1 to 100.

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- To Do List:
  - Create a loop that goes from 1 to 100.
  - ► If the number is divisible by 3, print "Fizz".

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- To Do List:
  - Create a loop that goes from 1 to 100.
  - ► If the number is divisible by 3, print "Fizz".
  - ► If the number is divisible by 5, print "Buzz".

 Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

• To Do List:

- Create a loop that goes from 1 to 100.
- ► If the number is divisible by 3, print "Fizz".
- ► If the number is divisible by 5, print "Buzz".
- ► If divisible by both, print "FizzBuzz".

 Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

• To Do List:

- Create a loop that goes from 1 to 100.
- ► If the number is divisible by 3, print "Fizz".
- ► If the number is divisible by 5, print "Buzz".
- ► If divisible by both, print "FizzBuzz".
- Otherwise print the number.

イロト 不得 トイヨト イヨト ヨー のくや

 Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

• To Do List:

- Create a loop that goes from 1 to 100.
- ► If the number is divisible by 3, print "Fizz".
- ► If the number is divisible by 5, print "Buzz".
- ► If divisible by both, print "FizzBuzz".
- Otherwise print the number. We should do this one first!

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- To Do List (Reordered):

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- To Do List (**Reordered**):
  - Create a loop that goes from 1 to 100.
  - Print the numbers not divisible by 3 or 5.
  - ► If the number is divisible by 3, print "Fizz".
  - ► If the number is divisible by 5, print "Buzz".
  - ► If divisible by both, print "FizzBuzz".

- Write a program that prints the numbers from 1 to 100. But for multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".
- To Do List (**Reordered**):
  - Create a loop that goes from 1 to 100.
  - Print the numbers not divisible by 3 or 5.
  - ► If the number is divisible by 3, print "Fizz".
  - ► If the number is divisible by 5, print "Buzz".
  - ► If divisible by both, print "FizzBuzz".
  - ► Also should print a new line (so each entry is on its own line).

イロト 不得 トイヨト イヨト ヨー のくや

- To Do List:
  - Create a loop that goes from 1 to 100.
  - Print the numbers not divisible by 3 or 5.
  - ► If the number is divisible by 3, print "Fizz".
  - ► If the number is divisible by 5, print "Buzz".
  - ► If divisible by both, print "FizzBuzz".
  - ► Also should print a new line (so each entry is on its own line).

- To Do List:
  - Create a loop that goes from 1 to 100.
  - Print the numbers not divisible by 3 or 5.
  - ► If the number is divisible by 3, print "Fizz".
  - ► If the number is divisible by 5, print "Buzz".
  - ► If divisible by both, print "FizzBuzz".
  - ► Also should print a new line (so each entry is on its own line).

• One solution (uses print(,end="") that prints all on the same line):

- To Do List:
  - Create a loop that goes from 1 to 100.
  - Print the numbers not divisible by 3 or 5.
  - ► If the number is divisible by 3, print "Fizz".
  - ► If the number is divisible by 5, print "Buzz".
  - ► If divisible by both, print "FizzBuzz".
  - ► Also should print a new line (so each entry is on its own line).
- One solution (uses print(,end="") that prints all on the same line):

for i in range(1,101):

イロト 不得 トイヨト イヨト ヨー のくや

- To Do List:
  - Create a loop that goes from 1 to 100.
  - Print the numbers not divisible by 3 or 5.
  - ► If the number is divisible by 3, print "Fizz".
  - ► If the number is divisible by 5, print "Buzz".
  - ► If divisible by both, print "FizzBuzz".
  - ► Also should print a new line (so each entry is on its own line).

• One solution (uses print(,end="") that prints all on the same line):

```
for i in range(1,101):
if i%3 != 0 and i%5 != 0:
```

- To Do List:
  - Create a loop that goes from 1 to 100.
  - Print the numbers not divisible by 3 or 5.
  - ► If the number is divisible by 3, print "Fizz".
  - ► If the number is divisible by 5, print "Buzz".
  - ► If divisible by both, print "FizzBuzz".
  - ► Also should print a new line (so each entry is on its own line).

• One solution (uses print(,end="") that prints all on the same line):

```
for i in range(1,101):
if i%3 != 0 and i%5 != 0:
    print(i, end="")
```

- To Do List:
  - Create a loop that goes from 1 to 100.
  - Print the numbers not divisible by 3 or 5.
  - ► If the number is divisible by 3, print "Fizz".
  - ► If the number is divisible by 5, print "Buzz".
  - ► If divisible by both, print "FizzBuzz".
  - ► Also should print a new line (so each entry is on its own line).

• One solution (uses print(,end="") that prints all on the same line):

```
for i in range(1,101):
if i%3 != 0 and i%5 != 0:
    print(i, end="")
if i%3 == 0:
```

- To Do List:
  - Create a loop that goes from 1 to 100.
  - Print the numbers not divisible by 3 or 5.
  - ► If the number is divisible by 3, print "Fizz".
  - ► If the number is divisible by 5, print "Buzz".
  - ► If divisible by both, print "FizzBuzz".
  - ► Also should print a new line (so each entry is on its own line).

• One solution (uses print(,end="") that prints all on the same line):

```
for i in range(1,101):
if i%3 != 0 and i%5 != 0:
    print(i, end="")
if i%3 == 0:
    print("Fizz", end="")
```

- To Do List:
  - Create a loop that goes from 1 to 100.
  - Print the numbers not divisible by 3 or 5.
  - ► If the number is divisible by 3, print "Fizz".
  - ► If the number is divisible by 5, print "Buzz".
  - ► If divisible by both, print "FizzBuzz".
  - ► Also should print a new line (so each entry is on its own line).

• One solution (uses print(,end="") that prints all on the same line):

```
for i in range(1,101):
if i%3 != 0 and i%5 != 0:
    print(i, end="")
if i%3 == 0:
    print("Fizz", end="")
if i%5 == 0:
```

CSci 127 (Hunter)

3 October 2018 27 / 30

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

- To Do List:
  - Create a loop that goes from 1 to 100.
  - Print the numbers not divisible by 3 or 5.
  - ► If the number is divisible by 3, print "Fizz".
  - ► If the number is divisible by 5, print "Buzz".
  - ► If divisible by both, print "FizzBuzz".
  - ► Also should print a new line (so each entry is on its own line).

• One solution (uses print(,end="") that prints all on the same line):

```
for i in range(1,101):
if i%3 != 0 and i%5 != 0:
    print(i, end="")
if i%3 == 0:
    print("Fizz", end="")
if i%5 == 0:
    print("Buzz", end="")
```

CSci 127 (Hunter)

3 October 2018 27 / 30

◆□▶ ◆□▶ ◆三▶ ◆三▶ ○○○

- To Do List:
  - Create a loop that goes from 1 to 100.
  - Print the numbers not divisible by 3 or 5.
  - ► If the number is divisible by 3, print "Fizz".
  - ► If the number is divisible by 5, print "Buzz".
  - ► If divisible by both, print "FizzBuzz".
  - ► Also should print a new line (so each entry is on its own line).

• One solution (uses print(,end="") that prints all on the same line):

```
for i in range(1,101):
if i%3 != 0 and i%5 != 0:
    print(i, end="")
if i%3 == 0:
    print("Fizz", end="")
if i%5 == 0:
    print("Buzz", end="")
print()
```

CSci 127 (Hunter)

3 October 2018 27 / 30

< □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □ > < □



• On lecture slip, write down a topic you wish we had spent more time (and why).

3

590



- On lecture slip, write down a topic you wish we had spent more time (and why).
- In Python, we introduced:

3

900



- On lecture slip, write down a topic you wish we had spent more time (and why).
- In Python, we introduced:
  - ► Recap: Colors
  - Indexing and Slicing
  - Design Question: Cropping Images
  - Decisions

Sar



- On lecture slip, write down a topic you wish we had spent more time (and why).
- In Python, we introduced:
  - ► Recap: Colors
  - Indexing and Slicing
  - Design Question: Cropping Images
  - Decisions
- Pass your lecture slips to the aisles for the UTAs to collect.



- On lecture slip, write down a topic you wish we had spent more time (and why).
- In Python, we introduced:
  - ► Recap: Colors
  - Indexing and Slicing
  - Design Question: Cropping Images
  - Decisions
- Pass your lecture slips to the aisles for the UTAs to collect.


(NYTimes)

(Hunter College)



• Since you must pass the final exam to pass the course, we end every lecture with final exam review.

12 b. 4



(NYTimes)

(Hunter College)



- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).



(NYTimes)

(Hunter College)



- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:



(NYTimes)

(Hunter College)



イロト イポト イヨト イ

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;



(NYTimes)

(Hunter College)



イロト イポト イヨト イ

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and

CSci 127 (Hunter)



(NYTimes)

(Hunter College)



- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - repeat.

CSci 127 (Hunter)

3 October 2018 29 / 30



(NYTimes)

(Hunter College)



・ コ ト ・ 雪 ト ・ ヨ ト ・

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - ▶ repeat.
- Past exams are on the webpage (under Final Exam Information).

CSci 127 (Hunter)

Lecture 4

3 October 2018 29 / 30



(NYTimes)

(Hunter College)



< □ > < 同 >

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - ► repeat.
- Past exams are on the webpage (under Final Exam Information).
- We're starting with Fall 2017, Version 3.

CSci 127 (Hunter)

Lecture 4

3 October 2018 29 / 30

# Writing Boards



• Return writing boards as you leave...

CSci 127 (Hunter)

Lecture 4

3 October 2018 30 / 30

3

990

< ロ ト < 団 ト < 三 ト < 三 ト</p>