

CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

Announcements



- Starting this week, each lecture will include a survey of computing research and the tech industry in NYC.

Announcements



- Starting this week, each lecture will include a survey of computing research and the tech industry in NYC.
- Today: Prof. William Sakas,
Chair of Computer Science:
Computational Linguistics.

Frequently Asked Questions

From lecture slips & recitation sections.

Frequently Asked Questions

From lecture slips & recitation sections.

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

Frequently Asked Questions

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.

Frequently Asked Questions

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.*

Frequently Asked Questions

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?

Frequently Asked Questions

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.

Frequently Asked Questions

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?

Frequently Asked Questions

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.

Frequently Asked Questions

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.

Frequently Asked Questions

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?

Frequently Asked Questions

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.

Frequently Asked Questions

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?

Frequently Asked Questions

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].

Frequently Asked Questions

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!*

Frequently Asked Questions

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?

Frequently Asked Questions

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.

Frequently Asked Questions

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?

Frequently Asked Questions

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.

One More FAQ: Why Paper Planes?



One More FAQ: Why Paper Planes?



Why paper planes?

One More FAQ: Why Paper Planes?



Why paper planes?

- It's a classic design question in introductory programming classes, since

One More FAQ: Why Paper Planes?



Why paper planes?

- It's a classic design question in introductory programming classes, since
 - ▶ Practice writing solutions in plain English (without getting tied up in syntax).

One More FAQ: Why Paper Planes?



Why paper planes?

- It's a classic design question in introductory programming classes, since
 - ▶ Practice writing solutions in plain English (without getting tied up in syntax).
 - ▶ Practice thinking (and writing) precisely.

One More FAQ: Why Paper Planes?



Why paper planes?

- It's a classic design question in introductory programming classes, since
 - ▶ Practice writing solutions in plain English (without getting tied up in syntax).
 - ▶ Practice thinking (and writing) precisely.
- Why in groups?

One More FAQ: Why Paper Planes?



Why paper planes?

- It's a classic design question in introductory programming classes, since
 - ▶ Practice writing solutions in plain English (without getting tied up in syntax).
 - ▶ Practice thinking (and writing) precisely.
- Why in groups?
 - ▶ From our industry partners, our students need stronger communication skills.

One More FAQ: Why Paper Planes?



Why paper planes?

- It's a classic design question in introductory programming classes, since
 - ▶ Practice writing solutions in plain English (without getting tied up in syntax).
 - ▶ Practice thinking (and writing) precisely.
- Why in groups?
 - ▶ From our industry partners, our students need stronger communication skills.
 - ▶ In particular, communicating technical ideas precisely, and

One More FAQ: Why Paper Planes?



Why paper planes?

- It's a classic design question in introductory programming classes, since
 - ▶ Practice writing solutions in plain English (without getting tied up in syntax).
 - ▶ Practice thinking (and writing) precisely.
- Why in groups?
 - ▶ From our industry partners, our students need stronger communication skills.
 - ▶ In particular, communicating technical ideas precisely, and communicating and working in teams.

Today's Topics



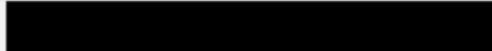
- Recap: Colors
- Indexing and Slicing
- Design Question: Hunter Logo
- Decisions
- CS Survey

Today's Topics



- **Recap: Colors**
- Indexing and Slicing
- Design Question: Hunter Logo
- Decisions
- CS Survey

Recap: Colors

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

- Can specify by name.

Recap: Colors

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

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

Recap: Colors

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

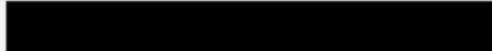
- Can specify by name.
- Can specify by numbers:
 - ▶ Amount of Red, Green, and Blue (RGB).

Recap: Colors

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

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

Recap: Colors

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

- 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

Recap: Colors

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

- 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

Recap: Colors

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

- Can specify by numbers (RGB):

Recap: Colors

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

- Can specify by numbers (RGB):
 - ▶ Fractions of each:
e.g. (1.0, 0, 0) is 100% red, no green, and no blue.

Recap: Colors

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

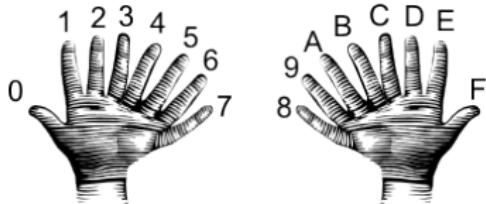
- Can specify by numbers (RGB):
 - ▶ Fractions of each:
e.g. (1.0, 0, 0) is 100% red, no green, and no blue.
 - ▶ 8-bit colors: numbers from 0 to 255:
e.g. (0, 255, 0) is no red, 100% green, and no blue.

Recap: Colors

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

- 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)...

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
A0 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
C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF
E0 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

Colors

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

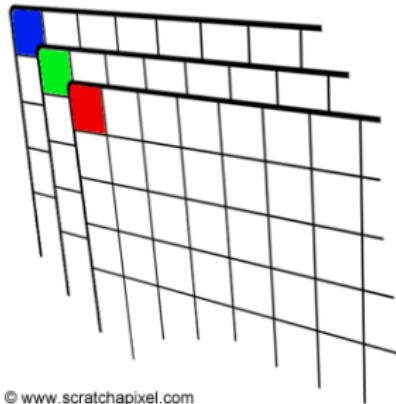
- Can specify by numbers (RGB):
 - ▶ Fractions of each:
e.g. (1.0, 0, 0) is 100% red, no green, and no blue.
 - ▶ 8-bit colors: numbers from 0 to 255:
e.g. (0, 255, 0) is no red, 100% green, and no blue.
 - ▶ Hexcodes (base-16 numbers):

Colors

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

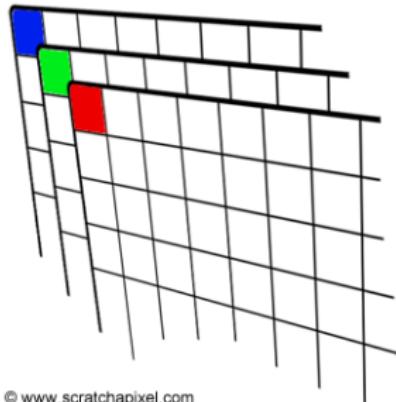
- Can specify by numbers (RGB):
 - ▶ Fractions of each:
e.g. (1.0, 0, 0) is 100% red, no green, and no blue.
 - ▶ 8-bit colors: numbers from 0 to 255:
e.g. (0, 255, 0) is no red, 100% green, and no blue.
 - ▶ Hexcodes (base-16 numbers):
e.g. #0000FF is no red, no green, and 100% blue.

Images



© www.scratchapixel.com

Images



© www.scratchapixel.com



`img[i, j, 0]`



`img[i, j, 1]`



`img[i, j, 2]`

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

Today's Topics



- Recap: Colors
- **Indexing and Slicing**
- Design Question: Hunter Logo
- Decisions
- CS Survey

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:", qPop[0])
print("Since 2000:", qPop[-3:len(qPop)])
```

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[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:", qPop[0])
print("Since 2000:", qPop[-3:len(qPop)])
```

(Demo with pythonTutor)

Today's Topics



- Recap: Colors
- Indexing and Slicing
- **Design Question: Hunter Logo**
- Decisions
- CS Survey

In Pairs or Triples...

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

- ① Design a 10 by 10 logo for Hunter College that contains a purple 'H'.

In Pairs or Triples...

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

- ① Design a 10 by 10 logo for Hunter College that contains a purple 'H'.
- ② Your logo should only contain the colors purple and white.

In Pairs or Triples...

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

- ① Design a 10 by 10 logo for Hunter College that contains a purple 'H'.
- ② Your logo should only contain the colors purple and white.
- ③ How can you make Python draw the logo?
Write down a "To Do" list of things you need to do.

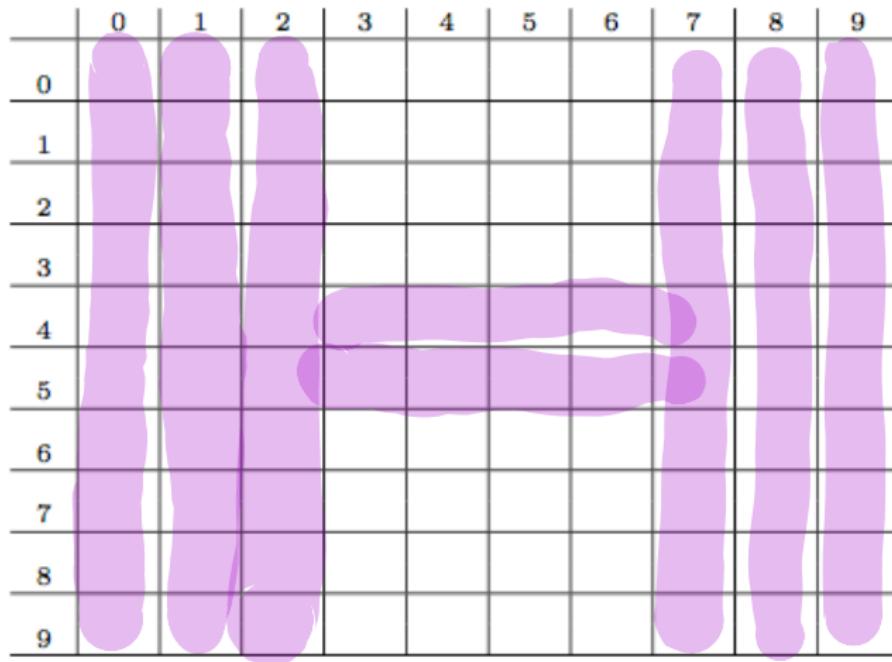
In Pairs or Triples...

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

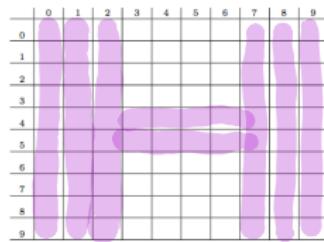
- ① Design a 10 by 10 logo for Hunter College that contains a purple 'H'.
- ② Your logo should only contain the colors purple and white.
- ③ How can you make Python draw the logo?
Write down a "To Do" list of things you need to do.
- ④ If time, refine your steps above into a Python program.

Design a Hunter Logo

One possible solution:

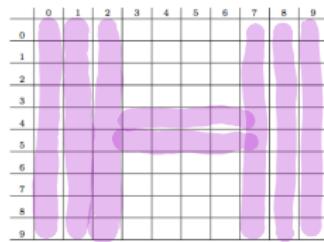


Design a Hunter Logo



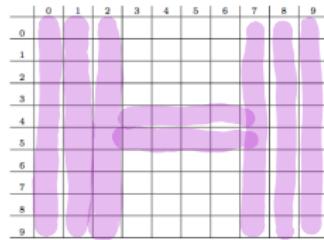
- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

Design a Hunter Logo



- ① Create a 10 by 10 array, logo, that starts out as all white pixels.
- ② Set the 3 left columns to be purple.

Design a Hunter Logo



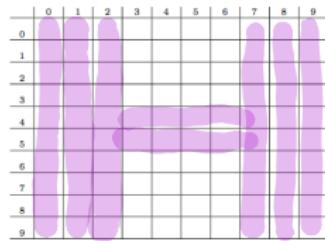
- ① Create a 10 by 10 array, logo, that starts out as all white pixels.
- ② Set the 3 left columns to be purple.
- ③ Set the 3 right columns to be purple.

Design a Hunter Logo

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

- ① Create a 10 by 10 array, logo, that starts out as all white pixels.
- ② Set the 3 left columns to be purple.
- ③ Set the 3 right columns to be purple.
- ④ Set the middle 2 rows to be purple.

Design a Hunter Logo



- ① Create a 10 by 10 array, logo, that starts out as all white pixels.
- ② Set the 3 left columns to be purple.
- ③ Set the 3 right columns to be purple.
- ④ Set the middle 2 rows to be purple.
- ⑤ Save logo array to a file.

Translating the Design to Code

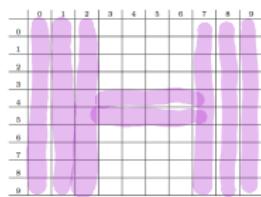
- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

	0	1	2	3	4	5	6	7	8	9
0										
1										
2										
3										
4										
5										
6										
7										
8										
9										

Translating the Design to Code

- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

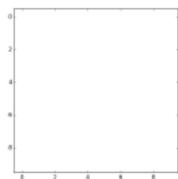
```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np #and for arrays (to hold images)
logoImg = np.ones((10,10,3)) #10x10 array with 3 sheets of 1's
```



Translating the Design to Code

- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np #and for arrays (to hold images)
logoImg = np.ones((10,10,3)) #10x10 array with 3 sheets of 1's
```

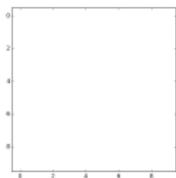


Translating the Design to Code

- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np #and for arrays (to hold images)
logoImg = np.ones((10,10,3)) #10x10 array with 3 sheets of 1's
```

- ② Set the 3 left columns to be purple.



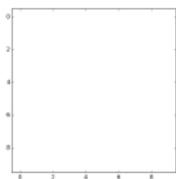
Translating the Design to Code

- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np #and for arrays (to hold images)
logoImg = np.ones((10,10,3)) #10x10 array with 3 sheets of 1's
```

- ② Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:, :, 1] = 0 #Turn the green to 0 for first 3 columns
```



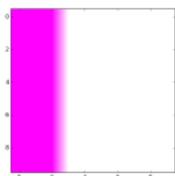
Translating the Design to Code

- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np #and for arrays (to hold images)
logoImg = np.ones((10,10,3)) #10x10 array with 3 sheets of 1's
```

- ② Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:, :, 1] = 0 #Turn the green to 0 for first 3 columns
```



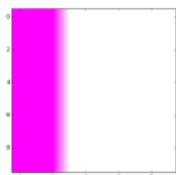
Translating the Design to Code

- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np #and for arrays (to hold images)
logoImg = np.ones((10,10,3)) #10x10 array with 3 sheets of 1's
```

- ② Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:, :3, 1] = 0 #Turn the green to 0 for first 3 columns
```



- ③ Set the 3 right columns to be purple.

```
logoImg[:, -3:, 1] = 0 #Turn the green to 0 for last 3 columns
```

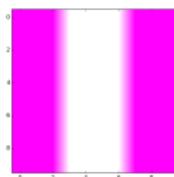
Translating the Design to Code

- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np #and for arrays (to hold images)
logoImg = np.ones((10,10,3)) #10x10 array with 3 sheets of 1's
```

- ② Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:, :, 1] = 0 #Turn the green to 0 for first 3 columns
```



- ③ Set the 3 right columns to be purple.

```
logoImg[:, -3:, 1] = 0 #Turn the green to 0 for last 3 columns
```

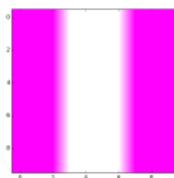
Translating the Design to Code

- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np #and for arrays (to hold images)
logoImg = np.ones((10,10,3)) #10x10 array with 3 sheets of 1's
```

- ② Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:, :, 1] = 0 #Turn the green to 0 for first 3 columns
```



- ③ Set the 3 right columns to be purple.

```
logoImg[:, -3:, 1] = 0 #Turn the green to 0 for last 3 columns
```

- ④ Set the middle 2 rows to be purple.

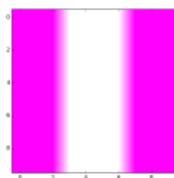
Translating the Design to Code

- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np #and for arrays (to hold images)
logoImg = np.ones((10,10,3)) #10x10 array with 3 sheets of 1's
```

- ② Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:, :, 1] = 0 #Turn the green to 0 for first 3 columns
```



- ③ Set the 3 right columns to be purple.

```
logoImg[:, -3:, 1] = 0 #Turn the green to 0 for last 3 columns
```

- ④ Set the middle 2 rows to be purple.

```
logoImg[4:6, :, 1] = 0 #Turn the green to 0 for middle rows
```

Translating the Design to Code

- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np             #and for arrays (to hold images)
logoImg = np.ones((10,10,3))   #10x10 array with 3 sheets of 1's
```

- ② Set the 3 left columns to be purple.

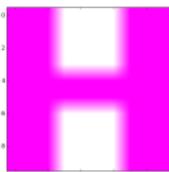
```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:, :3, 1] = 0 #Turn the green to 0 for first 3 columns
```

- ③ Set the 3 right columns to be purple.

```
logoImg[:, -3:, 1] = 0 #Turn the green to 0 for last 3 columns
```

- ④ Set the middle 2 rows to be purple.

```
logoImg[4:6, :, 1] = 0 #Turn the green to 0 for middle rows
```



Translating the Design to Code

- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np #and for arrays (to hold images)
logoImg = np.ones((10,10,3)) #10x10 array with 3 sheets of 1's
```

- ② Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:, :3, 1] = 0 #Turn the green to 0 for first 3 columns
```

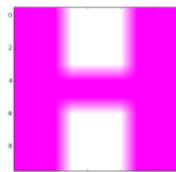
- ③ Set the 3 right columns to be purple.

```
logoImg[:, -3:, 1] = 0 #Turn the green to 0 for last 3 columns
```

- ④ Set the middle 2 rows to be purple.

```
logoImg[4:6, :, 1] = 0 #Turn the green to 0 for middle rows
```

- ⑤ Save logo array to file.



Translating the Design to Code

- ① Create a 10 by 10 array, logo, that starts out as all white pixels.

```
import matplotlib.pyplot as plt #import libraries for plotting
import numpy as np             #and for arrays (to hold images)
logoImg = np.ones((10,10,3))   #10x10 array with 3 sheets of 1's
```

- ② Set the 3 left columns to be purple.

```
#To make purple, we'll keep red and blue at 100% and turn green to 0%
logoImg[:, :3, 1] = 0 #Turn the green to 0 for first 3 columns
```

- ③ Set the 3 right columns to be purple.

```
logoImg[:, -3:, 1] = 0 #Turn the green to 0 for last 3 columns
```

- ④ Set the middle 2 rows to be purple.

```
logoImg[4:6, :, 1] = 0 #Turn the green to 0 for middle rows
```

- ⑤ Save logo array to file.

```
plt.imsave("logo.png", logoImg) #Save the image to logo.png
```

Today's Topics



- Recap: Colors
- Indexing and Slicing
- Design Question: Hunter Logo
- **Decisions**
- CS Survey

In Pairs or Triples...

Predict what these will do (novel concepts):

```
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')
```

```
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 == '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 == 'g':          #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)
```

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)

IDLE

```
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 == '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 == 'g':          #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)
```

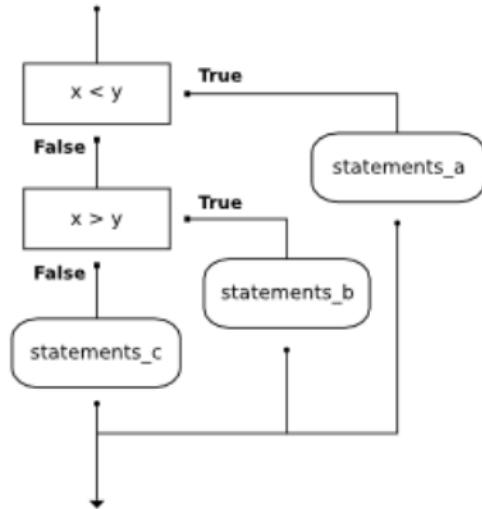
(Demo with IDLE)

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")
```

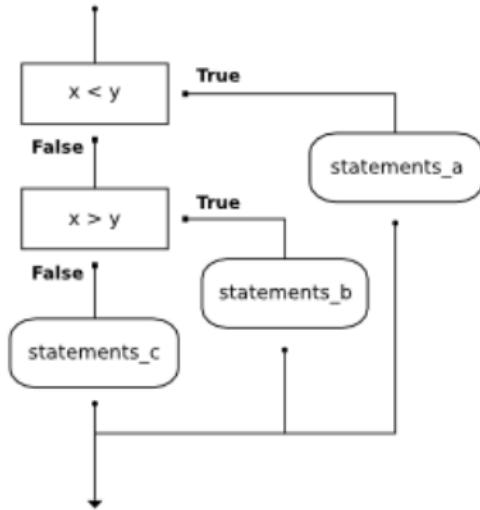
Decisions

```
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")
```



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.)

Today's Topics



- Recap: Colors
- Indexing and Slicing
- Design Question: Hunter Logo
- Decisions
- **CS Survey**

CS Survey: Prof. Sakas, Computational Linguistics



Language is Hard for Computers

Learning Language is Easy for my 3-year-old twins

CSCI 12700 Guest Bullet Talk

William Gregory Sakas



M.A./Ph.D. Program in Linguistics
@ The City University of New York





Language is Hard

- *Buffalo buffalo, Buffalo buffalo buffalo, buffalo, Buffalo buffalo*
- *Someone shot the servant of the actress who was on the balcony. Who was on the balcony?*
- *Who do you think Mary kissed?*
- *Who do you think **that** Mary kissed?*
- *Who do you think bought a radio?*
- * *Who do you think **that** bought a radio?*





So how to explain language?

Treat Language as a **scientific field - like Physics.**

Example: A scientific principle about sentences:

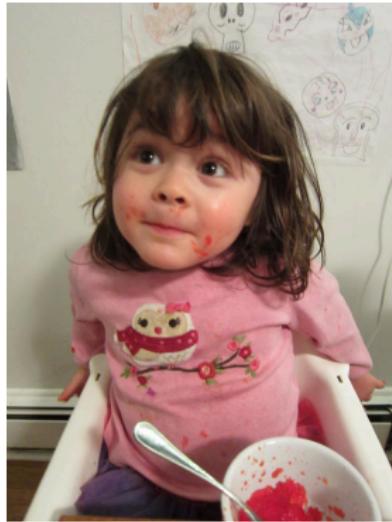
Given $\langle p \rangle = [\alpha [H \beta]]$,
where $\alpha = \text{edge}(\text{Spec}'s)$ β then:
the head H of $\langle p \rangle$ is inert after the phase is completed, triggering no further grammatical operations.

Language is complex!!!
Understanding how language works is hard!!!

Unless you're 3.



CS Survey: Prof. Sakas, Computational Linguistics



Linguistic experts!

4



Lecture Slip



Linguistic experts!

Design a program that counts the number of plural nouns in a list of nouns. Think about:

- what the input is,
- what the output is, and
- how you can determine if a noun is plural.

Note: To simplify the problem, assume all plural nouns end in “s”.

Recap

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



Recap

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



Recap



- 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: Hunter Logo
 - ▶ Decisions

Recap



- 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: Hunter Logo
 - ▶ Decisions
- Pass your lecture slips to the aisles for the UTAs to collect.

Recap



- 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: Hunter Logo
 - ▶ Decisions
- Pass your lecture slips to the aisles for the UTAs to collect.

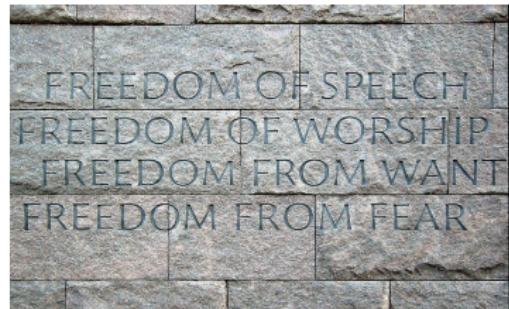
Practice Quiz & Final Questions



(NYTimes)



(Hunter College)



(FDR 4 FP)

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

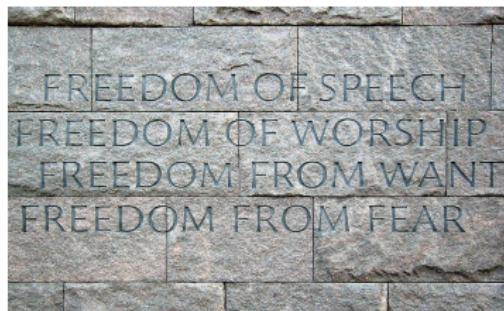
Practice Quiz & Final Questions



(NYTimes)



(Hunter College)



(FDR 4 FP)

- 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).

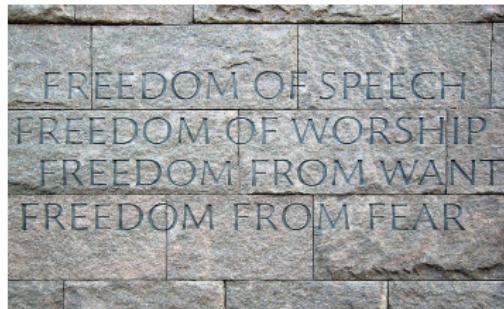
Practice Quiz & Final Questions



(NYTimes)



(Hunter College)



(FDR 4 FP)

- 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:

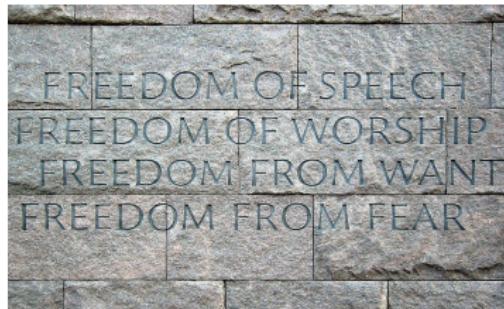
Practice Quiz & Final Questions



(NYTimes)



(Hunter College)



(FDR 4 FP)

- 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;

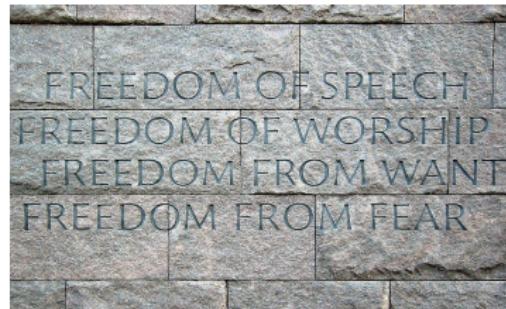
Practice Quiz & Final Questions



(NYTimes)



(Hunter College)



(FDR 4 FP)

- 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

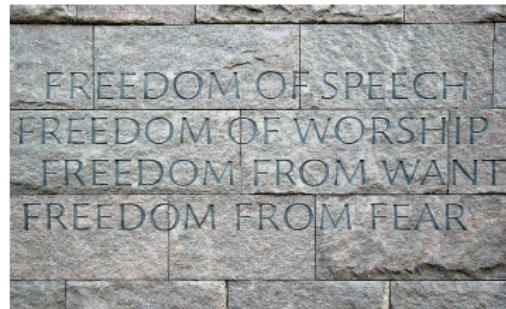
Practice Quiz & Final Questions



(NYTimes)



(Hunter College)



(FDR 4 FP)

- 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.

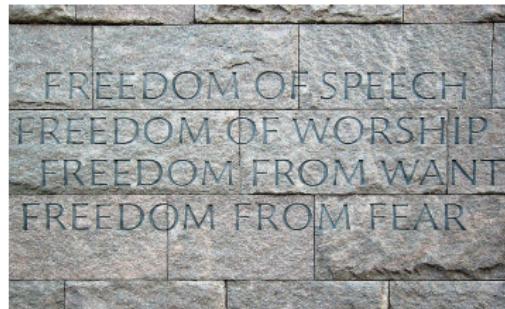
Practice Quiz & Final Questions



(NYTimes)



(Hunter College)



(FDR 4 FP)

- 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](#)).

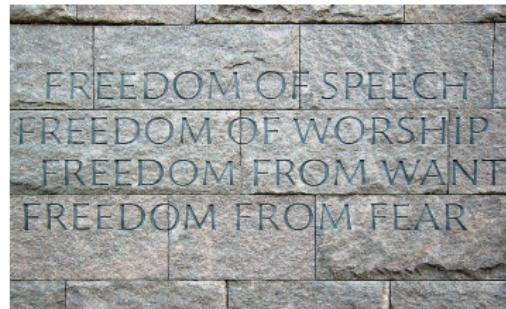
Practice Quiz & Final Questions



(NYTimes)



(Hunter College)



(FDR 4 FP)

- 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.

Writing Boards



- Return writing boards as you leave...