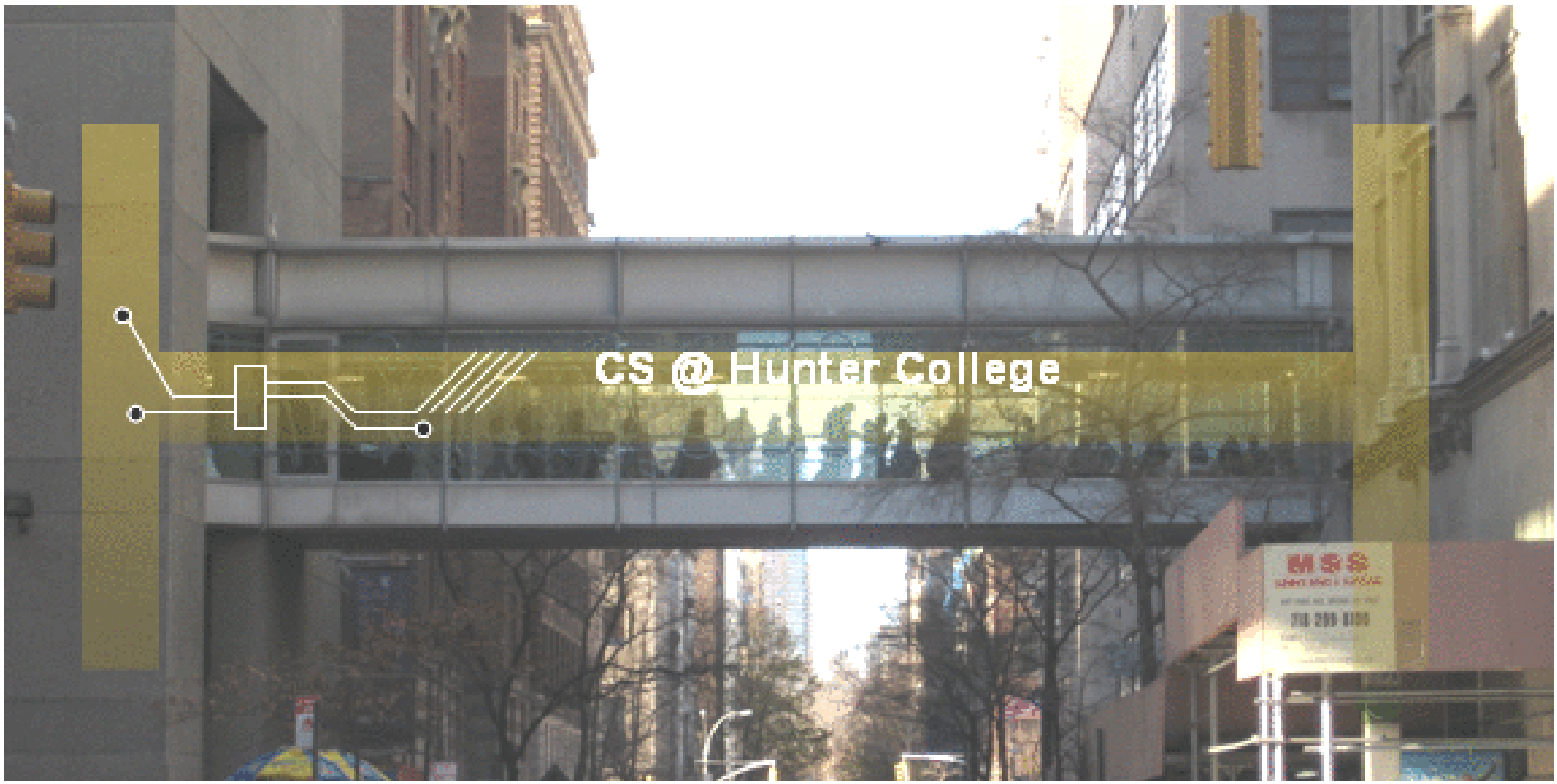
CSci 127: Introduction to Computer Science



hunter.cuny.edu/csci

Announcements

- Each lecture includes a survey of computing research and tech in NYC.

  Today: Keith Okrosy
  Career Development Services

Frequently Asked Questions

From lecture slips & recitation sections.

- Can you go through the OpenData challenge from last week?

Frequently Asked Questions

From lecture slips & recitation sections.

- Can you go through the OpenData challenge from last week?
  Yes, we'll start with functions, and then go on to the OpenData challenge.

Frequently Asked Questions

From lecture slips & recitation sections.

- Can you go through the OpenData challenge from last week?
  Yes, we'll start with functions, and then go on to the OpenData challenge.
- Do I have to take the final?
  Yes, you have to pass the final (60 out of 100 points) to the pass the class.

Frequently Asked Questions

From lecture slips & recitation sections.

- Can you go through the OpenData challenge from last week?
  Yes, we'll start with functions, and then go on to the OpenData challenge.
- Do I have to take the final?
  Yes, you have to pass the final (60 out of 100 points) to the pass the class.
- Can I take the course No Credit/Credit?
  Yes, but check with your advisor that it is possible with your major and standing.

Frequently Asked Questions

From lecture slips & recitation sections.

- Can you go through the OpenData challenge from last week?
  Yes, we'll start with functions, and then go on to the OpenData challenge.
- Do I have to take the final?
  Yes, you have to pass the final (60 out of 100 points) to the pass the class.
- Can I take the course No Credit/Credit?
  Yes, but check with your advisor that it is possible with your major and standing.
- To earn a Credit grade, what do I need?

Frequently Asked Questions

From lecture slips & recitation sections.

- Can you go through the OpenData challenge from last week?
  Yes, we'll start with functions, and then go on to the OpenData challenge.
- Do I have to take the final?
  Yes, you have to pass the final (60 out of 100 points) to the pass the class.
- Can I take the course No Credit/Credit?
  Yes, but check with your advisor that it is possible with your major and standing.
- To earn a Credit grade, what do I need?
  - *Final can replace missing lecture slips or quizzes. Programs are 30%.*

Frequently Asked Questions

From lecture slips & recitation sections.

- Can you go through the OpenData challenge from last week?
  Yes, we'll start with functions, and then go on to the OpenData challenge.
- Do I have to take the final?
  Yes, you have to pass the final (60 out of 100 points) to the pass the class.
- Can I take the course No Credit/Credit?
  Yes, but check with your advisor that it is possible with your major and standing.
- To earn a Credit grade, what do I need?
  - *Final can replace missing lecture slips or quizzes. Programs are 30%.*
  - *You need to pass the final, which takes 60 out of 100 points.*

Frequently Asked Questions

From lecture slips & recitation sections.

- Can you go through the OpenData challenge from last week?
  Yes, we'll start with functions, and then go on to the OpenData challenge.
- Do I have to take the final?
  Yes, you have to pass the final (60 out of 100 points) to the pass the class.
- Can I take the course No Credit/Credit?
  Yes, but check with your advisor that it is possible with your major and standing.
- To earn a Credit grade, what do I need?
  - *Final can replace missing lecture slips or quizzes. Programs are 30%.*
  - *You need to pass the final, which takes 60 out of 100 points.*
  - *If final counts 70%, that would be 60% of 70 = 42 points.*

Frequently Asked Questions

From lecture slips & recitation sections.

- Can you go through the OpenData challenge from last week?
  Yes, we'll start with functions, and then go on to the OpenData challenge.
- Do I have to take the final?
  Yes, you have to pass the final (60 out of 100 points) to the pass the class.
- Can I take the course No Credit/Credit?
  Yes, but check with your advisor that it is possible with your major and standing.
- To earn a Credit grade, what do I need?
  - *Final can replace missing lecture slips or quizzes. Programs are 30%.*
  - *You need to pass the final, which takes 60 out of 100 points.*
  - *If final counts 70%, that would be 60% of 70 = 42 points. Need 70 - 42 = 28 points (of 30) on the programs (or 52 programs).*

Frequently Asked Questions

From lecture slips & recitation sections.

- Can you go through the OpenData challenge from last week?
  Yes, we'll start with functions, and then go on to the OpenData challenge.
- Do I have to take the final?
  Yes, you have to pass the final (60 out of 100 points) to the pass the class.
- Can I take the course No Credit/Credit?
  Yes, but check with your advisor that it is possible with your major and standing.
- To earn a Credit grade, what do I need?
  - *Final can replace missing lecture slips or quizzes. Programs are 30%.*
  - *You need to pass the final, which takes 60 out of 100 points.*
  - *If final counts 70%, that would be 60% of 70 = 42 points. Need 70 - 42 = 28 points (of 30) on the programs (or 52 programs).*
  - *With higher final score, you need fewer programs: Final: 80, Programs: 27.*

Frequently Asked Questions

From lecture slips & recitation sections.

- Can you go through the OpenData challenge from last week?
  Yes, we'll start with functions, and then go on to the OpenData challenge.
- Do I have to take the final?
  Yes, you have to pass the final (60 out of 100 points) to the pass the class.
- Can I take the course No Credit/Credit?
  Yes, but check with your advisor that it is possible with your major and standing.
- To earn a Credit grade, what do I need?
  - *Final can replace missing lecture slips or quizzes. Programs are 30%.*
  - *You need to pass the final, which takes 60 out of 100 points.*
  - *If final counts 70%, that would be 60% of 70 = 42 points. Need 70 - 42 = 28 points (of 30) on the programs (or 52 programs).*
  - *With higher final score, you need fewer programs: Final: 80, Programs: 27.*
  - *More lecture slips & quizzes help: 10 lectures slips (5%) and 5 quizzes (10%) leave 50% for the final. Passing final with 60% would need 46 programs for credit. 80% on final, need 28 programs...*

Frequently Asked Questions

From lecture slips & recitation sections.

- Can you go through the OpenData challenge from last week?
  Yes, we'll start with functions, and then go on to the OpenData challenge.
- Do I have to take the final?
  Yes, you have to pass the final (60 out of 100 points) to the pass the class.
- Can I take the course No Credit/Credit?
  Yes, but check with your advisor that it is possible with your major and standing.
- To earn a Credit grade, what do I need?
  - *Final can replace missing lecture slips or quizzes. Programs are 30%.*
  - *You need to pass the final, which takes 60 out of 100 points.*
  - *If final counts 70%, that would be 60% of 70 = 42 points. Need 70 - 42 = 28 points (of 30) on the programs (or 52 programs).*
  - *With higher final score, you need fewer programs: Final: 80, Programs: 27.*
  - *More lecture slips & quizzes help: 10 lectures slips (5%) and 5 quizzes (10%) leave 50% for the final. Passing final with 60% would need 46 programs for credit. 80% on final, need 28 programs...*
  - *Always good to aim a bit higher!*

Today's Topics

- More on Functions
- Recap: Open Data
- Top Down Design
- Github
- CS Survey: Career Services

Today's Topics

- **More on Functions**
- Recap: Open Data
- Top Down Design
- Github
- CS Survey: Career Services

Input Parameters & Return Values

- Functions can have **input parameters**.

```
def totalWithTax(food,tip):
    total = 0
    tax = 0.0875
    total = food + food * tax
    total = total + tip
    return(total)

lunch = float(input('Enter lunch total: '))
lTip = float(input('Enter lunch tip:' ))
lTotal = totalWithTax(lunch, lTip)
print('Lunch total is', lTotal)

dinner= float(input('Enter dinner total: '))
dTip = float(input('Enter dinner tip:' ))
dTotal = totalWithTax(dinner, dTip)
print('Dinner total is', dTotal)
```

Input Parameters & Return Values

```
def totalWithTax(food,tip):
    total = 0
    tax = 0.0875
    total = food + food * tax
    total = total + tip
    return(total)

lunch = float(input('Enter lunch total: '))
lTip = float(input('Enter lunch tip:' ))
lTotal = totalWithTax(lunch, lTip)
print('Lunch total is', lTotal)

dinner= float(input('Enter dinner total: '))
dTip = float(input('Enter dinner tip:' ))
dTotal = totalWithTax(dinner, dTip)
print('Dinner total is', dTotal)
```

- Functions can have **input parameters**.
- Surrounded by parentheses, both in the function definition, and in the function call (invocation).

Input Parameters & Return Values

```
def totalWithTax(food,tip):
    total = 0
    tax = 0.0875
    total = food + food * tax
    total = total + tip
    return(total)

lunch = float(input('Enter lunch total: '))
lTip = float(input('Enter lunch tip:' ))
lTotal = totalWithTax(lunch, lTip)
print('Lunch total is', lTotal)

dinner= float(input('Enter dinner total: '))
dTip = float(input('Enter dinner tip:' ))
dTotal = totalWithTax(dinner, dTip)
print('Dinner total is', dTotal)
```

- Functions can have **input parameters**.
- Surrounded by parentheses, both in the function definition, and in the function call (invocation).
- The "placeholders" in the function definition: **formal parameters**.

Input Parameters & Return Values

```
def totalWithTax(food,tip):
    total = 0
    tax = 0.0875
    total = food + food * tax
    total = total + tip
    return(total)

lunch = float(input('Enter lunch total: '))
lTip = float(input('Enter lunch tip:' ))
lTotal = totalWithTax(lunch, lTip)
print('Lunch total is', lTotal)

dinner= float(input('Enter dinner total: '))
dTip = float(input('Enter dinner tip:' ))
dTotal = totalWithTax(dinner, dTip)
print('Dinner total is', dTotal)
```

- Functions can have **input parameters**.
- Surrounded by parentheses, both in the function definition, and in the function call (invocation).
- The "placeholders" in the function definition: **formal parameters**.
- The ones in the function call: **actual parameters**

Input Parameters & Return Values

```
def totalWithTax(food,tip):
    total = 0
    tax = 0.0875
    total = food + food * tax
    total = total + tip
    return(total)

lunch = float(input('Enter lunch total: '))
lTip = float(input('Enter lunch tip:' ))
lTotal = totalWithTax(lunch, lTip)
print('Lunch total is', lTotal)

dinner= float(input('Enter dinner total: '))
dTip = float(input('Enter dinner tip:' ))
dTotal = totalWithTax(dinner, dTip)
print('Dinner total is', dTotal)
```

- Functions can have **input parameters**.
- Surrounded by parentheses, both in the function definition, and in the function call (invocation).
- The "placeholders" in the function definition: **formal parameters**.
- The ones in the function call: **actual parameters**
- Functions can also **return values** to where it was called.

Input Parameters & Return Values

```
def totalWithTax(food,tip):
    total = 0
    tax = 0.0875
    total = food + food * tax
    total = total + tip
    return(total)

lunch = float(input('Enter lunch total: '))
lTip = float(input('Enter lunch tip:' ))
lTotal = totalWithTax(lunch, lTip)
print('Lunch total is', lTotal)

dinner= float(input('Enter dinner total: '))
dTip = float(input('Enter dinner tip:' ))
dTotal = totalWithTax(dinner, dTip)
print('Dinner total is', dTotal)
```

Formal Parameters

Actual Parameters

- Functions can have **input parameters**.
- Surrounded by parenthesis, both in the function definition, and in the function call (invocation).
- The "placeholders" in the function definition: **formal parameters**.
- The ones in the function call: **actual parameters**.
- Functions can also **return values** to where it was called.

In Pairs or Triples:

- What are the formal parameters? What is returned?

```
def enigma1(x,y,z):
    if x == len(y):
        return(z)
    elif x < len(y):
        return(y[0:x])
    else:
        s = cont1(z)
        return(s+y)
```

```
def cont1(st):
    r = ""
    for i in range(len(st)-1,-1,-1):
        r = r + st[i]
    return(r)
```

(a) `enigma1(7,"caramel","dulce de leche")`

Return: _____

(b) `enigma1(3,"cupcake","vanilla")`

Return: _____

(c) `enigma1(10,"pie","nomel")`

Return: _____

Python Tutor

```
def enigma1(x,y,z):
    if x == len(y):
        return(z)
    elif x < len(y):
        return(y[0:x])
    else:
        s = cont1(z)
        return(s+y)
```

(a) enigma1(7,"caramel","dulce de leche")

Return:

(b) enigma1(3,"cupcake","vanilla")

Return:

(c) enigma1(10,"pie","nomel")

Return:

```
def cont1(st):
    r = ""
    for i in range(len(st)-1,-1,-1):
        r = r + st[i]
    return(r)
```

(Demo with `pythonTutor`)

Input Parameters

```
def totalWithTax(food,tip):
    total = 0
    tax = 0.0875
    total = food + food * tax
    total = total + tip
    return(total)

lunch = float(input('Enter lunch total: '))
lTip = float(input('Enter lunch tip:' ))
lTotal = totalWithTax(lunch, lTip)
print('Lunch total is', lTotal)

dinner= float(input('Enter dinner total: '))
dTip = float(input('Enter dinner tip:' ))
dTotal = totalWithTax(dinner, dTip)
print('Dinner total is', dTotal)
```

Formal Parameters

Actual Parameters

- When called, the actual parameter values are copied to the formal parameters.

Input Parameters

```
def totalWithTax(food,tip):
    total = 0
    tax = 0.0875
    total = food + food * tax
    total = total + tip
    return(total)

lunch = float(input('Enter lunch total: '))
lTip = float(input('Enter lunch tip:' ))
lTotal = totalWithTax(lunch, lTip)
print('Lunch total is', lTotal)

dinner= float(input('Enter dinner total: '))
dTip = float(input('Enter dinner tip:' ))
dTotal = totalWithTax(dinner, dTip)
print('Dinner total is', dTotal)
```

Formal Parameters

Actual Parameters

- When called, the actual parameter values are copied to the formal parameters.
- All the commands inside the function are performed on the copies.

Input Parameters

```
def totalWithTax(food,tip):
    total = 0
    tax = 0.0875
    total = food + food * tax
    total = total + tip
    return(total)

lunch = float(input('Enter lunch total: '))
lTip = float(input('Enter lunch tip:' ))
lTotal = totalWithTax(lunch, lTip)
print('Lunch total is', lTotal)

dinner= float(input('Enter dinner total: '))
dTip = float(input('Enter dinner tip:' ))
dTotal = totalWithTax(dinner, dTip)
print('Dinner total is', dTotal)
```

Formal Parameters

Actual Parameters

- When called, the actual parameter values are copied to the formal parameters.
- All the commands inside the function are performed on the copies.
- The actual parameters do not change.

Input Parameters

```
def totalWithTax(food,tip):
    total = 0
    tax = 0.0875
    total = food + food * tax
    total = total + tip
    return(total)

lunch = float(input('Enter lunch total: '))
lTip = float(input('Enter lunch tip:' ))
lTotal = totalWithTax(lunch, lTip)
print('Lunch total is', lTotal)

dinner= float(input('Enter dinner total: '))
dTip = float(input('Enter dinner tip:' ))
dTotal = totalWithTax(dinner, dTip)
print('Dinner total is', dTotal)
```

Formal Parameters

Actual Parameters

- When called, the actual parameter values are copied to the formal parameters.
- All the commands inside the function are performed on the copies.
- The actual parameters do not change.
- The copies are discarded when the function is done.

Input Parameters

```
def totalWithTax(food,tip):
    total = 0
    tax = 0.0875
    total = food + food * tax
    total = total + tip
    return(total)

lunch = float(input('Enter lunch total: '))
lTip = float(input('Enter lunch tip:' ))
lTotal = totalWithTax(lunch, lTip)
print('Lunch total is', lTotal)

dinner= float(input('Enter dinner total: '))
dTip = float(input('Enter dinner tip:' ))
dTotal = totalWithTax(dinner, dTip)
print('Dinner total is', dTotal)
```

Formal Parameters

Actual Parameters

- When called, the actual parameter values are copied to the formal parameters.
- All the commands inside the function are performed on the copies.
- The actual parameters do not change.
- The copies are discarded when the function is done.
- The time a variable exists is called its **scope**.

Input Parameters: What about Lists?

```
#Fall 2013 Final Exam, 5

def kuwae( inLst ):
    tot = 1
    for item in inLst:
        tot = tot * item
    return tot

def foo( inLst ):
    if ( inLst[-1] > inLst[0] ):
        return kuwae( inLst )
    else:
        return -1

foo( [2, 4, 6, 8] )

foo( [4002, 328, 457, 1] )
```

- When called, the actual parameter values are copied to the formal parameters.

Input Parameters: What about Lists?

```
#Fall 2013 Final Exam, 5

def kuwae( inLst ):
    tot = 1
    for item in inLst:
        tot = tot * item
    return tot

def foo( inLst ):
    if ( inLst[-1] > inLst[0] ):
        return kuwae( inLst )
    else:
        return -1

foo( [2, 4, 6, 8] )

foo( [4002, 328, 457, 1] )
```

- When called, the actual parameter values are copied to the formal parameters.
- What is copied with a list?

Input Parameters: What about Lists?

```
#Fall 2013 Final Exam, 5

def kuwae( inLst ):
    tot = 1
    for item in inLst:
        tot = tot * item
    return tot

def foo( inLst ):
    if ( inLst[-1] > inLst[0] ):
        return kuwae( inLst )
    else:
        return -1

foo( [2, 4, 6, 8] )

foo( [4002, 328, 457, 1] )
```

- When called, the actual parameter values are copied to the formal parameters.
- What is copied with a list?
- The address of the list, but not the individual elements.

Input Parameters: What about Lists?

```
#Fall 2013 Final Exam, 5

def kuwae( inLst ):
    tot = 1
    for item in inLst:
        tot = tot * item
    return tot

def foo( inLst ):
    if ( inLst[-1] > inLst[0] ):
        return kuwae( inLst )
    else:
        return -1

foo( [2, 4, 6, 8] )

foo( [4002, 328, 457, 1] )
```

- When called, the actual parameter values are copied to the formal parameters.
- What is copied with a list?
- The address of the list, but not the individual elements.
- The actual parameters do not change, but the inside elements might.

Input Parameters: What about Lists?

```
#Fall 2013 Final Exam, 5

def kuwae( inLst ):
    tot = 1
    for item in inLst:
        tot = tot * item
    return tot

def foo( inLst ):
    if ( inLst[-1] > inLst[0] ):
        return kuwae( inLst )
    else:
        return -1

foo( [2, 4, 6, 8] )

foo( [4002, 328, 457, 1] )
```

- When called, the actual parameter values are copied to the formal parameters.
- What is copied with a list?
- The address of the list, but not the individual elements.
- The actual parameters do not change, but the inside elements might.
- Easier to see with a demo.

Python Tutor

```
#Fall 2013 Final Exam, 5

def kuwae( inLst ):
    tot = 1
    for item in inLst:
        tot = tot * item
    return tot

def foo( inLst ):
    if ( inLst[-1] > inLst[0] ):
        return kuwae( inLst )
    else:
        return -1

foo( [2, 4, 6, 8] )

foo( [4002, 328, 457, 1] )
```

(Demo with pythonTutor)

In Pairs or Triples:

```
def bar(n):
    if n <= 8:
        return 1
    else:
        return 0

def foo(l):
    n = bar(l[-1])
    return l[n]
```

- What are the formal parameters for the functions?
- What is the output of:

```
r = foo([1,2,3,4])
print("Return:  ", r)
```

- What is the output of:

```
r = foo([1024,512,256,128])
print("Return:  ", r)
```

Python Tutor

```
def bar(n):
    if n <= 8:
        return 1
    else:
        return 0

def foo(l):
    n = bar(l[-1])
    return l[n]
```

(Demo with pythonTutor)

In Pairs or Triples:

Predict what the code will do:

```
#CSci 127 Teaching Staff
#Triangles two ways...
import turtle

def setUp(t, dist, col):
     t.penup()
     t.forward(dist)
     t.pendown()
     t.color(col)

def nestedTriangle(t, side):
     if side > 10:
          for i in range(3):
               t.forward(side)
               t.left(120)
          nestedTriangle(t, side/2)

def fractalTriangle(t, side):
     if side > 10:
          for i in range(3):
               t.forward(side)
               t.left(120)
               fractalTriangle(t, side/2)
```

```
def main():
     nessa = turtle.Turtle()
     setUp(nessa, 100, "violet")
     nestedTriangle(nessa, 160)

     frank = turtle.Turtle()
     setUp(frank, -100, "red")
     fractalTriangle(frank, 160)

if __name__ == "__main__":
     main()
```

IDLE

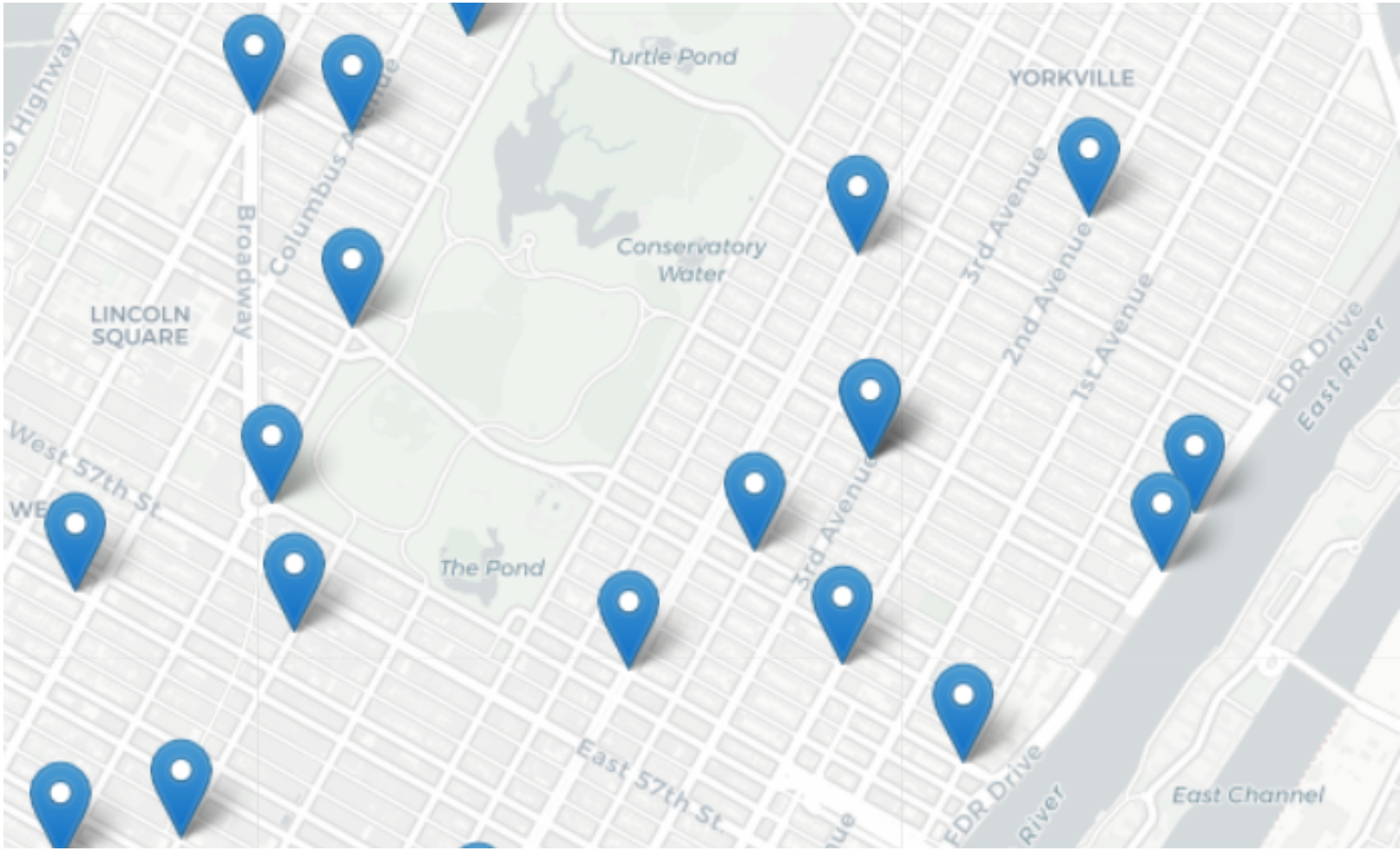
```
#CSci 127 Teaching Staff
#Triangles two ways...
import turtle

def setUp(t, dist, col):
     t.penup()
     t.forward(dist)
     t.pendown()
     t.color(col)

def nestedTriangle(t, side):
     if side > 10:
          for i in range(3):
               t.forward(side)
               t.left(120)
          nestedTriangle(t, side/2)

def fractalTriangle(t, side):
     if side > 10:
          for i in range(3):
               t.forward(side)
               t.left(120)
               fractalTriangle(t, side/2)
```

(Demo with IDLE)

Today's Topics

- More on Functions
- **Recap: Open Data**
- Top Down Design
- Github
- CS Survey: Career Services

OpenData Design Question



Design an algorithm that finds the closest collision.

(Sample NYC OpenData collision data file on back of lecture slip.)

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

How to approach this:

- Create a "To Do" list of what your program has to accomplish.

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

How to approach this:

- Create a "To Do" list of what your program has to accomplish.
- Read through the problem, and break it into "To Do" items.

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

How to approach this:

- Create a "To Do" list of what your program has to accomplish.
- Read through the problem, and break it into "To Do" items.
- Don't worry if you don't know how to do all the items you write down.

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

How to approach this:

- Create a "To Do" list of what your program has to accomplish.
- Read through the problem, and break it into "To Do" items.
- Don't worry if you don't know how to do all the items you write down.
- Example:

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

How to approach this:

- Create a "To Do" list of what your program has to accomplish.
- Read through the problem, and break it into "To Do" items.
- Don't worry if you don't know how to do all the items you write down.
- Example:
  1. Find data set (great place to look: NYC OpenData).

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

How to approach this:

- Create a "To Do" list of what your program has to accomplish.
- Read through the problem, and break it into "To Do" items.
- Don't worry if you don't know how to do all the items you write down.
- Example:
  1. Find data set (great place to look: NYC OpenData).
  2. Ask user for current location.

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

How to approach this:

- Create a "To Do" list of what your program has to accomplish.
- Read through the problem, and break it into "To Do" items.
- Don't worry if you don't know how to do all the items you write down.
- Example:
  1. Find data set (great place to look: NYC OpenData).
  2. Ask user for current location.
  3. Open up the CSV file.

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

How to approach this:

- Create a "To Do" list of what your program has to accomplish.
- Read through the problem, and break it into "To Do" items.
- Don't worry if you don't know how to do all the items you write down.
- Example:
  1. Find data set (great place to look: NYC OpenData).
  2. Ask user for current location.
  3. Open up the CSV file.
  4. Check distance to each to user's location.

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

How to approach this:

- Create a "To Do" list of what your program has to accomplish.
- Read through the problem, and break it into "To Do" items.
- Don't worry if you don't know how to do all the items you write down.
- Example:
  1. Find data set (great place to look: NYC OpenData).
  2. Ask user for current location.
  3. Open up the CSV file.
  4. Check distance to each to user's location.
  5. Print the location with the smallest distance.

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

How to approach this:

- Create a "To Do" list of what your program has to accomplish.
- Read through the problem, and break it into "To Do" items.
- Don't worry if you don't know how to do all the items you write down.
- Example:
  1. Find data set (great place to look: NYC OpenData).
  2. Ask user for current location.
  3. Open up the CSV file.
  4. Check distance to each to user's location.
  5. Print the location with the smallest distance.
- Let's use function names as placeholders for the ones we're unsure...

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

1. Find data set (great place to look: NYC OpenData).

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

1. Find data set (great place to look: NYC OpenData).

```
import pandas as pd
inF = input('Enter CSV file name:')
```

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

1. Find data set (great place to look: NYC OpenData).

```
import pandas as pd
inF = input('Enter CSV file name:')
```

2. Ask user for current location.

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

1. Find data set (great place to look: NYC OpenData).

```
import pandas as pd
inF = input('Enter CSV file name:')
```

2. Ask user for current location.

```
lat = float(input('Enter latitude:'))
lon = float(input('Enter longitude:'))
```

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

1. Find data set (great place to look: NYC OpenData).

```
import pandas as pd
inF = input('Enter CSV file name:')
```

2. Ask user for current location.

```
lat = float(input('Enter latitude:'))
lon = float(input('Enter longitude:'))
```

3. Open up the CSV file.

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

1. Find data set (great place to look: NYC OpenData).

```
import pandas as pd
inF = input('Enter CSV file name:')
```

2. Ask user for current location.

```
lat = float(input('Enter latitude:'))
lon = float(input('Enter longitude:'))
```

3. Open up the CSV file.

```
collisions = pd.read_csv(inF)
```

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

1. Find data set (great place to look: NYC OpenData).

```
import pandas as pd
inF = input('Enter CSV file name:')
```

2. Ask user for current location.

```
lat = float(input('Enter latitude:'))
lon = float(input('Enter longitude:'))
```

3. Open up the CSV file.

```
collisions = pd.read_csv(inF)
```

4. Check distance to each to user's location.

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

1. Find data set (great place to look: NYC OpenData).

```
import pandas as pd
inF = input('Enter CSV file name:')
```

2. Ask user for current location.

```
lat = float(input('Enter latitude:'))
lon = float(input('Enter longitude:'))
```

3. Open up the CSV file.

```
collisions = pd.read_csv(inF)
```

4. Check distance to each to user's location.

```
closestLat, closestLon = findClosest(collisions, lat, lon)
```

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

1. Find data set (great place to look: NYC OpenData).

```
import pandas as pd
inF = input('Enter CSV file name:')
```

2. Ask user for current location.

```
lat = float(input('Enter latitude:'))
lon = float(input('Enter longitude:'))
```

3. Open up the CSV file.

```
collisions = pd.read_csv(inF)
```

4. Check distance to each to user's location.

```
closestLat, closestLon = findClosest(collisions, lat, lon)
```

5. Print the location with the smallest distance.

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

1. Find data set (great place to look: NYC OpenData).

```
import pandas as pd
inF = input('Enter CSV file name:')
```

2. Ask user for current location.

```
lat = float(input('Enter latitude:'))
lon = float(input('Enter longitude:'))
```

3. Open up the CSV file.

```
collisions = pd.read_csv(inF)
```

4. Check distance to each to user's location.

```
closestLat, closestLon = findClosest(collisions, lat, lon)
```

5. Print the location with the smallest distance.

```
print("The closest is at lat:", lat, "and lon:", lon)
```

OpenData Design Question

Design an algorithm that uses NYC OpenData collision data and computes the closest collision to the location the user provides.

1. Find data set (great place to look: NYC OpenData).

```
import pandas as pd
inF = input('Enter CSV file name:')
```

2. Ask user for current location.

```
lat = float(input('Enter latitude:'))
lon = float(input('Enter longitude:'))
```

3. Open up the CSV file.

```
collisions = pd.read_csv(inF)
```

4. Check distance to each to user's location.

```
closestLat, closestLon = findClosest(collisions, lat, lon)
```

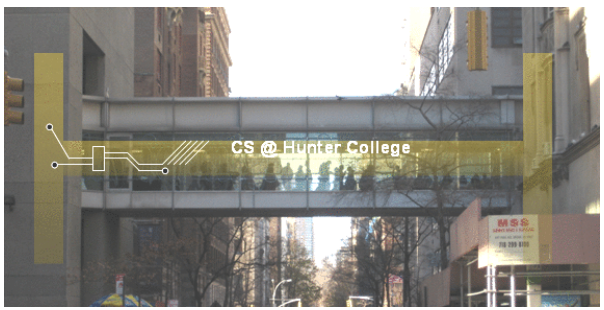
5. Print the location with the smallest distance.

```
print("The closest is at lat:", lat, "and lon:", lon)
```

Today's Topics

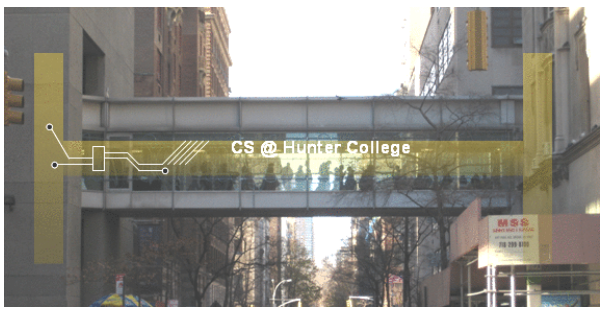
- More on Functions
- Recap: Open Data
- **Top Down Design**
- Github
- CS Survey: Career Services

Top-Down Design



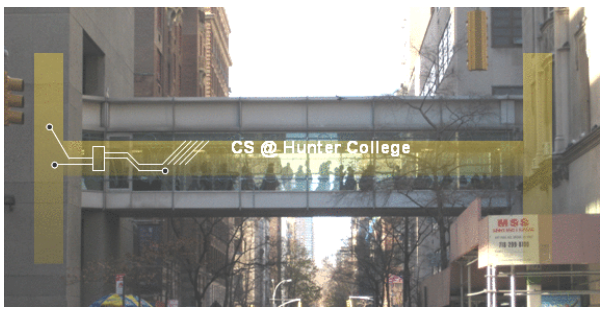
- The last example demonstrates **top-down design**: breaking into subproblems, and implementing each part separately.

Top-Down Design



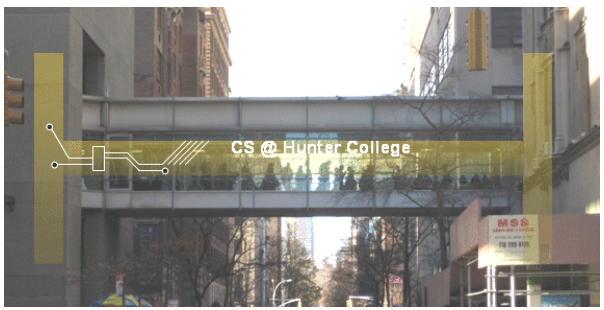
- The last example demonstrates **top-down design**: breaking into subproblems, and implementing each part separately.
  - Break the problem into tasks for a "To Do" list.

Top-Down Design



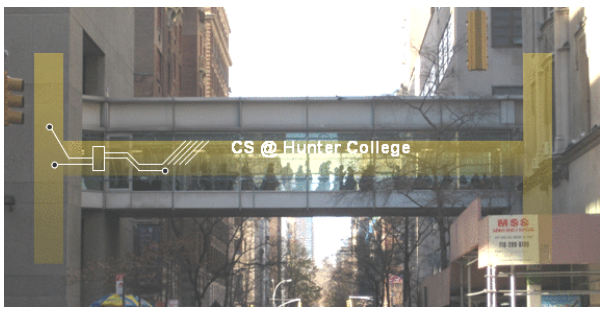
- The last example demonstrates **top-down design**: breaking into subproblems, and implementing each part separately.
  - Break the problem into tasks for a "To Do" list.
  - Translate list into function names & inputs/returns.

Top-Down Design



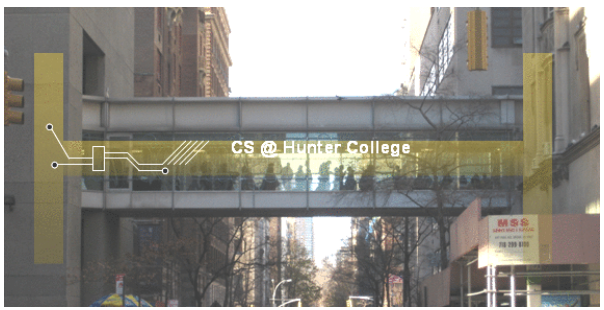
- The last example demonstrates **top-down design**: breaking into subproblems, and implementing each part separately.
  - Break the problem into tasks for a "To Do" list.
  - Translate list into function names & inputs/returns.
  - Implement the functions, one-by-one.

Top-Down Design



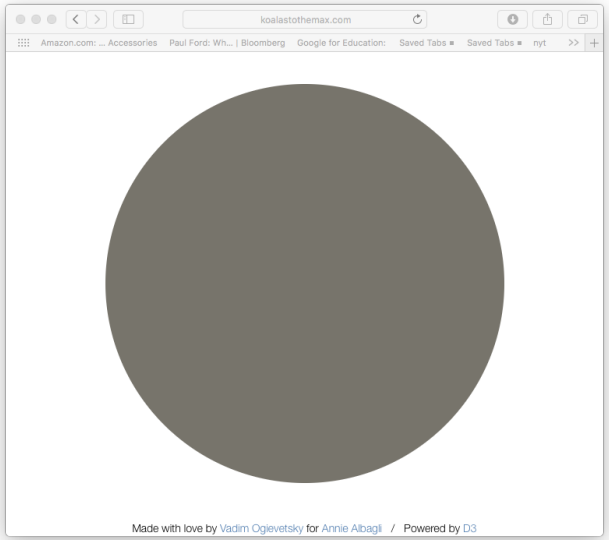
- The last example demonstrates **top-down design**: breaking into subproblems, and implementing each part separately.
  - Break the problem into tasks for a "To Do" list.
  - Translate list into function names & inputs/returns.
  - Implement the functions, one-by-one.
- Excellent approach since you can then test each part separately before adding it to a large program.

Top-Down Design



- The last example demonstrates **top-down design**: breaking into subproblems, and implementing each part separately.
  - Break the problem into tasks for a "To Do" list.
  - Translate list into function names & inputs/returns.
  - Implement the functions, one-by-one.
- Excellent approach since you can then test each part separately before adding it to a large program.
- Very common when working with a team: each has their own functions to implement and maintain.

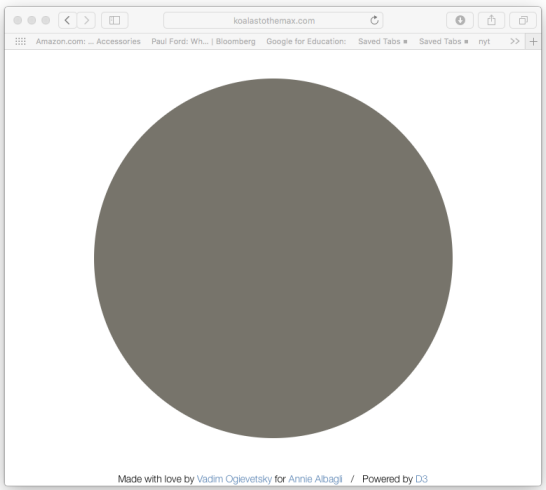
In Pairs or Triples:



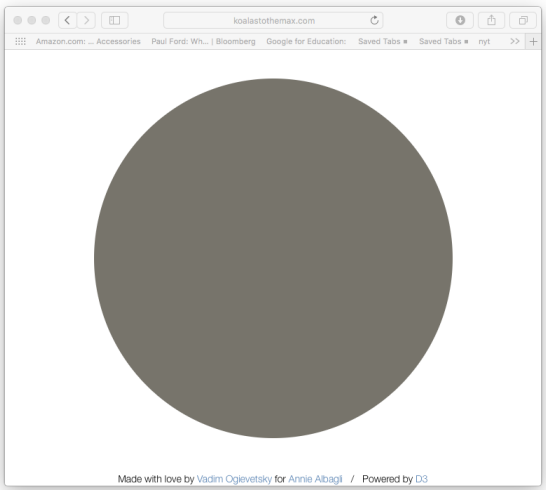
`http://koalastothemax.com`

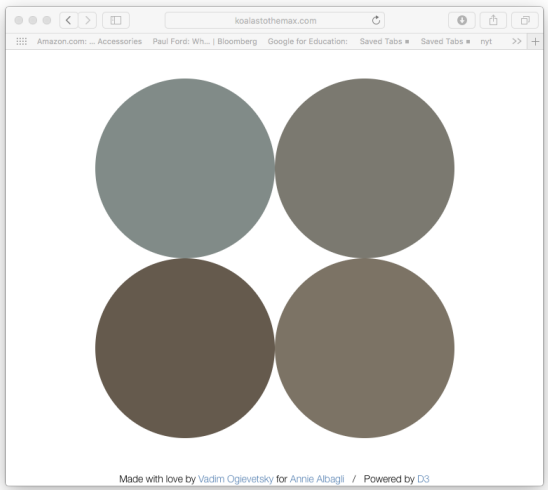
- Top-down design puzzle:
  - What does `koalastomax` do?
  - What does each circle represent?
- Write a high-level design for it.
- Translate into code with function calls.

Demo

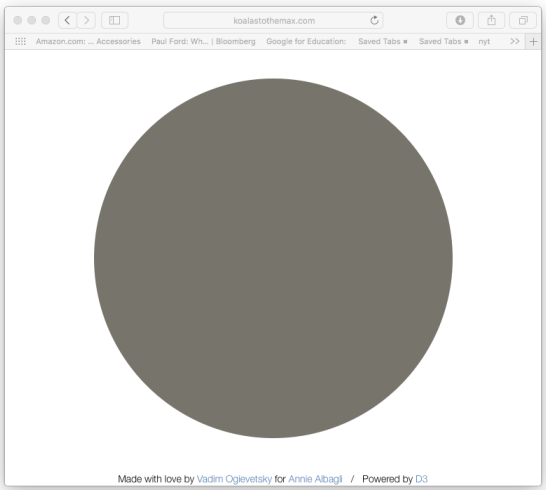

koalastothemax.com
Made with love by Vadim Ogievetsky for Annie Albagli / Powered by D3

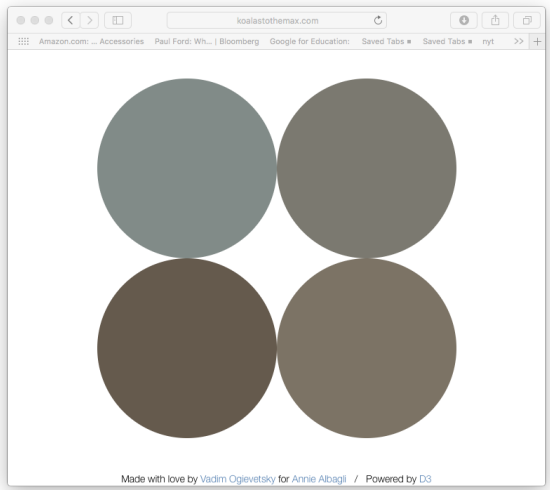
Demo

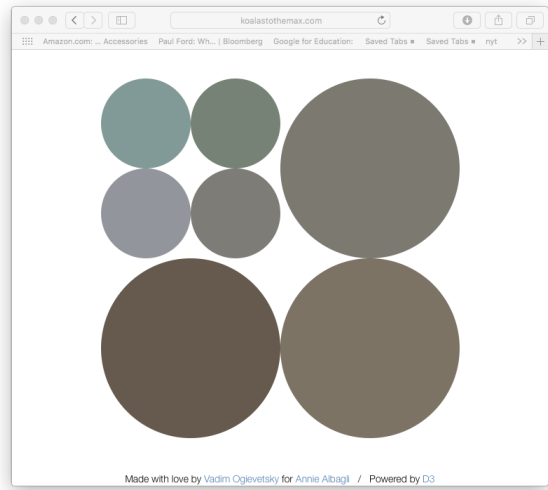

koalastothemax.com
Made with love by Vadim Ogievetsky for Annie Albagli / Powered by D3


koalastothemax.com
Made with love by Vadim Ogievetsky for Annie Albagli / Powered by D3

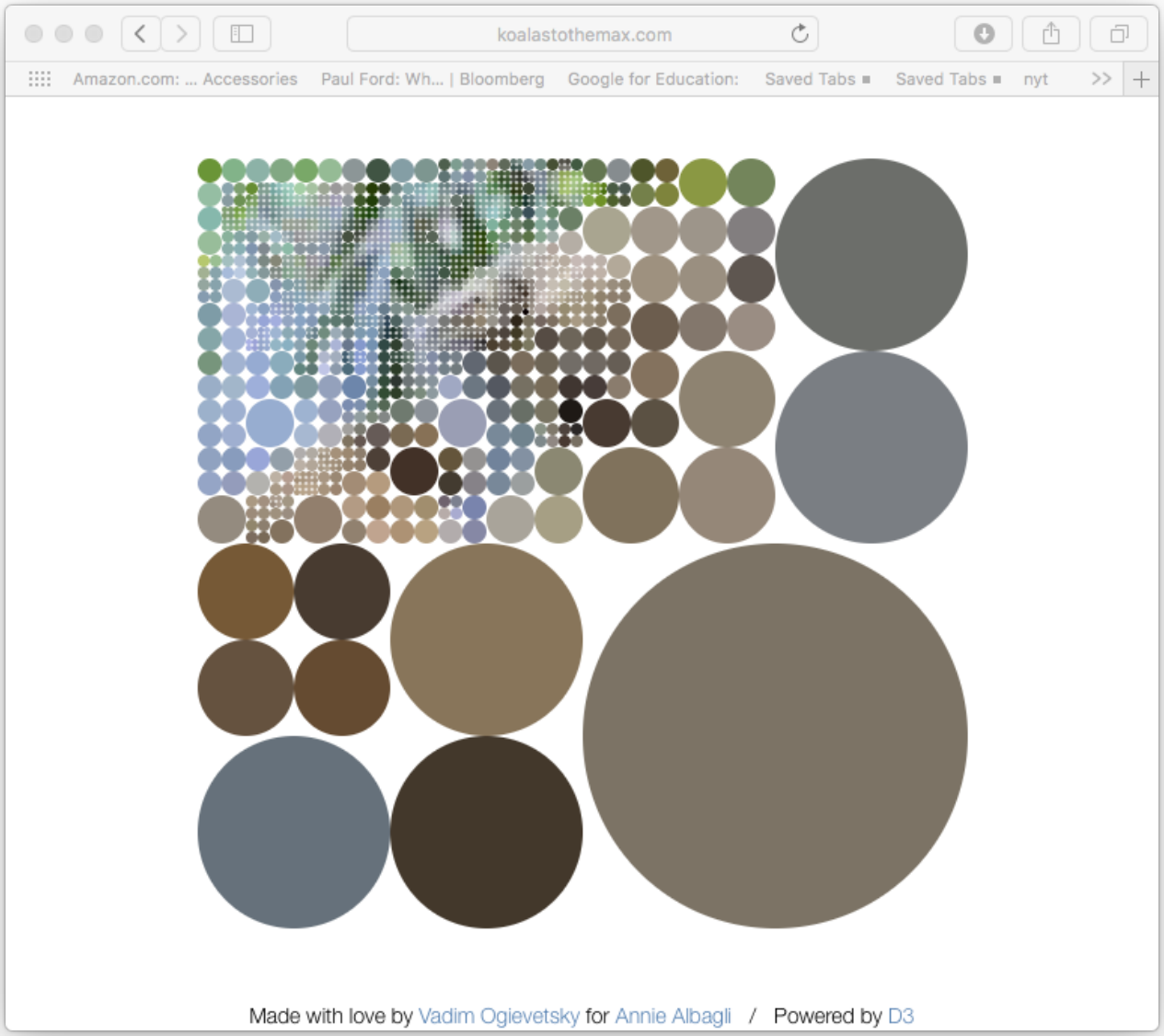
Demo


koalastothemax.com
Made with love by Vadim Ogievetsky for Annie Albagli / Powered by D3

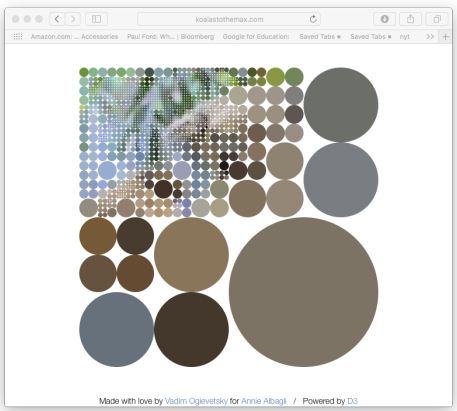

koalastothemax.com
Made with love by Vadim Ogievetsky for Annie Albagli / Powered by D3


koalastothemax.com
Made with love by Vadim Ogievetsky for Annie Albagli / Powered by D3

Demo

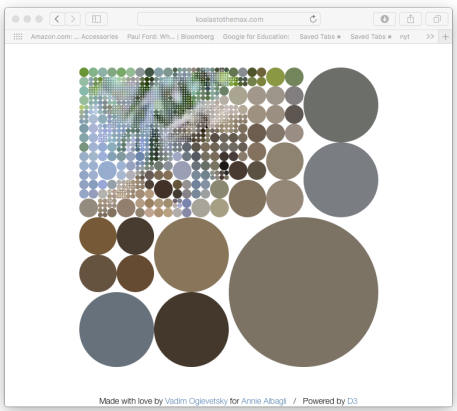

koalastothemax.com
Amazon.com: ... Accessories
Paul Ford: Wh... | Bloomberg
Google for Education:
Saved Tabs
Saved Tabs
nyt
Made with love by Vadim Ogievetsky for Annie Albagli / Powered by D3

Design: Koalas to the Max



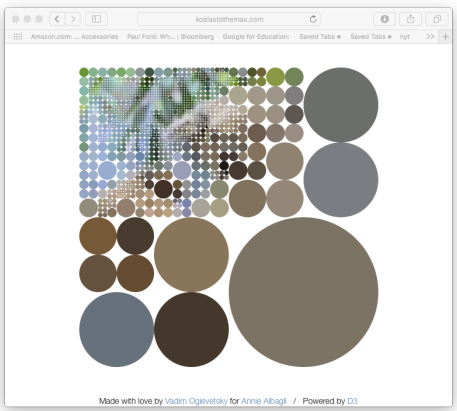
- **Input:** Image & mouse movements

Design: Koalas to the Max



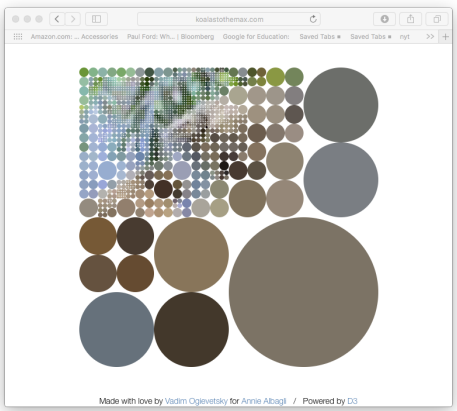
- **Input:** Image & mouse movements
- **Output:** Completed image

Design: Koalas to the Max



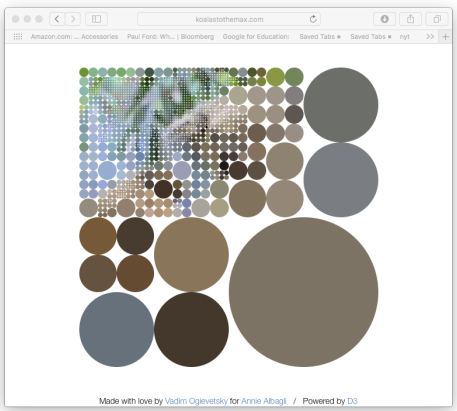
- **Input:** Image & mouse movements
- **Output:** Completed image
- **Design:**

Design: Koalas to the Max



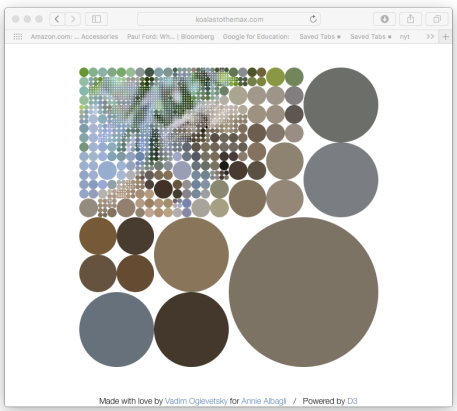
- **Input:** Image & mouse movements
- **Output:** Completed image
- **Design:**
  - Every mouse movement,

Design: Koalas to the Max



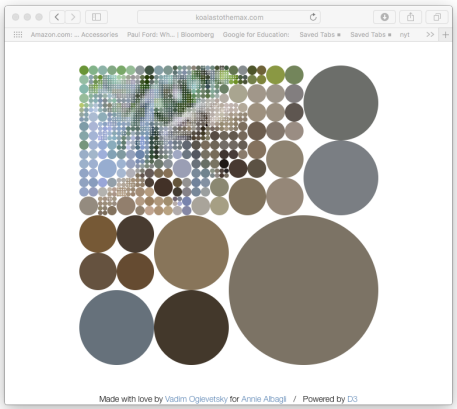
- **Input:** Image & mouse movements
- **Output:** Completed image
- **Design:**
  - Every mouse movement,
  - Divide the region into 4 quarters.

Design: Koalas to the Max



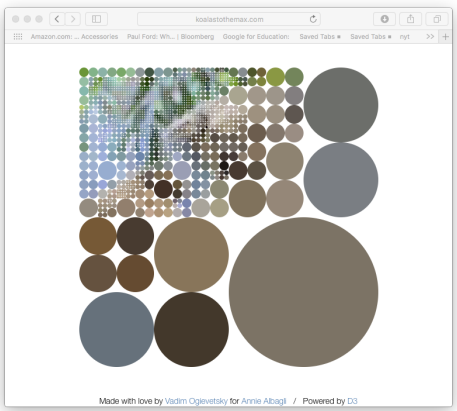
- **Input:** Image & mouse movements
- **Output:** Completed image
- **Design:**
  - Every mouse movement,
  - Divide the region into 4 quarters.
  - Average the color of each quarter.

Design: Koalas to the Max



- **Input:** Image & mouse movements
- **Output:** Completed image
- **Design:**
  - Every mouse movement,
  - Divide the region into 4 quarters.
  - Average the color of each quarter.
  - Set each quarter to its average.

Design: Koalas to the Max



- **Input:** Image & mouse movements
- **Output:** Completed image
- **Design:**
  - Every mouse movement,
  - Divide the region into 4 quarters.
  - Average the color of each quarter.
  - Set each quarter to its average.

(Demo program from github.)

Today's Topics

- More on Functions
- Recap: Open Data
- Top Down Design
- **Github**
- CS Survey: Career Services

Github

- Like Google docs for code...

Octocat

Github

Octocat

- Like Google docs for code...
- Used to share code, documents, etc.

Github

Octocat

- Like Google docs for code...
- Used to share code, documents, etc.
- More formally: `git` is a version control protocol for tracking changes and versions of documents.

Github

Octocat

- Like Google docs for code...
- Used to share code, documents, etc.
- More formally: `git` is a version control protocol for tracking changes and versions of documents.
- Github provides hosting for repositories (‘**repos**’) of code.

Github

Octocat

- Like Google docs for code...
- Used to share code, documents, etc.
- More formally: `git` is a version control protocol for tracking changes and versions of documents.
- Github provides hosting for repositories (‘**repos**’) of code.
- Also convenient place to host websites (i.e. `stjohn.github.io`).

Github

Octocat

- Like Google docs for code...
- Used to share code, documents, etc.
- More formally: `git` is a version control protocol for tracking changes and versions of documents.
- Github provides hosting for repositories (‘**repos**’) of code.
- Also convenient place to host websites (i.e. `stjohn.github.io`).
- In lab, we will set up github accounts and copy (‘**clone**’) documents from the class repo. (More in future courses.)

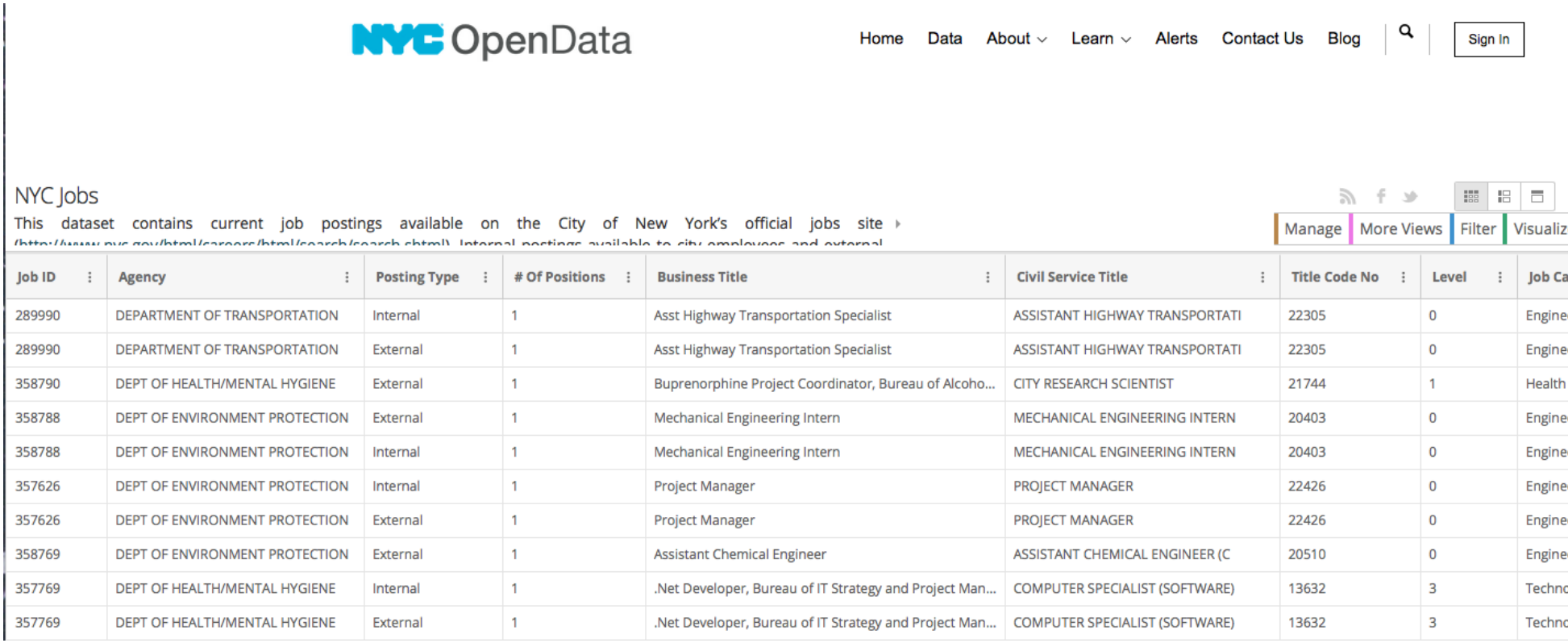
Today's Topics

- More on Functions
- Recap: Open Data
- Top Down Design
- Github
- **CS Survey: Career Services**

CS Survey Talk

Keith Okrosy
Career Development Services

Design Challenge

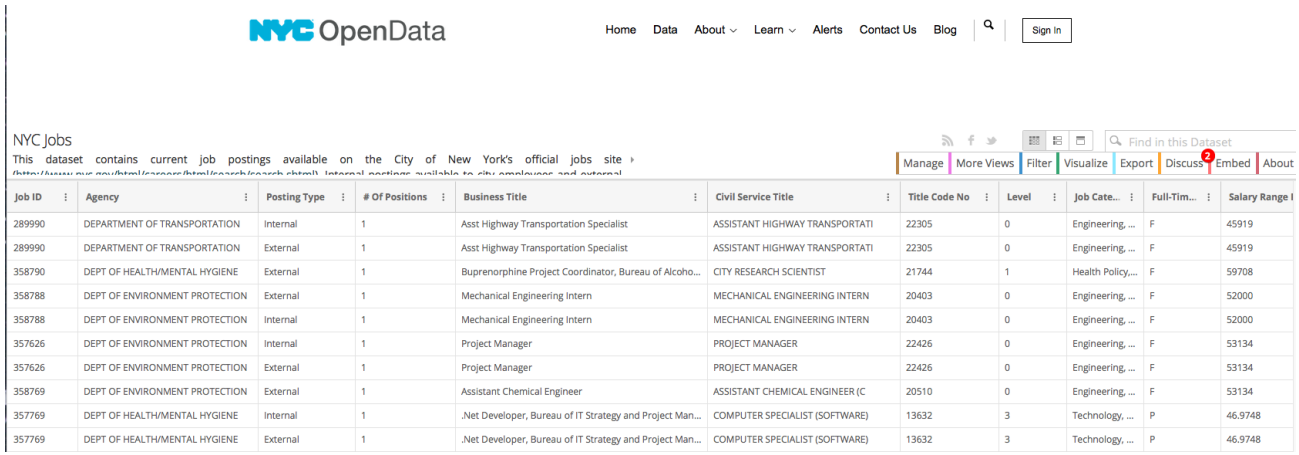


Job ID	Agency	Posting Type	# Of Positions	Business Title	Civil Service Title	Title Code No	Level	Job Ca
289990	DEPARTMENT OF TRANSPORTATION	Internal	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engine
289990	DEPARTMENT OF TRANSPORTATION	External	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engine
358790	DEPT OF HEALTH/MENTAL HYGIENE	External	1	Buprenorphine Project Coordinator, Bureau of Alcoho...	CITY RESEARCH SCIENTIST	21744	1	Health
358788	DEPT OF ENVIRONMENT PROTECTION	External	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engine
358788	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engine
357626	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Project Manager	PROJECT MANAGER	22426	0	Engine
357626	DEPT OF ENVIRONMENT PROTECTION	External	1	Project Manager	PROJECT MANAGER	22426	0	Engine
358769	DEPT OF ENVIRONMENT PROTECTION	External	1	Assistant Chemical Engineer	ASSISTANT CHEMICAL ENGINEER (C	20510	0	Engine
357769	DEPT OF HEALTH/MENTAL HYGIENE	Internal	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Techno
357769	DEPT OF HEALTH/MENTAL HYGIENE	External	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Techno

`(data.cityofnewyork.us/City-Government/NYC-Jobs/kpav-sd4t)`

Find all current city job postings for internship positions.

Design Challenge

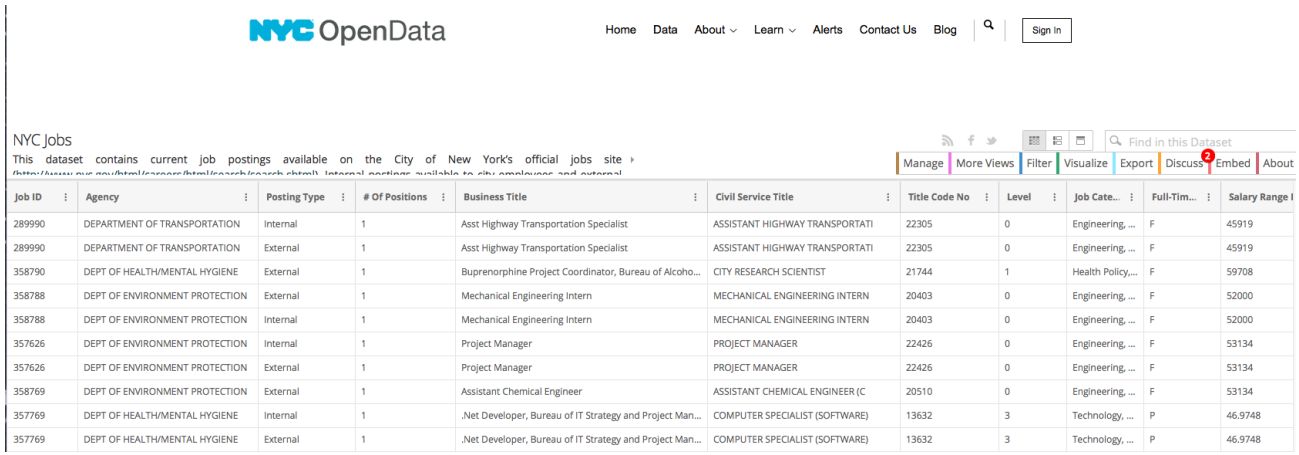


Job ID	Agency	Posting Type	# Of Positions	Business Title	Civil Service Title	Title Code No	Level	Job Cate...	Full-Tim...	Salary Range
289990	DEPARTMENT OF TRANSPORTATION	Internal	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engineering, ...	F	45919
289990	DEPARTMENT OF TRANSPORTATION	External	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engineering, ...	F	45919
358790	DEPT OF HEALTH/MENTAL HYGIENE	External	1	Buprenorphine Project Coordinator, Bureau of Alcoho...	CITY RESEARCH SCIENTIST	21744	1	Health Policy,...	F	59708
358788	DEPT OF ENVIRONMENT PROTECTION	External	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engineering, ...	F	52000
358788	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engineering, ...	F	52000
357626	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Project Manager	PROJECT MANAGER	22426	0	Engineering, ...	F	53134
357626	DEPT OF ENVIRONMENT PROTECTION	External	1	Project Manager	PROJECT MANAGER	22426	0	Engineering, ...	F	53134
358769	DEPT OF ENVIRONMENT PROTECTION	External	1	Assistant Chemical Engineer	ASSISTANT CHEMICAL ENGINEER (C	20510	0	Engineering, ...	F	53134
357769	DEPT OF HEALTH/MENTAL HYGIENE	Internal	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Technology, ...	P	46.9748
357769	DEPT OF HEALTH/MENTAL HYGIENE	External	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Technology, ...	P	46.9748

`(data.cityofnewyork.us/City-Government/NYC-Jobs/kpav-sd4t)`

- **Input:** CSV file from NYC OpenData.

Design Challenge

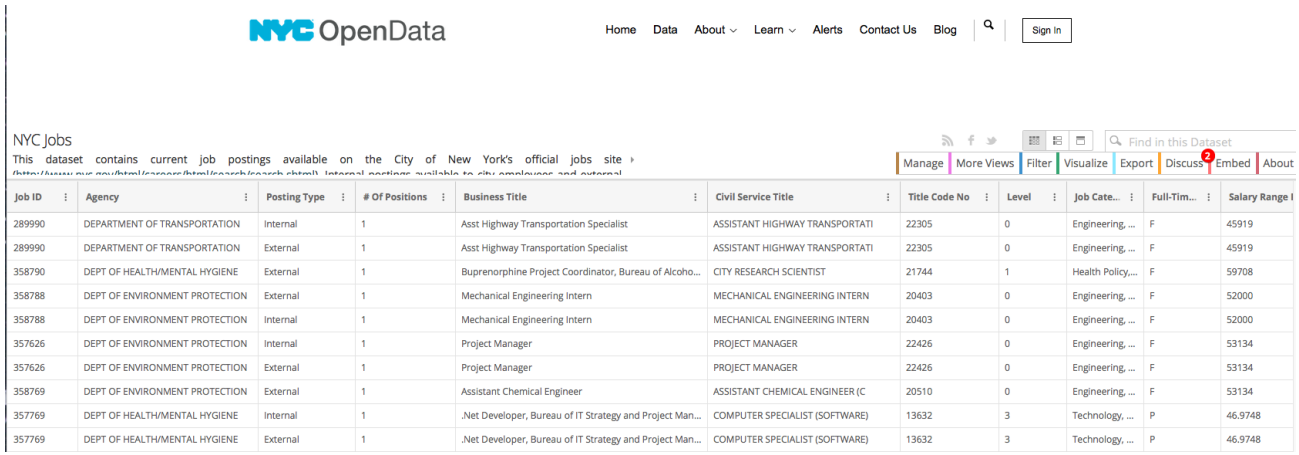


Job ID	Agency	Posting Type	# Of Positions	Business Title	Civil Service Title	Title Code No	Level	Job Cate...	Full-Tim...	Salary Range
289990	DEPARTMENT OF TRANSPORTATION	Internal	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engineering, ...	F	45919
289990	DEPARTMENT OF TRANSPORTATION	External	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engineering, ...	F	45919
358790	DEPT OF HEALTH/MENTAL HYGIENE	External	1	Buprenorphine Project Coordinator, Bureau of Alcoho...	CITY RESEARCH SCIENTIST	21744	1	Health Policy,...	F	59708
358788	DEPT OF ENVIRONMENT PROTECTION	External	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engineering, ...	F	52000
358788	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engineering, ...	F	52000
357626	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Project Manager	PROJECT MANAGER	22426	0	Engineering, ...	F	53134
357626	DEPT OF ENVIRONMENT PROTECTION	External	1	Project Manager	PROJECT MANAGER	22426	0	Engineering, ...	F	53134
358769	DEPT OF ENVIRONMENT PROTECTION	External	1	Assistant Chemical Engineer	ASSISTANT CHEMICAL ENGINEER (C	20510	0	Engineering, ...	F	53134
357769	DEPT OF HEALTH/MENTAL HYGIENE	Internal	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Technology, ...	P	46.9748
357769	DEPT OF HEALTH/MENTAL HYGIENE	External	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Technology, ...	P	46.9748

`(data.cityofnewyork.us/City-Government/NYC-Jobs/kpav-sd4t)`

- **Input:** CSV file from NYC OpenData.
- **Output:** A list of internships offered by the city.

Design Challenge

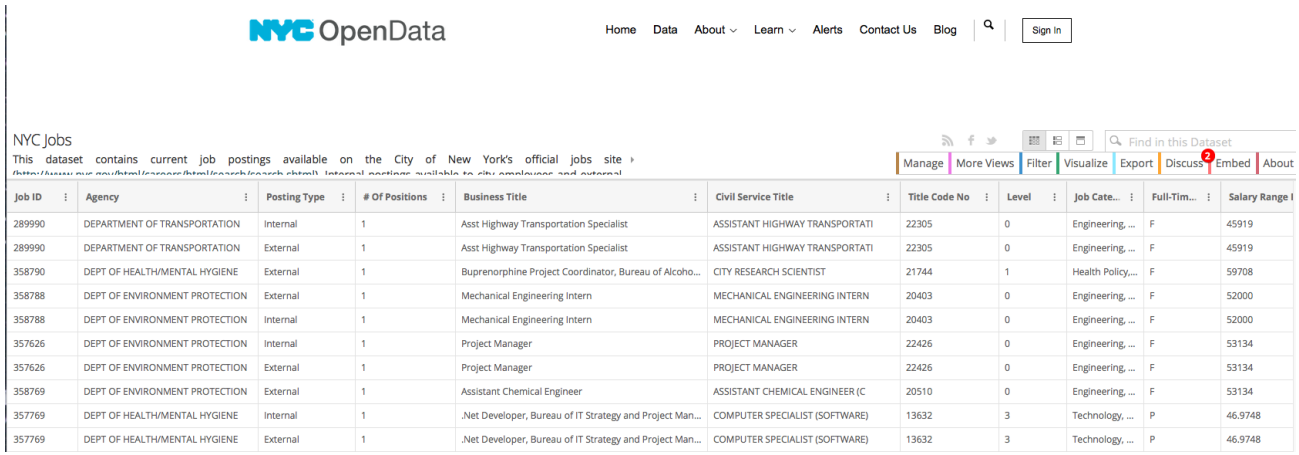


Job ID	Agency	Posting Type	# Of Positions	Business Title	Civil Service Title	Title Code No	Level	Job Cate...	Full-Tim...	Salary Range
289990	DEPARTMENT OF TRANSPORTATION	Internal	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engineering, ...	F	45919
289990	DEPARTMENT OF TRANSPORTATION	External	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engineering, ...	F	45919
358790	DEPT OF HEALTH/MENTAL HYGIENE	External	1	Buprenorphine Project Coordinator, Bureau of Alcoho...	CITY RESEARCH SCIENTIST	21744	1	Health Policy,...	F	59708
358788	DEPT OF ENVIRONMENT PROTECTION	External	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engineering, ...	F	52000
358788	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engineering, ...	F	52000
357626	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Project Manager	PROJECT MANAGER	22426	0	Engineering, ...	F	53134
357626	DEPT OF ENVIRONMENT PROTECTION	External	1	Project Manager	PROJECT MANAGER	22426	0	Engineering, ...	F	53134
358769	DEPT OF ENVIRONMENT PROTECTION	External	1	Assistant Chemical Engineer	ASSISTANT CHEMICAL ENGINEER (C	20510	0	Engineering, ...	F	53134
357769	DEPT OF HEALTH/MENTAL HYGIENE	Internal	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Technology, ...	P	46.9748
357769	DEPT OF HEALTH/MENTAL HYGIENE	External	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Technology, ...	P	46.9748

`(data.cityofnewyork.us/City-Government/NYC-Jobs/kpav-sd4t)`

- **Input:** CSV file from NYC OpenData.
- **Output:** A list of internships offered by the city.
- **Process:**

Design Challenge

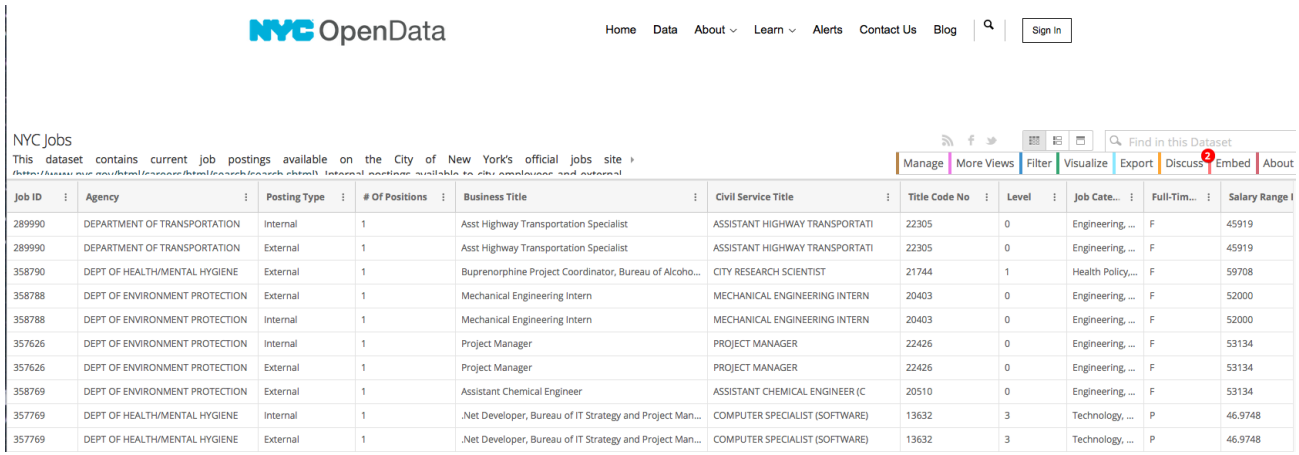


Job ID	Agency	Posting Type	# Of Positions	Business Title	Civil Service Title	Title Code No	Level	Job Cate...	Full-Tim...	Salary Range
289990	DEPARTMENT OF TRANSPORTATION	Internal	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engineering, ...	F	45919
289990	DEPARTMENT OF TRANSPORTATION	External	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engineering, ...	F	45919
358790	DEPT OF HEALTH/MENTAL HYGIENE	External	1	Buprenorphine Project Coordinator, Bureau of Alcoho...	CITY RESEARCH SCIENTIST	21744	1	Health Policy,...	F	59708
358788	DEPT OF ENVIRONMENT PROTECTION	External	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engineering, ...	F	52000
358788	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engineering, ...	F	52000
357626	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Project Manager	PROJECT MANAGER	22426	0	Engineering, ...	F	53134
357626	DEPT OF ENVIRONMENT PROTECTION	External	1	Project Manager	PROJECT MANAGER	22426	0	Engineering, ...	F	53134
358769	DEPT OF ENVIRONMENT PROTECTION	External	1	Assistant Chemical Engineer	ASSISTANT CHEMICAL ENGINEER (C	20510	0	Engineering, ...	F	53134
357769	DEPT OF HEALTH/MENTAL HYGIENE	Internal	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Technology, ...	P	46.9748
357769	DEPT OF HEALTH/MENTAL HYGIENE	External	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Technology, ...	P	46.9748

`(data.cityofnewyork.us/City-Government/NYC-Jobs/kpav-sd4t)`

- **Input:** CSV file from NYC OpenData.
- **Output:** A list of internships offered by the city.
- **Process:**
  1. Open the file.

Design Challenge

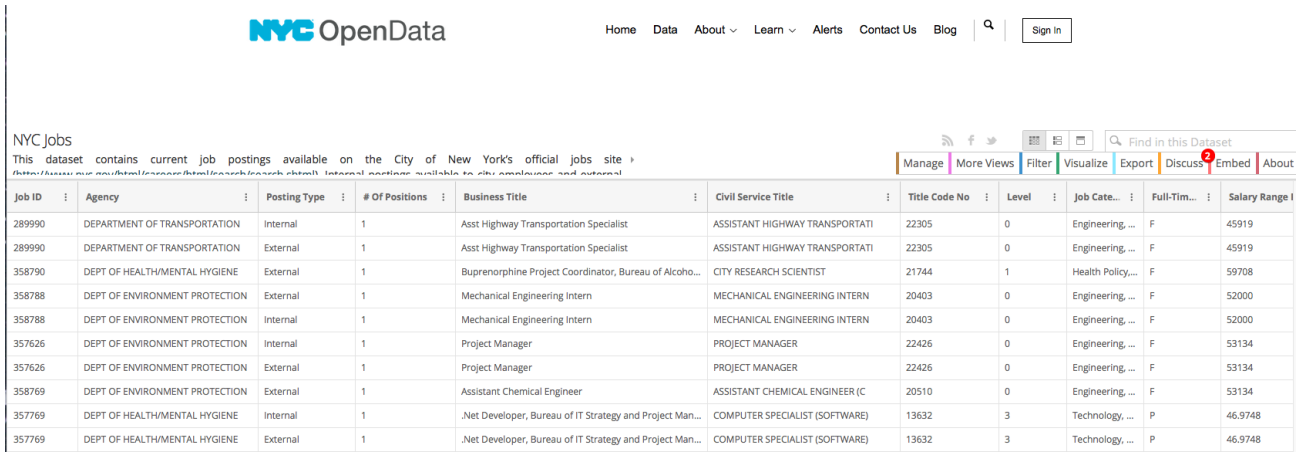


Job ID	Agency	Posting Type	# Of Positions	Business Title	Civil Service Title	Title Code No	Level	Job Cate...	Full-Tim...	Salary Range
289990	DEPARTMENT OF TRANSPORTATION	Internal	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engineering, ...	F	45919
289990	DEPARTMENT OF TRANSPORTATION	External	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engineering, ...	F	45919
358790	DEPT OF HEALTH/MENTAL HYGIENE	External	1	Buprenorphine Project Coordinator, Bureau of Alcoho...	CITY RESEARCH SCIENTIST	21744	1	Health Policy,...	F	59708
358788	DEPT OF ENVIRONMENT PROTECTION	External	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engineering, ...	F	52000
358788	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engineering, ...	F	52000
357626	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Project Manager	PROJECT MANAGER	22426	0	Engineering, ...	F	53134
357626	DEPT OF ENVIRONMENT PROTECTION	External	1	Project Manager	PROJECT MANAGER	22426	0	Engineering, ...	F	53134
358769	DEPT OF ENVIRONMENT PROTECTION	External	1	Assistant Chemical Engineer	ASSISTANT CHEMICAL ENGINEER (C	20510	0	Engineering, ...	F	53134
357769	DEPT OF HEALTH/MENTAL HYGIENE	Internal	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Technology, ...	P	46.9748
357769	DEPT OF HEALTH/MENTAL HYGIENE	External	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Technology, ...	P	46.9748

`(data.cityofnewyork.us/City-Government/NYC-Jobs/kpav-sd4t)`

- **Input:** CSV file from NYC OpenData.
- **Output:** A list of internships offered by the city.
- **Process:**
  1. Open the file.
  2. Select the rows that have "intern" in the business title.

Design Challenge



Job ID	Agency	Posting Type	# Of Positions	Business Title	Civil Service Title	Title Code No	Level	Job Cate...	Full-Tim...	Salary Range
289990	DEPARTMENT OF TRANSPORTATION	Internal	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engineering, ...	F	45919
289990	DEPARTMENT OF TRANSPORTATION	External	1	Asst Highway Transportation Specialist	ASSISTANT HIGHWAY TRANSPORTATI	22305	0	Engineering, ...	F	45919
358790	DEPT OF HEALTH/MENTAL HYGIENE	External	1	Buprenorphine Project Coordinator, Bureau of Alcoho...	CITY RESEARCH SCIENTIST	21744	1	Health Policy,...	F	59708
358788	DEPT OF ENVIRONMENT PROTECTION	External	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engineering, ...	F	52000
358788	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Mechanical Engineering Intern	MECHANICAL ENGINEERING INTERN	20403	0	Engineering, ...	F	52000
357626	DEPT OF ENVIRONMENT PROTECTION	Internal	1	Project Manager	PROJECT MANAGER	22426	0	Engineering, ...	F	53134
357626	DEPT OF ENVIRONMENT PROTECTION	External	1	Project Manager	PROJECT MANAGER	22426	0	Engineering, ...	F	53134
358769	DEPT OF ENVIRONMENT PROTECTION	External	1	Assistant Chemical Engineer	ASSISTANT CHEMICAL ENGINEER (C	20510	0	Engineering, ...	F	53134
357769	DEPT OF HEALTH/MENTAL HYGIENE	Internal	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Technology, ...	P	46.9748
357769	DEPT OF HEALTH/MENTAL HYGIENE	External	1	.Net Developer, Bureau of IT Strategy and Project Man...	COMPUTER SPECIALIST (SOFTWARE)	13632	3	Technology, ...	P	46.9748

`(data.cityofnewyork.us/City-Government/NYC-Jobs/kpav-sd4t)`

- **Input:** CSV file from NYC OpenData.
- **Output:** A list of internships offered by the city.
- **Process:**
  1. Open the file.
  2. Select the rows that have "intern" in the business title.
  3. Print out those rows.

Recap

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

```
#Name:  your name here
#Date:  October 2017
#This program, uses functions,
#     says hello to the world!

def main():
     print("Hello, World!")

if __name__ == "__main__":
     main()
```

Recap

- On lecture slip, write down a topic you wish we had spent more time (and why).
- Functions are a way to break code into pieces, that can be easily reused.

```
#Name:  your name here
#Date:  October 2017
#This program, uses functions,
#     says hello to the world!

def main():
     print("Hello, World!")

if __name__ == "__main__":
     main()
```

Recap

- On lecture slip, write down a topic you wish we had spent more time (and why).
- Functions are a way to break code into pieces, that can be easily reused.
- Functions can have **input parameters** that bring information into the function,

```
#Name:  your name here
#Date:  October 2017
#This program, uses functions,
#     says hello to the world!

def main():
     print("Hello, World!")

if __name__ == "__main__":
     main()
```

Recap

```
#Name:  your name here
#Date:  October 2017
#This program, uses functions,
#     says hello to the world!

def main():
     print("Hello, World!")

if __name__ == "__main__":
     main()
```

- On lecture slip, write down a topic you wish we had spent more time (and why).
- Functions are a way to break code into pieces, that can be easily reused.
- Functions can have **input parameters** that bring information into the function,
- And **return values** that send information back.

Recap

```
#Name:  your name here
#Date:  October 2017
#This program, uses functions,
#     says hello to the world!

def main():
     print("Hello, World!")

if __name__ == "__main__":
     main()
```

- On lecture slip, write down a topic you wish we had spent more time (and why).
- Functions are a way to break code into pieces, that can be easily reused.
- Functions can have **input parameters** that bring information into the function,
- And **return values** that send information back.
- Top-down design: breaking into subproblems, and implementing each part separately.

Recap

```
#Name:  your name here
#Date:  October 2017
#This program, uses functions,
#     says hello to the world!

def main():
     print("Hello, World!")

if __name__ == "__main__":
     main()
```

- On lecture slip, write down a topic you wish we had spent more time (and why).
- Functions are a way to break code into pieces, that can be easily reused.
- Functions can have **input parameters** that bring information into the function,
- And **return values** that send information back.
- Top-down design: breaking into subproblems, and implementing each part separately.
- Excellent approach: can then test each part separately before adding it to a large program.

Recap

```
#Name:  your name here
#Date:  October 2017
#This program, uses functions,
#     says hello to the world!

def main():
    print("Hello, World!")

if __name__ == "__main__":
    main()
```

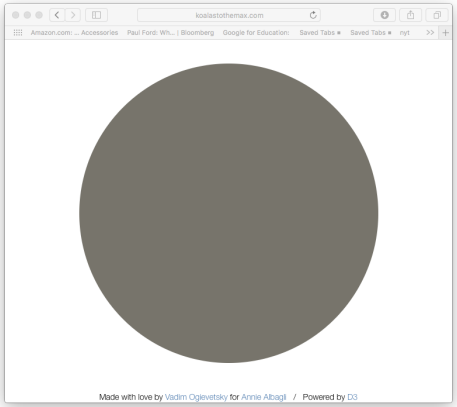
- On lecture slip, write down a topic you wish we had spent more time (and why).
- Functions are a way to break code into pieces, that can be easily reused.
- Functions can have **input parameters** that bring information into the function,
- And **return values** that send information back.
- Top-down design: breaking into subproblems, and implementing each part separately.
- Excellent approach: can then test each part separately before adding it to a large program.
- Github provides a platform for sharing work that allows collaboration (and version control).

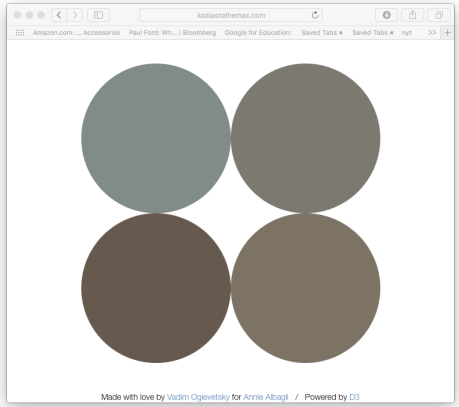
Recap

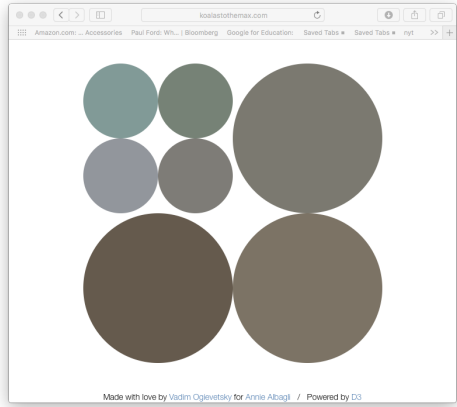
```
#Name:  your name here
#Date:  October 2017
#This program, uses functions,
#     says hello to the world!

def main():
     print("Hello, World!")

if __name__ == "__main__":
     main()
```

- On lecture slip, write down a topic you wish we had spent more time (and why).
- Functions are a way to break code into pieces, that can be easily reused.
- Functions can have **input parameters** that bring information into the function,
- And **return values** that send information back.
- Top-down design: breaking into subproblems, and implementing each part separately.
- Excellent approach: can then test each part separately before adding it to a large program.
- Github provides a platform for sharing work that allows collaboration (and version control).
- Pass your lecture slips to the aisles for the UTAs to collect.

Practice Quiz & Final Questions

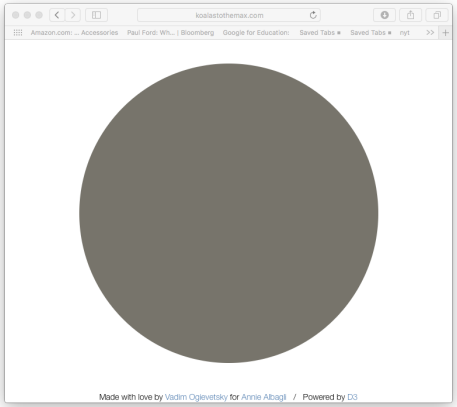


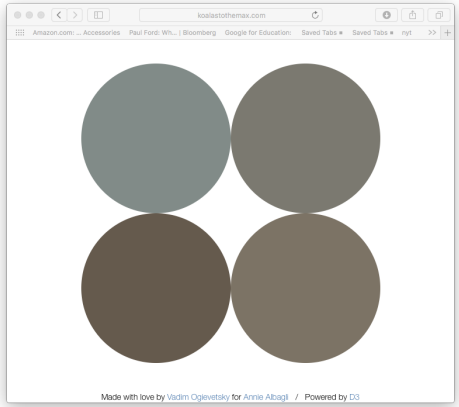


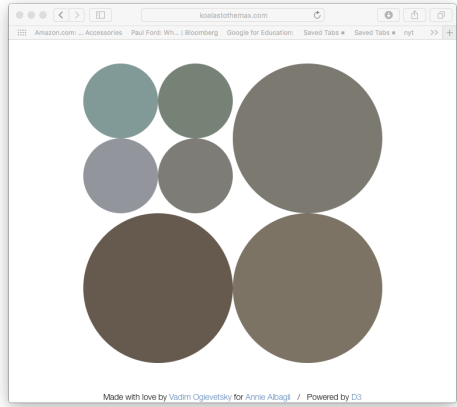


- Since you must pass the final exam to pass the course, we end every lecture with final exam review.

Practice Quiz & Final Questions

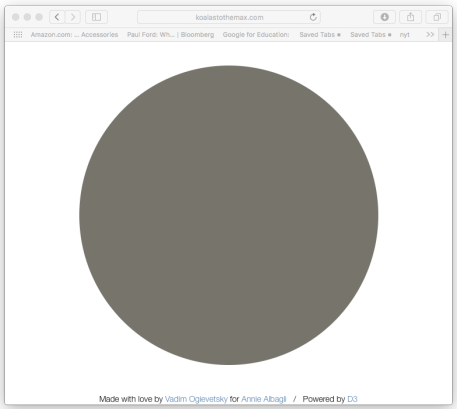


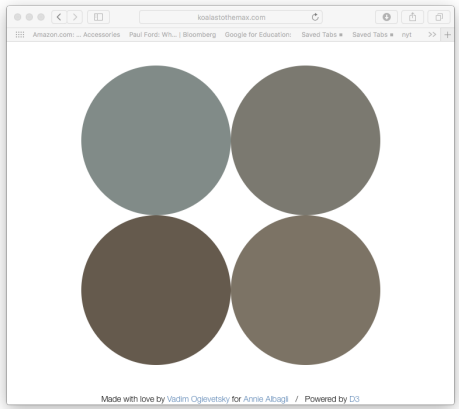


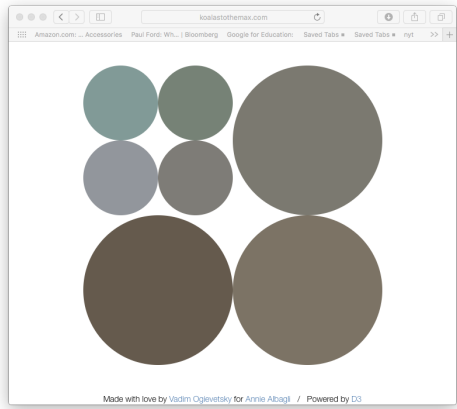


- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).

Practice Quiz & Final Questions

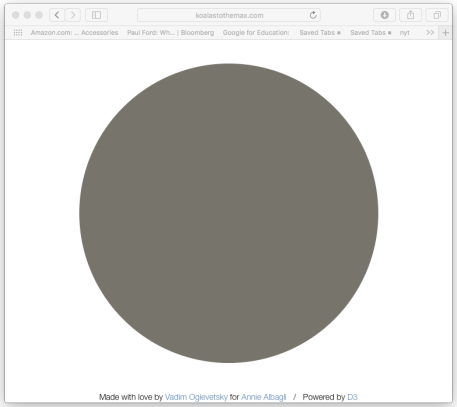


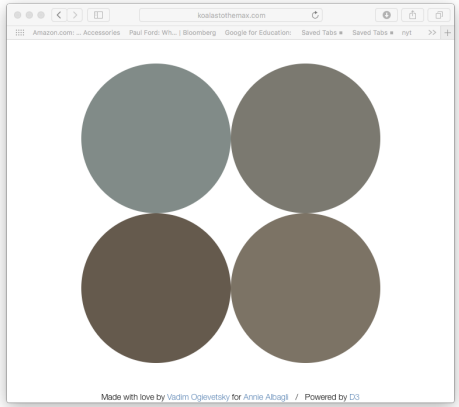


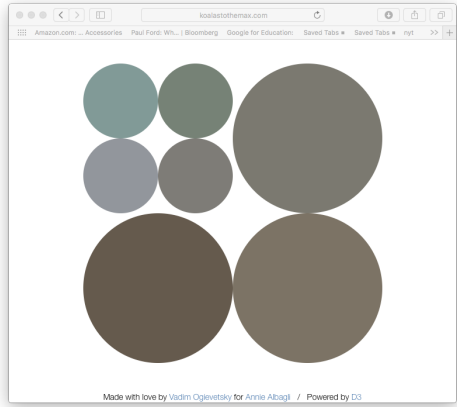


- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:

Practice Quiz & Final Questions

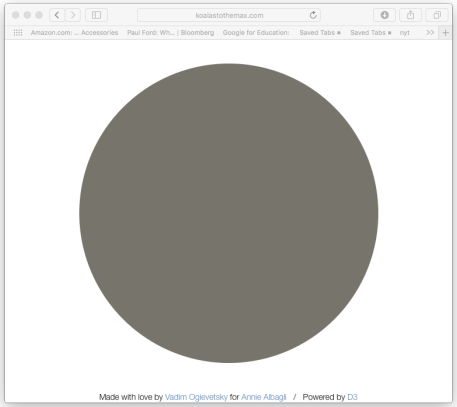


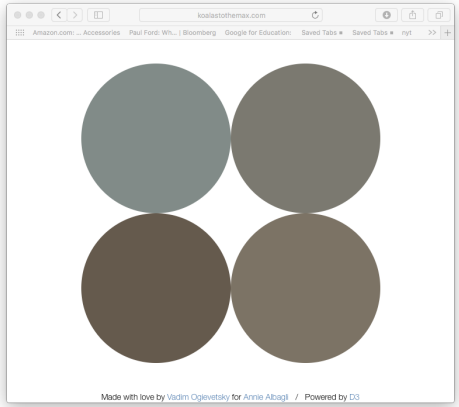


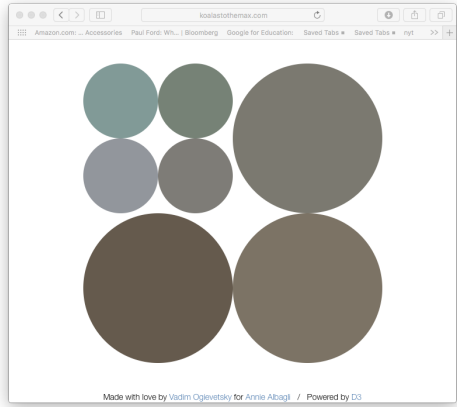


- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;

Practice Quiz & Final Questions

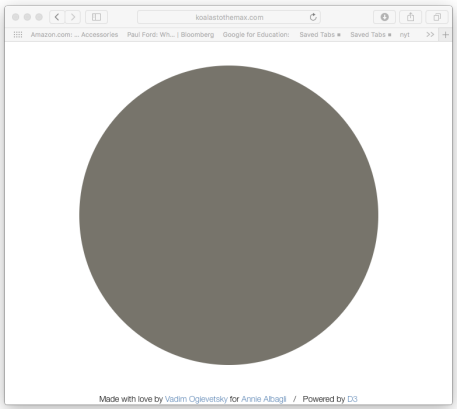


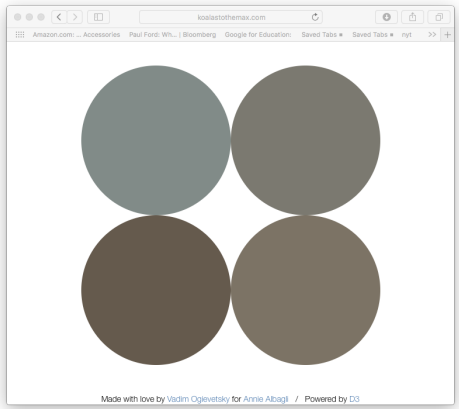


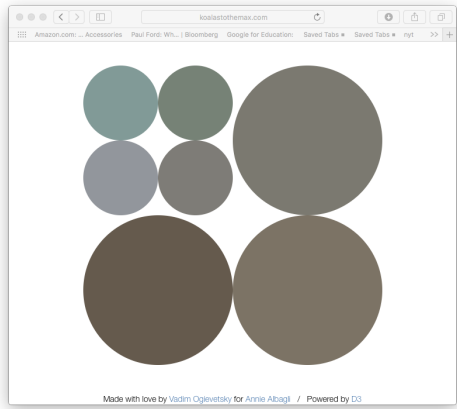


- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and

Practice Quiz & Final Questions

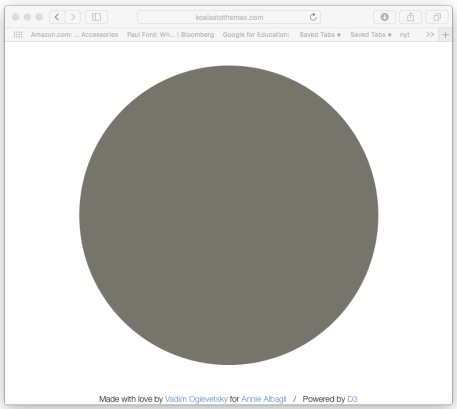


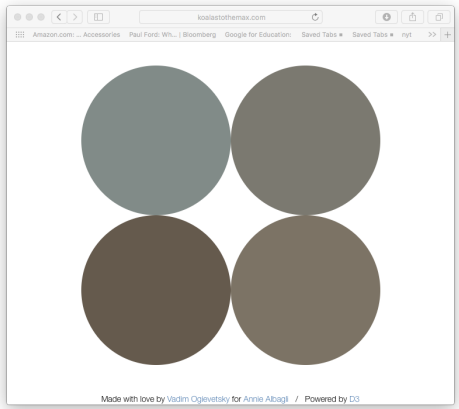


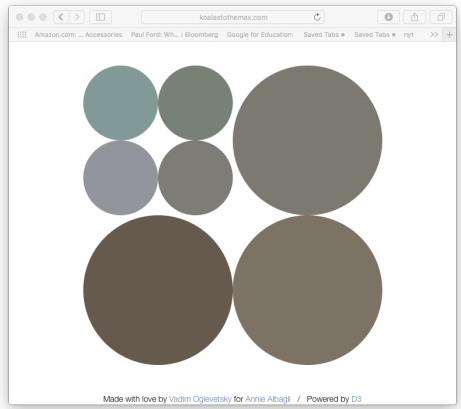


- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - repeat.

Practice Quiz & Final Questions

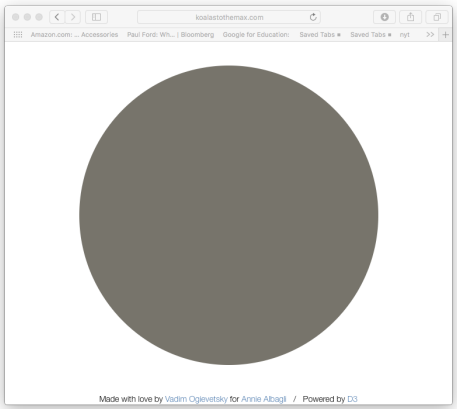


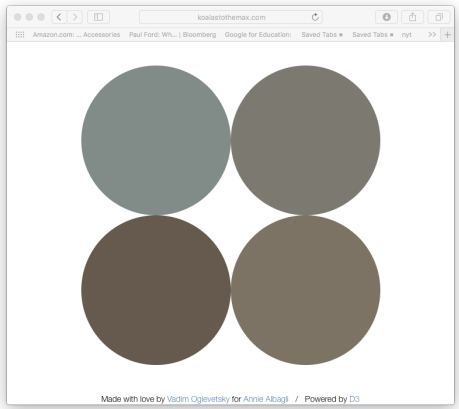


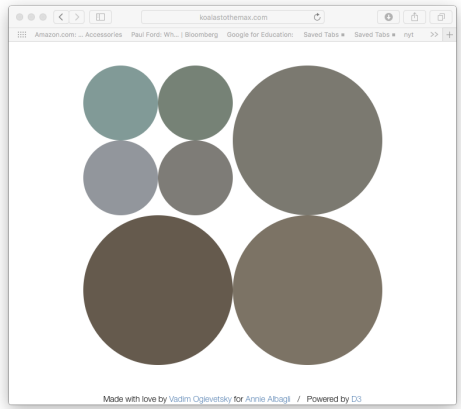


- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - repeat.
- Past exams are on the webpage (under Final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

Practice Quiz & Final Questions

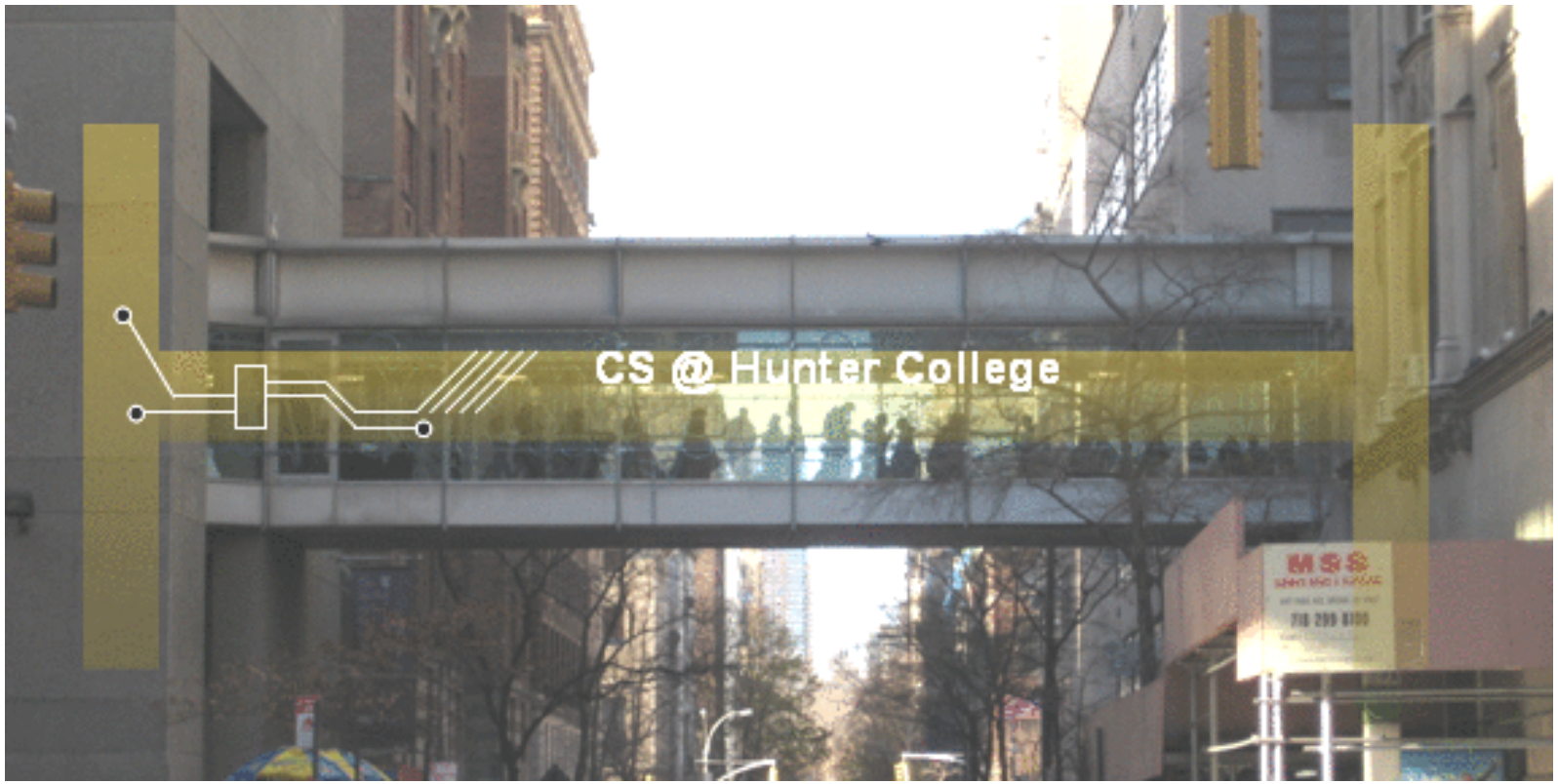






- Since you must pass the final exam to pass the course, we end every lecture with final exam review.
- Pull out something to write on (not to be turned in).
- Lightning rounds:
  - write as much you can for 60 seconds;
  - followed by answer; and
  - repeat.
- Past exams are on the webpage (under Final Exam Information).
- Theme: Functions! Starting with S18, V1, #4 and #7.

Writing Boards



- Return writing boards as you leave...