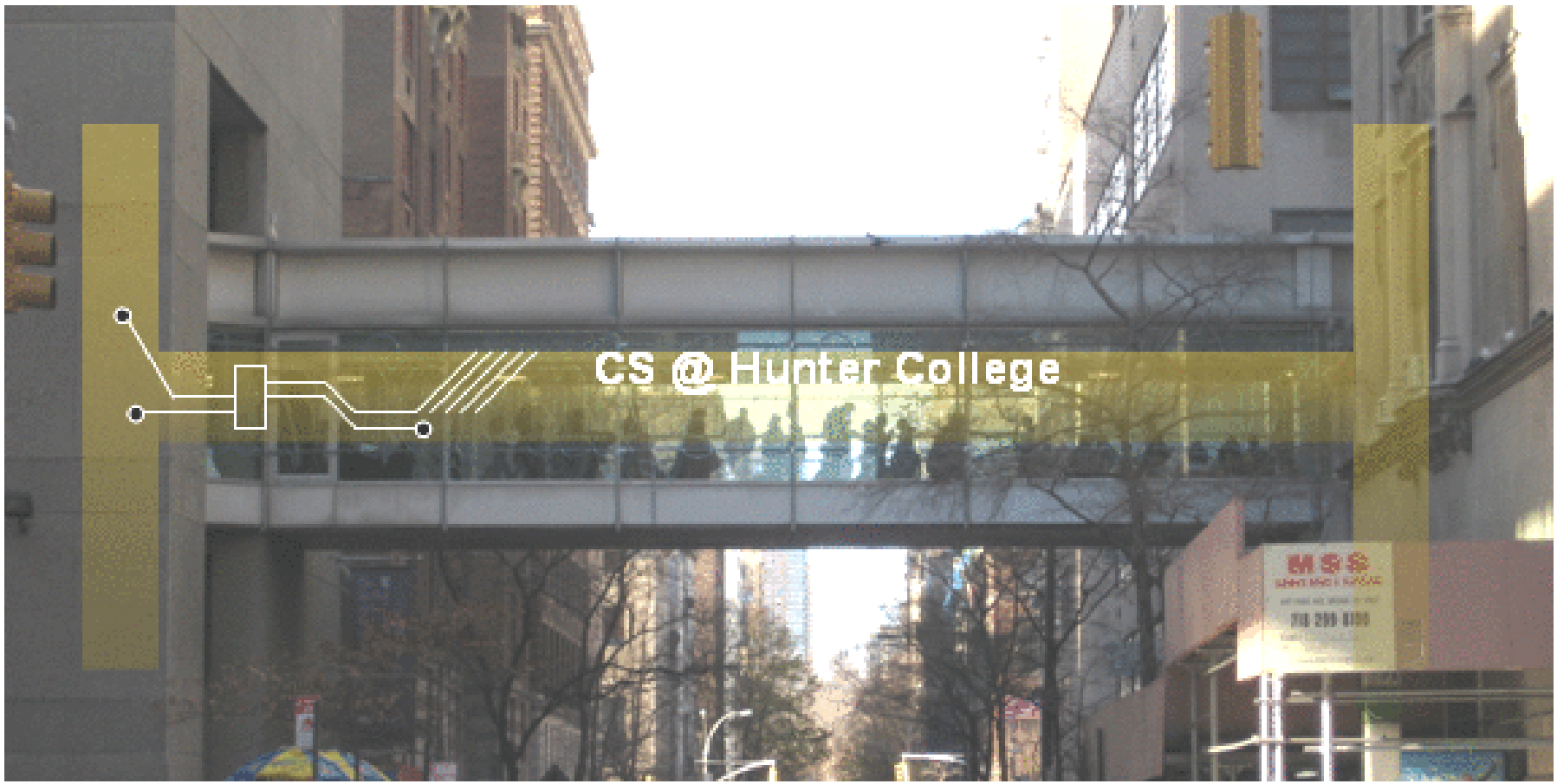
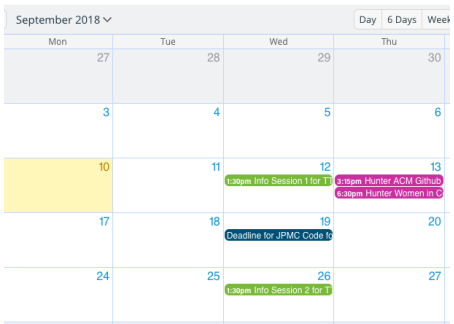
CSci 127: Introduction to Computer Science



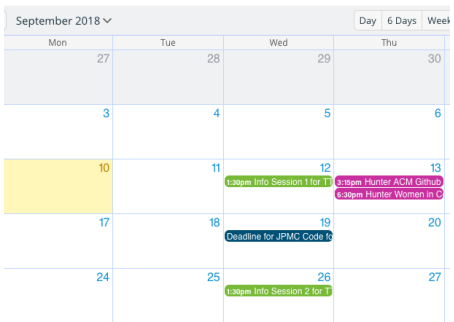
hunter.cuny.edu/csci

Announcements

- A plea from those who scan/grade:



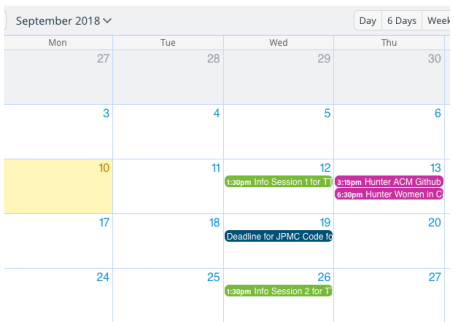
Announcements



- A plea from those who scan/grade:

  If we can't read it, we can't give you credit.
  Please use dark colored ink & write legibly.

Announcements



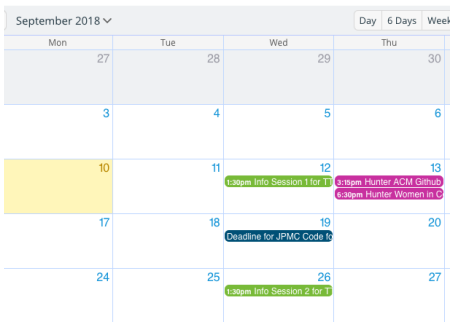
- A plea from those who scan/grade:

  If we can't read it, we can't give you credit.
  Please use dark colored ink & write legibly.

- CS Survey:

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

Announcements



- A plea from those who scan/grade:

  If we can't read it, we can't give you credit.
  Please use dark colored ink & write legibly.

- CS Survey:

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

- Guest Lecturer: Dr. Tiziana Ligorio

Frequently Asked Questions

From lecture slips & recitation sections.

Frequently Asked Questions

From lecture slips & recitation sections.

- Where is the final? When are we taking it?

Frequently Asked Questions

From lecture slips & recitation sections.

- Where is the final? When are we taking it?
  Tuesday, 21 May, 9-11am, 118 North.

Frequently Asked Questions

From lecture slips & recitation sections.

- Where is the final? When are we taking it?
  Tuesday, 21 May, 9-11am, 118 North.
- Can we do more on colors, images, numpy & matplotlib?

Frequently Asked Questions

From lecture slips & recitation sections.

- Where is the final? When are we taking it?
  Tuesday, 21 May, 9-11am, 118 North.
- Can we do more on colors, images, numpy & matplotlib?
  Yes, we will in Labs 6-9 & Lectures 6-9.
  Today, we'll focus on decisions, and logical expressions & circuits.

Frequently Asked Questions

From lecture slips & recitation sections.

- Where is the final? When are we taking it?
  Tuesday, 21 May, 9-11am, 118 North.
- Can we do more on colors, images, numpy & matplotlib?
  Yes, we will in Labs 6-9 & Lectures 6-9.
  Today, we'll focus on decisions, and logical expressions & circuits.
- What is pseudocode? Why do we use it?

Frequently Asked Questions

From lecture slips & recitation sections.

- Where is the final? When are we taking it?
  Tuesday, 21 May, 9-11am, 118 North.
- Can we do more on colors, images, numpy & matplotlib?
  Yes, we will in Labs 6-9 & Lectures 6-9.
  Today, we'll focus on decisions, and logical expressions & circuits.
- What is pseudocode? Why do we use it?
  Pseudocode is the "informal high-level description of the operating principle of a computer program or other algorithm."

Frequently Asked Questions

From lecture slips & recitation sections.

- Where is the final? When are we taking it?
  Tuesday, 21 May, 9-11am, 118 North.
- Can we do more on colors, images, numpy & matplotlib?
  Yes, we will in Labs 6-9 & Lectures 6-9.
  Today, we'll focus on decisions, and logical expressions & circuits.
- What is pseudocode? Why do we use it?
  Pseudocode is the "informal high-level description of the operating principle of a computer program or other algorithm."
  We use it to write down the ideas, before getting deep into the details.

Frequently Asked Questions

From lecture slips & recitation sections.

- Where is the final? When are we taking it?
  Tuesday, 21 May, 9-11am, 118 North.
- Can we do more on colors, images, numpy & matplotlib?
  Yes, we will in Labs 6-9 & Lectures 6-9.
  Today, we'll focus on decisions, and logical expressions & circuits.
- What is pseudocode? Why do we use it?
  Pseudocode is the "informal high-level description of the operating principle of a computer program or other algorithm."
  We use it to write down the ideas, before getting deep into the details.
- What are types of variables?

Frequently Asked Questions

From lecture slips & recitation sections.

- Where is the final? When are we taking it?
  Tuesday, 21 May, 9-11am, 118 North.
- Can we do more on colors, images, numpy & matplotlib?
  Yes, we will in Labs 6-9 & Lectures 6-9.
  Today, we'll focus on decisions, and logical expressions & circuits.
- What is pseudocode? Why do we use it?
  Pseudocode is the "informal high-level description of the operating principle of a computer program or other algorithm."
  We use it to write down the ideas, before getting deep into the details.
- What are types of variables?
  Different kinds of information takes different amounts of space.
  Types we have seen so far: `int`, `float`, `str` *and objects (e.g. turtles).*

Frequently Asked Questions

From lecture slips & recitation sections.

- Where is the final? When are we taking it?
  Tuesday, 21 May, 9-11am, 118 North.
- Can we do more on colors, images, numpy & matplotlib?
  Yes, we will in Labs 6-9 & Lectures 6-9.
  Today, we'll focus on decisions, and logical expressions & circuits.
- What is pseudocode? Why do we use it?
  Pseudocode is the "informal high-level description of the operating principle of a computer program or other algorithm."
  We use it to write down the ideas, before getting deep into the details.
- What are types of variables?
  Different kinds of information takes different amounts of space.
  Types we have seen so far: `int`, `float`, `str` *and objects (e.g. turtles).*
- How can I tell strings from variables?

Frequently Asked Questions

From lecture slips & recitation sections.

- Where is the final? When are we taking it?
  Tuesday, 21 May, 9-11am, 118 North.
- Can we do more on colors, images, numpy & matplotlib?
  Yes, we will in Labs 6-9 & Lectures 6-9.
  Today, we'll focus on decisions, and logical expressions & circuits.
- What is pseudocode? Why do we use it?
  Pseudocode is the "informal high-level description of the operating principle of a computer program or other algorithm."
  We use it to write down the ideas, before getting deep into the details.
- What are types of variables?
  Different kinds of information takes different amounts of space.
  Types we have seen so far: `int`, `float`, `str` *and objects (e.g. turtles).*
- How can I tell strings from variables?
  Strings are surrounded by quotes (either single or double).

Frequently Asked Questions

From lecture slips & recitation sections.

- Where is the final? When are we taking it?
  Tuesday, 21 May, 9-11am, 118 North.
- Can we do more on colors, images, numpy & matplotlib?
  Yes, we will in Labs 6-9 & Lectures 6-9.
  Today, we'll focus on decisions, and logical expressions & circuits.
- What is pseudocode? Why do we use it?
  Pseudocode is the "informal high-level description of the operating principle of a computer program or other algorithm."
  We use it to write down the ideas, before getting deep into the details.
- What are types of variables?
  Different kinds of information takes different amounts of space.
  Types we have seen so far: `int`, `float`, `str` *and objects (e.g. turtles).*
- How can I tell strings from variables?
  Strings are surrounded by quotes (either single or double).
  Variables names (identifiers) for memory locations are not.

Frequently Asked Questions

From lecture slips & recitation sections.

- Where is the final? When are we taking it?
  Tuesday, 21 May, 9-11am, 118 North.
- Can we do more on colors, images, numpy & matplotlib?
  Yes, we will in Labs 6-9 & Lectures 6-9.
  Today, we'll focus on decisions, and logical expressions & circuits.
- What is pseudocode? Why do we use it?
  Pseudocode is the "informal high-level description of the operating principle of a computer program or other algorithm."
  We use it to write down the ideas, before getting deep into the details.
- What are types of variables?
  Different kinds of information takes different amounts of space.
  Types we have seen so far: `int`, `float`, `str` *and objects (e.g. turtles).*
- How can I tell strings from variables?
  Strings are surrounded by quotes (either single or double).
  Variables names (identifiers) for memory locations are not. Ex: `'num'` *vs.* `num`.

Today's Topics

- Recap: Indexing, Slicing, & Decisions
- Logical Expressions
- Circuits
- CS Survey

Today's Topics

- **Recap: Indexing, Slicing, & Decisions**
- Logical Expressions
- Circuits
- CS Survey

Recap: Linguistics Challenge

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: Linguistics Challenge

Linguistic experts!

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

- **Input:**
- **Ouput:**
- how you can determine if a noun is plural.

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

Recap: Linguistics Challenge

Linguistic experts!

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

- **Input:** *A list of nouns*
- **Ouput:**
- how you can determine if a noun is plural.

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

Recap: Linguistics Challenge

Linguistic experts!

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

- **Input:** *A list of nouns*
- **Ouput:** *The number of plural nouns*
- how you can determine if a noun is plural.

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

Recap: Linguistics Challenge

```
nouns = "hats coats glasses scarves"
```

Linguistic experts!

Recap: Linguistics Challenge

```
nouns = "hats coats glasses scarves"
```

Linguistic experts!

How you can determine if a noun is plural?

Recap: Linguistics Challenge

Linguistic experts!

```
nouns = "hats coats glasses scarves"
```

How you can determine if a noun is plural?

- Ends in a 's'.

Recap: Linguistics Challenge

Linguistic experts!

```
nouns = "hats coats glasses scarves"
```

How you can determine if a noun is plural?

- Ends in a 's'.
- If you count 's', you will get too many:

Recap: Linguistics Challenge

Linguistic experts!

```
nouns = "hats coats glasses scarves"
```

How you can determine if a noun is plural?

- Ends in a 's'.
- If you count 's', you will get too many: `print(nouns.count('s'))`

Recap: Linguistics Challenge

Linguistic experts!

```
nouns = "hats coats glasses scarves"
```

How you can determine if a noun is plural?

- Ends in a 's'.
- If you count 's', you will get too many: `print(nouns.count('s'))`

Recap: Linguistics Challenge

```
nouns = "hats coats glasses scarves"
```

Linguistic experts!

Recap: Linguistics Challenge

```
nouns = "hats coats glasses scarves"
```

Linguistic experts!

How you can determine when a word ends?

Recap: Linguistics Challenge

Linguistic experts!

```
nouns = "hats coats glasses scarves"
```

How you can determine when a word ends?

- There's spaces in between.

Recap: Linguistics Challenge

Linguistic experts!

```
nouns = "hats coats glasses scarves"
```

How you can determine when a word ends?

- There's spaces in between.
- To count words:

Recap: Linguistics Challenge

Linguistic experts!

```
nouns = "hats coats glasses scarves"
```

How you can determine when a word ends?

- There's spaces in between.
- To count words:

```
print(nouns.count(' ')+1)
```

Recap: Linguistics Challenge

Linguistic experts!

```
nouns = "hats_coats_glasses_scarves"
```

How you can determine when a word ends?

- There's spaces in between.
- To count words:

```
print(nouns.count(' ')+1)
```

Recap: Linguistics Challenge

```
nouns = "hats coats glasses scarves"
```

Linguistic experts!

Recap: Linguistics Challenge

```
nouns = "hats coats glasses scarves"
```

Linguistic experts!

When a word end with an 's'?

Recap: Linguistics Challenge

Linguistic experts!

```
nouns = "hats coats glasses scarves"
```

When a word end with an 's'?

- Have the pattern: 's '

Recap: Linguistics Challenge

Linguistic experts!

```
nouns = "hats coats glasses scarves"
```

When a word end with an 's'?

- Have the pattern: 's '
- To count plural words:

Recap: Linguistics Challenge

Linguistic experts!

```
nouns = "hats coats glasses scarves"
```

When a word end with an 's'?

- Have the pattern: 's '
- To count plural words:

```
print(nouns.count('s '))
```

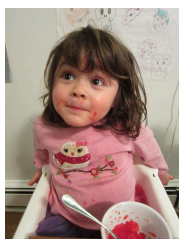
Recap: Linguistics Challenge

```
nouns = "hats coats glasses scarves"
```

When a word end with an 's'?

- Have the pattern: 's '
- To count plural words:

```
print(nouns.count('s '))
```




Linguistic experts!

Recap: Linguistics Challenge

Linguistic experts!

```
nouns = "hats coats glasses scarves"
```

When a word end with an 's'?

- Have the pattern: 's '
- To count plural words:

```
print(nouns.count('s '))
```

- Not quite right– missing `scarves` since no space at the end.

Recap: Linguistics Challenge

Linguistic experts!

```
nouns = "hats coats glasses scarves"
```

When a word end with an 's' ?

- Have the pattern: 's '
- To count plural words:

```
print(nouns.count('s '))
```

- Not quite right– missing `scarves` since no space at the end.
- To fix this, let's add a space, then count:

```
nouns = nouns + " "
print(nouns.count('s '))
```

Lecture Slip: In Pairs or Triples...

Some review:

1.
```
motto = "Mihi cura futuri"
print(motto[2:4])
print(motto[2:4].upper())
```

2.
```
ER = "The future belongs to those who believe in the beauty of their dreams."
print(ER.upper()[2], ER[13], ER[2], "a", ER[15], ER[14], "r  R.")
```

Recap: Indexing & Slicing

```
motto = "Mihi cura futuri"
print(motto[2:4])
print(motto[2:4].upper())
```

Recap: Indexing & Slicing

```
motto = "Mihi cura futuri"
print(motto[2:4])
print(motto[2:4].upper())
```

M	i	h	i		c	u	r	a		f	u	t	u	r	i

Recap: Indexing & Slicing

```
motto = "Mihi cura futuri"
print(motto[2:4])
print(motto[2:4].upper())
```

M	i	h	i		c	u	r	a		f	u	t	u	r	i
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Recap: Indexing & Slicing

```
motto = "Mihi cura futuri"
print(motto[2:4])
print(motto[2:4].upper())
```

M	i	**h**	**i**		c	u	r	a		f	u	t	u	r	i
0	1	**2**	**3**	4	5	6	7	8	9	10	11	12	13	14	15

Recap: Indexing & Slicing

```
motto = "Mihi cura futuri"
print(motto[2:4])
print(motto[2:4].upper())
```

M	i	**h**	**i**		c	u	r	a		f	u	t	u	r	i
0	1	**2**	**3**	4	5	6	7	8	9	10	11	12	13	14	15

Output:

```
hi
```

Recap: Indexing & Slicing

```
motto = "Mihi cura futuri"
print(motto[2:4])
print(motto[2:4].upper())
```

M	i	**h**	**i**		c	u	r	a		f	u	t	u	r	i
0	1	**2**	**3**	4	5	6	7	8	9	10	11	12	13	14	15

Output:

```
hi
HI
```

Recap: Indexing & Slicing

```
ER = "The future belongs to those who believe in the beauty of their dreams."
print(ER.upper()[2], ER[13], ER[2], "a", ER[15], ER[14], "r  R.")
```

Recap: Indexing & Slicing

```
ER = "The future belongs to those who believe in the beauty of their dreams."
print(ER.upper()[2], ER[13], ER[2], "a", ER[15], ER[14], "r  R.")
```

T	h	e		f	u	t	u	r	e		b	e	l	o	n	g	s
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Recap: Indexing & Slicing

```
ER = "The future belongs to those who believe in the beauty of their dreams."
print(ER.upper()[2], ER[13], ER[2], "a", ER[15], ER[14], "r  R.")
```

T	h	**e**		f	u	t	u	r	e		b	e	l	o	n	g	s
0	1	**2**	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Recap: Indexing & Slicing

```
ER = "The future belongs to those who believe in the beauty of their dreams."
print(ER.upper()[2], ER[13], ER[2], "a", ER[15], ER[14], "r  R.")
```

T	h	**e**		f	u	t	u	r	e		b	e	**l**	o	n	g	s
0	1	**2**	3	4	5	6	7	8	9	10	11	12	**13**	14	15	16	17

Recap: Indexing & Slicing

```
ER = "The future belongs to those who believe in the beauty of their dreams."
print(ER.upper()[2], ER[13], ER[2], "a", ER[15], ER[14], "r  R.")
```

T	h	**e**		f	u	t	u	r	e		b	e	**l**	o	**n**	g	s
0	1	**2**	3	4	5	6	7	8	9	10	11	12	**13**	14	**15**	16	17

Recap: Indexing & Slicing

```
ER = "The future belongs to those who believe in the beauty of their dreams."
print(ER.upper()[2], ER[13], ER[2], "a", ER[15], ER[14], "r  R.")
```

T	h	**e**		f	u	t	u	r	e		b	e	**l**	**o**	**n**	g	s
0	1	**2**	3	4	5	6	7	8	9	10	11	12	**13**	**14**	**15**	16	17

Recap: Indexing & Slicing

```
ER = "The future belongs to those who believe in the beauty of their dreams."
print(ER.upper()[2], ER[13], ER[2], "a", ER[15], ER[14], "r  R.")
```

T	h	**e**		f	u	t	u	r	e		b	e	**l**	**o**	**n**	g	s
0	1	**2**	3	4	5	6	7	8	9	10	11	12	**13**	**14**	**15**	16	17

Output:

Recap: Indexing & Slicing

```
ER = "The future belongs to those who believe in the beauty of their dreams."
print(ER.upper()[2], ER[13], ER[2], "a", ER[15], ER[14], "r  R.")
```

T	h	**e**		f	u	t	u	r	e		b	e	**l**	**o**	**n**	g	s
0	1	**2**	3	4	5	6	7	8	9	10	11	12	**13**	**14**	**15**	16	17

Output:

E l e a n o r R.

Today's Topics

- Recap: Indexing, Slicing, & Decisions
- **Logical Expressions**
- Circuits
- CS Survey

In Pairs or Triples...

Some challenges with types & decisions:

```
#What are the types:
y1 = 2017
y2 = "2018"
print(type(y1))
print(type("y1"))
print(type(2017))
print(type("2017"))
print(type(y2))
print(type(y1/4.0))

x = int(y2) - y1
if x < 0:
    print(y2)
else:
    print(y1)
```

```
cents = 432
dollars = cents // 100
change = cents % 100
if dollars > 0:
    print('$'+str(dollars))
if change > 0:
    quarters = change // 25
    pennies = change % 25
    print(quarters, "quarters")
    print("and", pennies, "pennies")
```

Python Tutor

```
#What are the types:
y1 = 2017
y2 = "2018"
print(type(y1))
print(type("y1"))
print(type(2017))
print(type("2017"))
print(type(y2))
print(type(y1/4.0))

x = int(y2) - y1
if x < 0:
    print(y2)
else:
    print(y1)
```

(Demo with pythonTutor)

Decisions

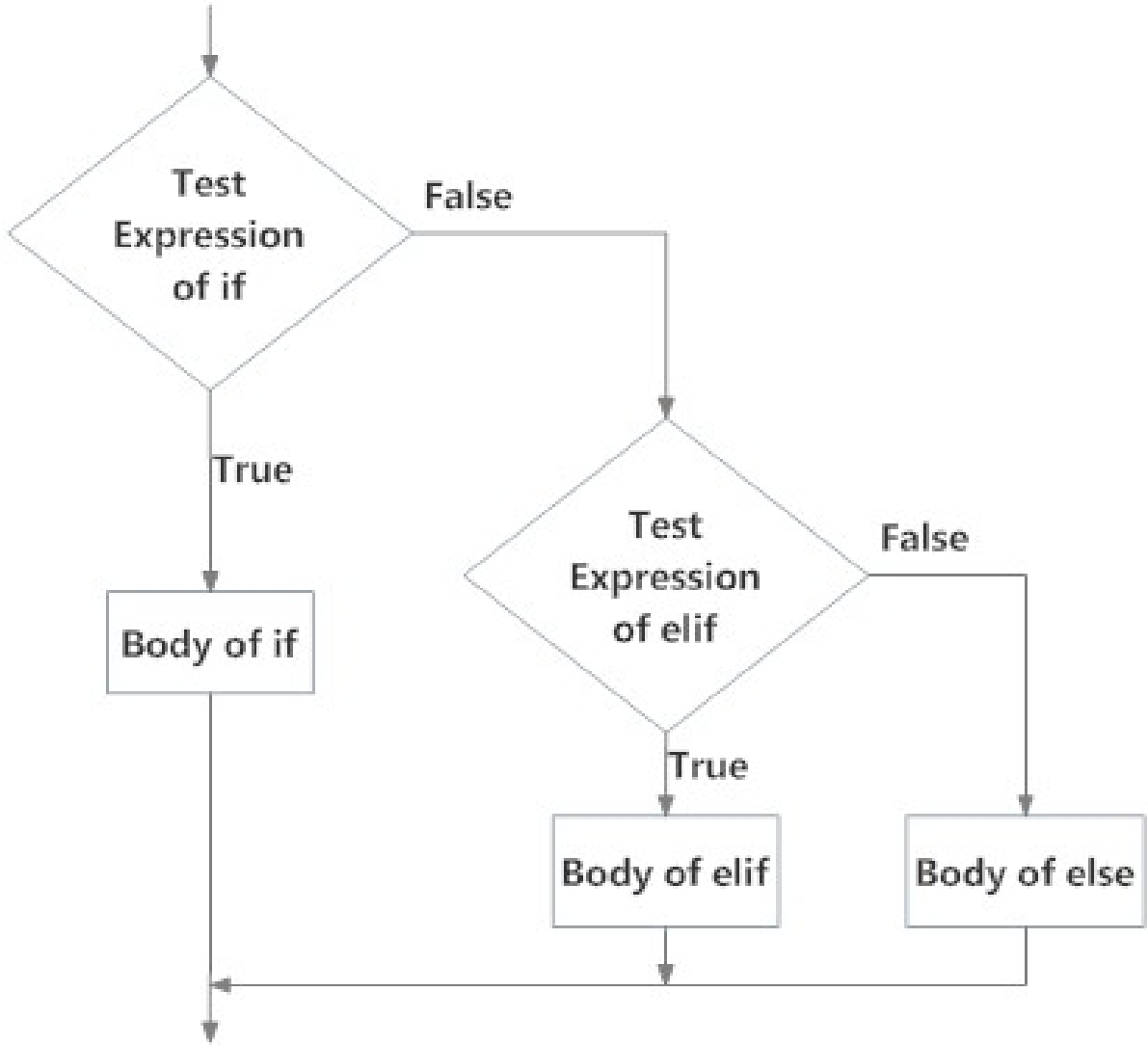
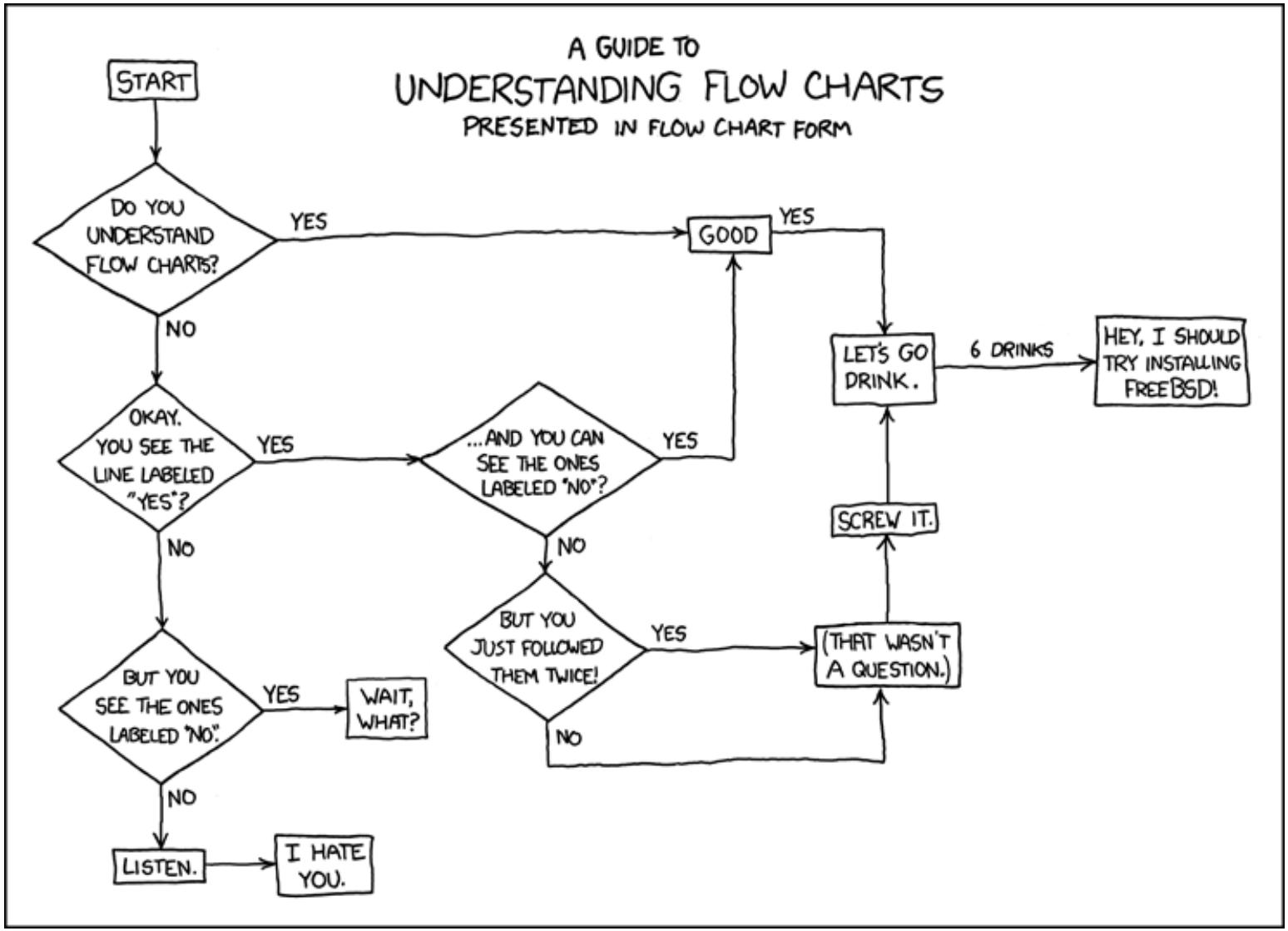


Fig: Operation of if...elif...else statement

(programiz)

Side Note: Reading Flow Charts



(xkcd/518)

In Pairs or Triples

Predict what the code will do:

```
origin = "Indian Ocean"
winds = 100
if (winds > 74):
    print("Major storm, called a ", end="")
    if origin == "Indian Ocean" or origin == "South Pacific":
        print("cyclone.")
    elif origin == "North Pacific":
        print("typhoon.")
    else:
        print("hurricane.")

visibility = 0.2
winds = 40
conditions = "blowing snow"
if (winds > 35) and (visibility < 0.25) and \
      (conditions == "blowing snow" or conditions == "heavy snow"):
    print("Blizzard!")
```

Python Tutor

```
origin = "Indian Ocean"
winds = 100
if (winds > 74):
    print("Major storm, called a ", end="")
    if origin == "Indian Ocean" or origin == "South Pacific":
        print("cyclone.")
    elif origin == "North Pacific":
        print("typhoon.")
    else:
        print("hurricane.")

visibility = 0.2
winds = 40
conditions = "blowing snow"
if (winds > 35) and (visibility < 0.25) and \
     (conditions == "blowing snow" or conditions == "heavy snow"):
    print("Blizzard!")
```

(Demo with `pythonTutor`)

Logical Operators

and

in1		in2	*returns:*
`False`	`and`	`False`	`False`
`False`	`and`	`True`	`False`
`True`	`and`	`False`	`False`
`True`	`and`	`True`	`True`

Logical Operators

and

in1		in2	*returns:*
`False`	`and`	`False`	`False`
`False`	`and`	`True`	`False`
`True`	`and`	`False`	`False`
`True`	`and`	`True`	`True`

or

in1		in2	*returns:*
`False`	`or`	`False`	`False`
`False`	`or`	`True`	`True`
`True`	`or`	`False`	`True`
`True`	`or`	`True`	`True`

Logical Operators

and

in1		in2	*returns:*
`False`	`and`	`False`	`False`
`False`	`and`	`True`	`False`
`True`	`and`	`False`	`False`
`True`	`and`	`True`	`True`

or

in1		in2	*returns:*
`False`	`or`	`False`	`False`
`False`	`or`	`True`	`True`
`True`	`or`	`False`	`True`
`True`	`or`	`True`	`True`

not

	in1	*returns:*
`not`	`False`	`True`
`not`	`True`	`False`

In Pairs or Triples

Predict what the code will do:

```
semHours = 18
reqHours = 120
if semHours >= 12:
    print('Full Time')
else:
    print('Part Time')

pace = reqHours // semHours
if reqHours % semHours != 0:
    pace = pace + 1
print('At this pace, you will graduate in', pace, 'semesters,')
yrs = pace / 2
print('(or', yrs, 'years).')


for i in range(1,20):
    if (i > 10) and (i % 2 == 1):
        print('oddly large')
    else:
        print(i)
```

Python Tutor

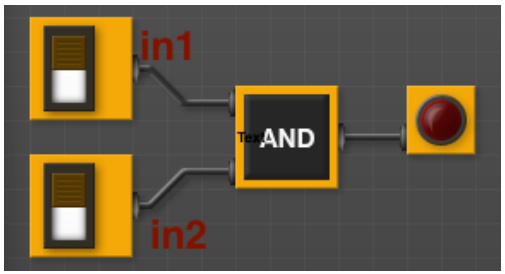
```
semHours = 18
reqHours = 120
if semHours >= 12:
     print('Full Time')
else:
     print('Part Time')

pace = reqHours // semHours
if reqHours % semHours != 0:
     pace = pace + 1
print('At this pace, you will graduate in', pace, 'semesters,')
yrs = pace / 2
print('(or', yrs, 'years).')


for i in range(1,20):
     if (i > 10) and (i % 2 == 1):
          print('oddly large')
     else:
          print(i)
```

(Demo with pythonTutor)

Today's Topics

- Recap: Indexing, Slicing, & Decisions
- Logical Expressions
- **Circuits**
- CS Survey

Circuit Demo

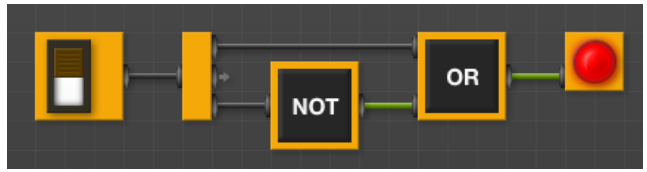


(Demo with `neuroproductions`)

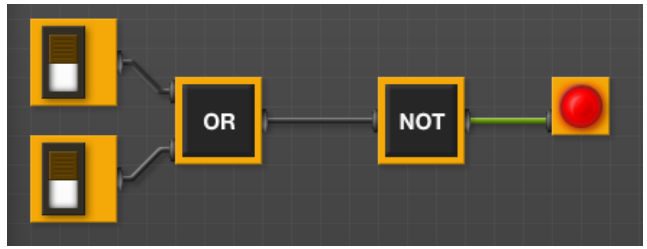
In Pairs or Triples

Predict when these expressions are true:

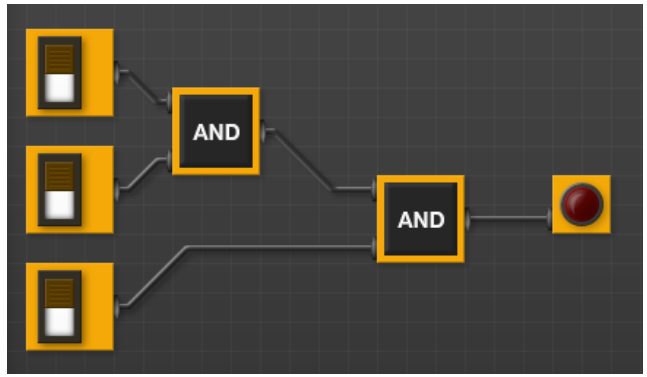
- `in1 or not in1`:



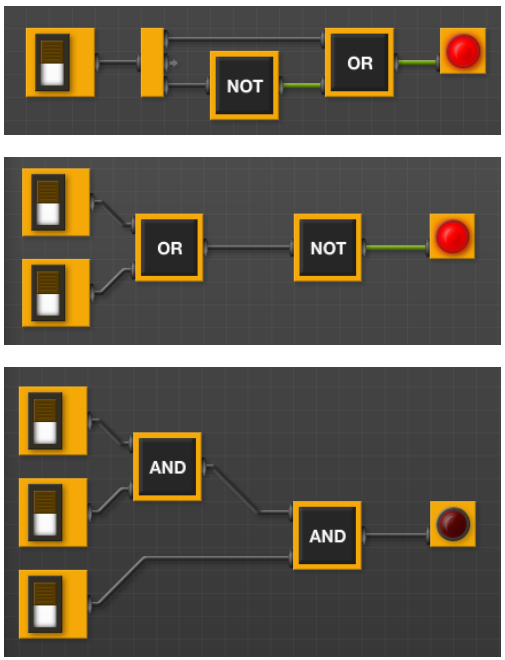
- `not(in1 or in2)`:



- `(in1 and in2) and in3`:



Circuit Demo



(Demo with `neuroproductions`)

Today's Topics

- Recap: Indexing, Slicing, & Decisions
- Logical Expressions
- Circuits
- **CS Survey**

CS Survey Talk: CUNY2X & TTP @Hunter

Bernard Desert & Elise Harris

CS Survey Talk: CUNY2X & TTP @Hunter

Bernard Desert & Elise Harris

- Brief overview of CUNY 2X & Tech Talent Pipeline

CS Survey Talk: CUNY2X & TTP @Hunter

Bernard Desert & Elise Harris

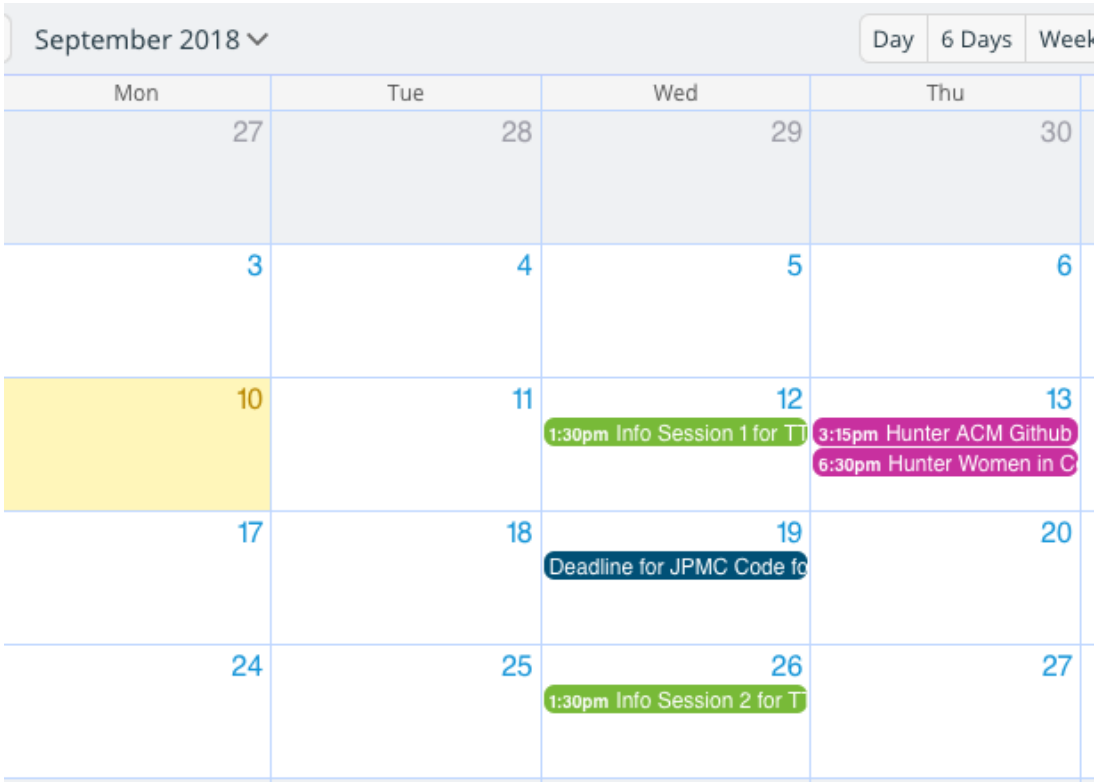
- Brief overview of CUNY 2X & Tech Talent Pipeline
- What Bernard & Elise love about their jobs.

CS Survey Talk: CUNY2X & TTP @Hunter

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.

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.

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:

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:
  1

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:
  1
  2

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:
  1
  2
  Fizz

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:
  1
  2
  Fizz
  4

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:
  1
  2
  Fizz
  4
  Buzz

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:
  1
  2
  Fizz
  4
  Buzz
  5

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:
  1
  2
  Fizz
  4
  Buzz
  5
  Fizz

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:
  1
  2
  Fizz
  4
  Buzz
  5
  Fizz
  7

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

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

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”.
- To Do List:

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”.
- To Do List:
  - Create a loop that goes from 1 to 100.

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”.
- To Do List:
  - Create a loop that goes from 1 to 100.
  - If the number is divisible by 3, print “Fizz”.

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”.
- 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”.

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”.
- 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”.

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”.
- 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.**

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”.
- 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!

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”.
- To Do List (**Reordered**):

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”.
- 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”.

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”.
- 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).

Tech Interview Classic

- 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).

Tech Interview Classic

- 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):

Tech Interview Classic

- 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):
```

Tech Interview Classic

- 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:
```

Tech Interview Classic

- 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="")
```

Tech Interview Classic

- 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:
```

Tech Interview Classic

- 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="")
```

Tech Interview Classic

- 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:
```

Tech Interview Classic

- 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="")
```

Tech Interview Classic

- 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()
```

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:
  - Decisions
  - Logical Expressions
  - Circuits

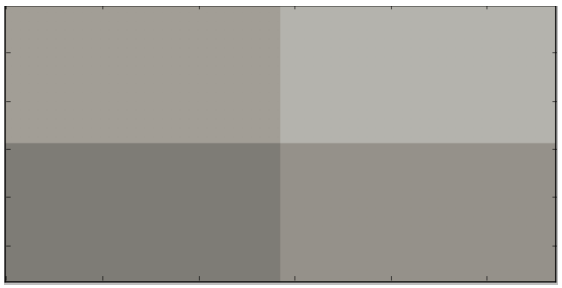
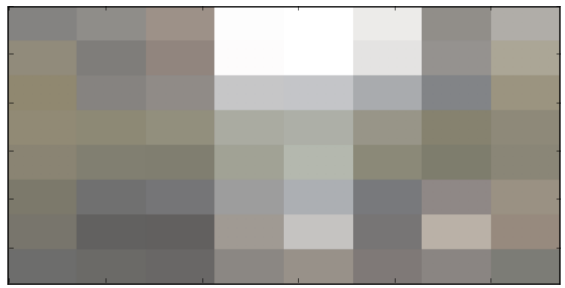
Recap

- On lecture slip, write down a topic you wish we had spent more time (and why).
- In Python, we introduced:
  - Decisions
  - Logical Expressions
  - Circuits
- Pass your lecture slips to the aisles for the UTAs to collect.

Recap

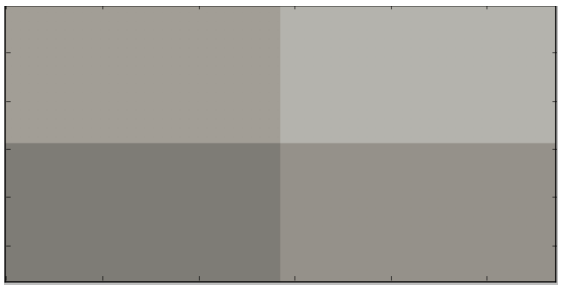
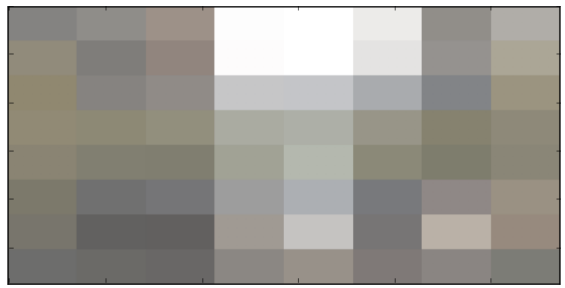
- On lecture slip, write down a topic you wish we had spent more time (and why).
- In Python, we introduced:
  - Decisions
  - Logical Expressions
  - Circuits
- Pass your lecture slips to the aisles for the UTAs to collect.

Practice Quiz & Final Questions

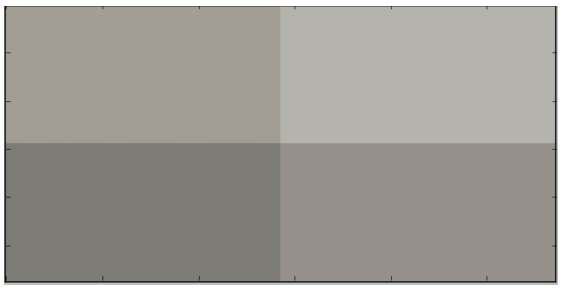
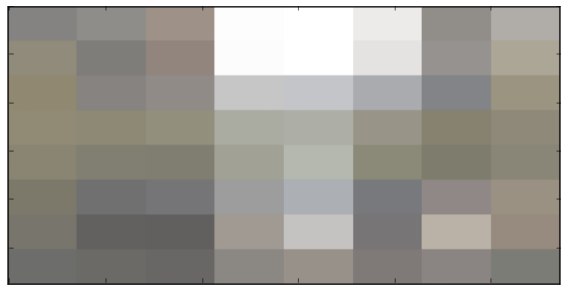

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.

Practice Quiz & Final Questions

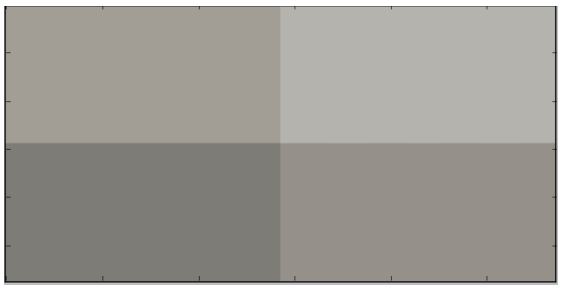
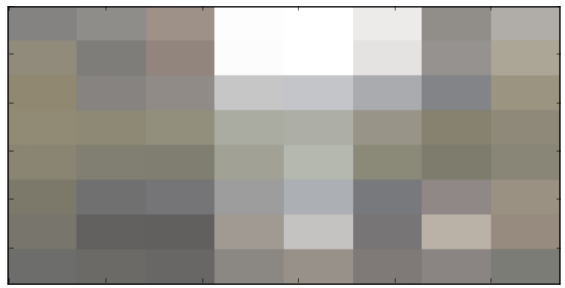

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).

Practice Quiz & Final Questions

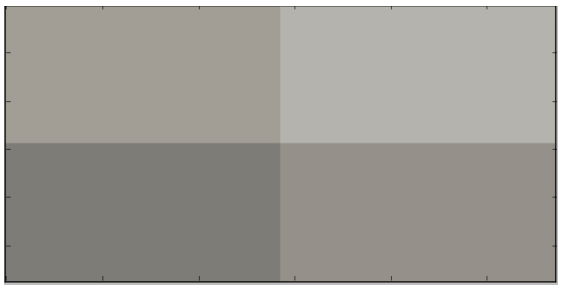
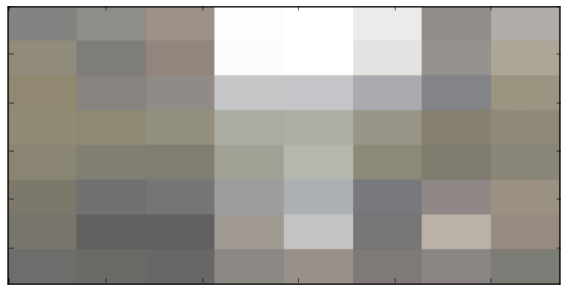

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:

Practice Quiz & Final Questions

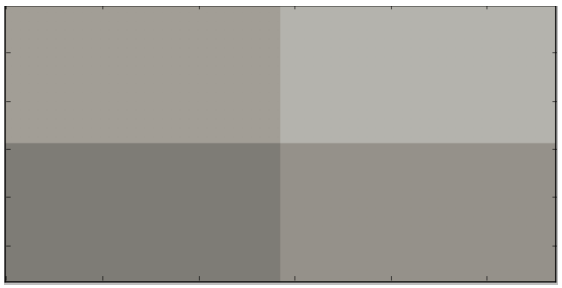
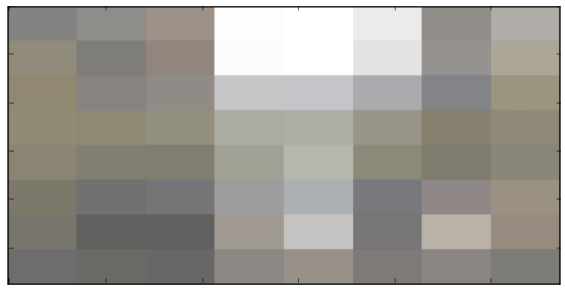

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;

Practice Quiz & Final Questions

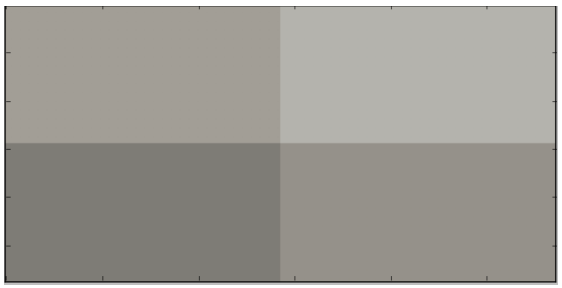
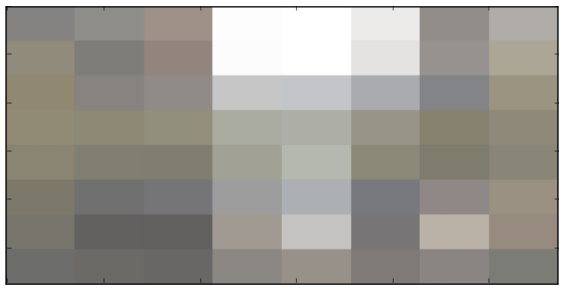
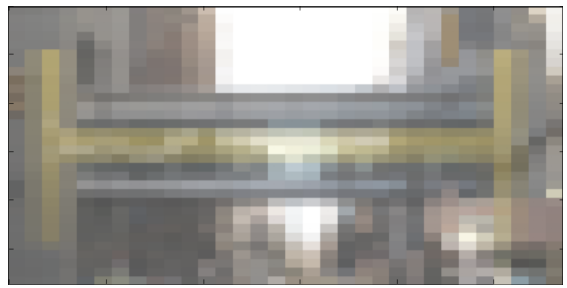

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and

Practice Quiz & Final Questions

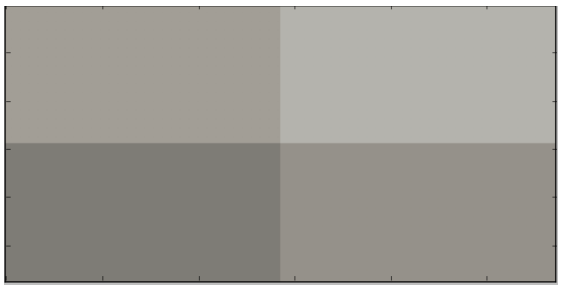
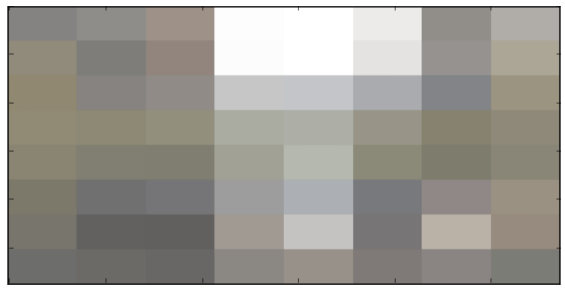

- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - repeat.

Practice Quiz & Final Questions

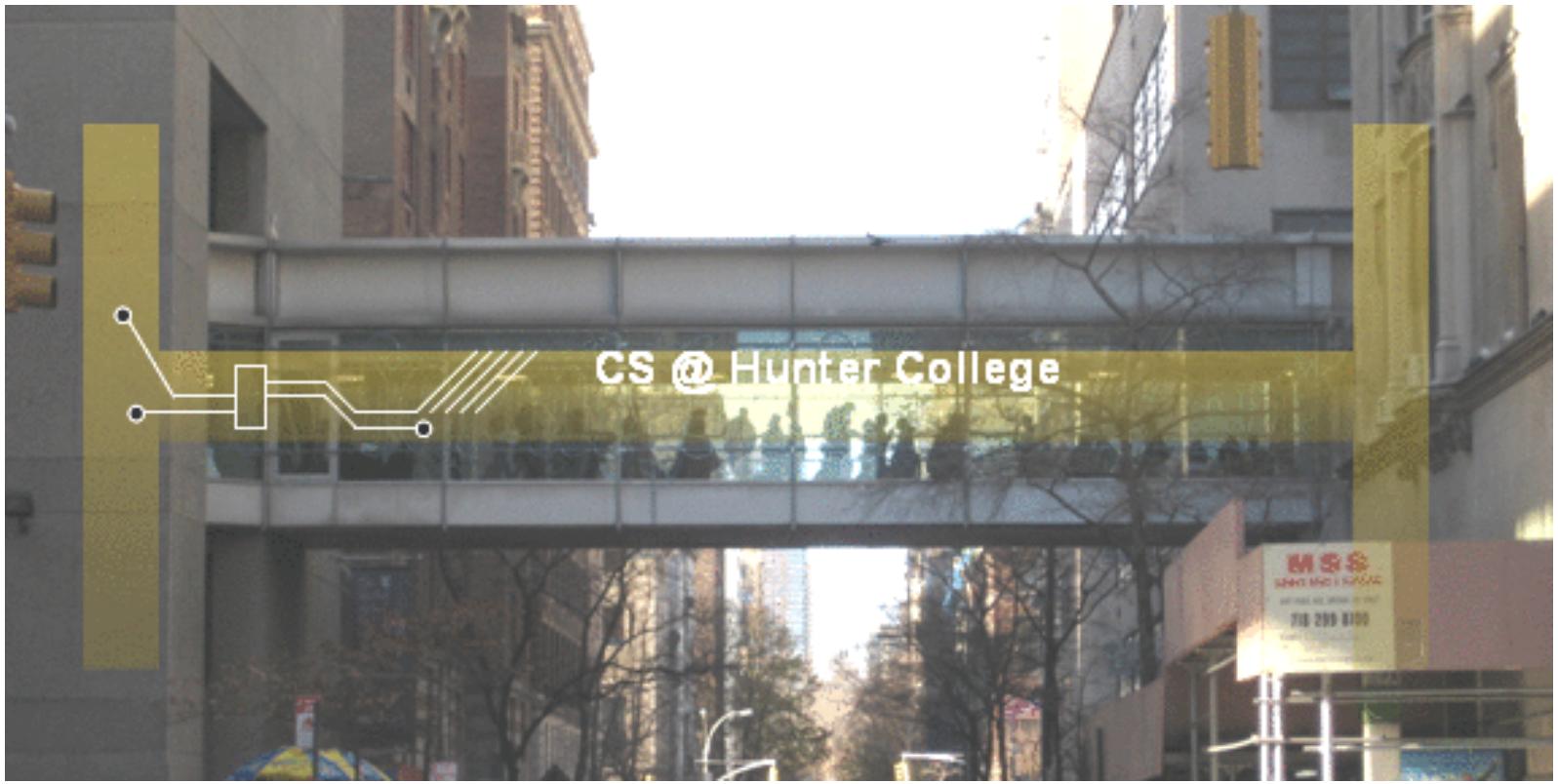
- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - repeat.
- Past exams are on the webpage (under Final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

Practice Quiz & Final Questions


- 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 Spring 2018, Version 1.

Writing Boards



- Return writing boards as you leave...