

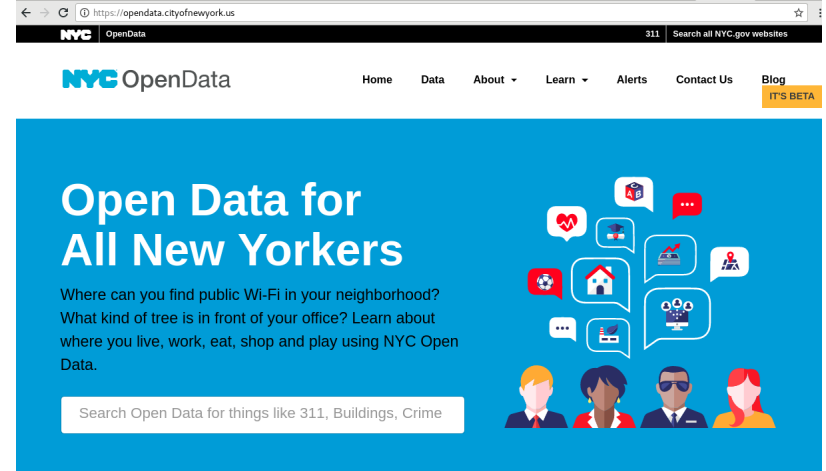
# **YOU** can Create Tools to Influence Policy

using  python<sup>™</sup> and  **CrossCompute**  
CREATE. SHARE. GROW.

**YOU** can influence  
policy by creating  
tools that let  
policymakers  
explore the impact  
of their decisions.



# You can CREATE TOOLS to influence policy using open data.



How You Can Get Involved





# CrossCompute

CREATE. SHARE. GROW.

CrossCompute is a website where you can create and run tools for analyzing open data.



## Merge Tables

If two tables have the same column, then we can [merge them into a single table by matching rows](#).

Thanks to the [National Centers for Environmental Information](#) for collecting temperature and precipitation data by state.

```
{ a_table : Table 1 ? Table to Include in Merge }
```

```
{ b_table : Table 2 ? Table to Include in Merge }
```

```
{ key_column_name : Key Column ? Column to Use for Merge }
```

```
In [ ]: # Click the Blue Button to preview this as a CrossCompute Tool
a_table_path = 'usa-temperature-by-state.csv'
b_table_path = 'usa-precipitation-by-state.csv'
key_column_name = 'State'
target_folder = '/tmp'
```

```
In [ ]: import pandas as pd
a_table = pd.read_csv(a_table_path)
b_table = pd.read_csv(b_table_path)
c_table = pd.merge(a_table, b_table, on=key_column_name)
```

```
In [ ]: from os.path import join
target_path = join(target_folder, 'table.csv')
c_table.to_csv(target_path, index=False)
print('c_table_path = ' + target_path)
```

## Merge Tables

If two tables have the same column, then we can [merge them into a single table by matching rows](#).

Thanks to the [National Centers for Environmental Information](#) for collecting temperature and precipitation data by state.

### Table 1

Upload

State	Average Annual Temperature in Fahrenheit from 1971 to 2000
Alabama	62.8
Alaska	26.6
Arizona	60.3
Arkansas	60.4
California	59.4
Colorado	45.1
Connecticut	49.0

### Table 2

Upload

State	Average Annual Precipitation in Inches from 1971 to 2000
Alabama	58.3
Alaska	22.5
Arizona	13.6
Arkansas	50.6
California	22.2
Colorado	15.9
Connecticut	49.0

### Key Column

State

Run

If you can write code in Jupyter Notebook, you can create tools on the CrossCompute website that thousands of people around the world can run.

## Calculate Federal Income Tax for Grad Students under the House Tax Plan ★

Based on [excel sheet](#) from [this npr article](#)

Total Income ⓘ

35000

Health Insurance Plan ⓘ

4304

Tuition ⓘ

17500

Fellowship? ⓘ

0

Who can see this result? *Anyone with the link*

Public Hidden Secret

What name would like to give to this result?

Leave blank for no name

What type of worker do you want to use to generate your result?

Automated Manual

Which environment would you like to run?

Numerical Computational

How much memory would you like to reserve?

Tiny Small Medium Large Huge

How much time would you like to reserve?

1 minute 10 minutes 30 minutes 60 minutes 180 minutes

360 minutes

How much will this result cost?

Pay Tool Creator: Salah Ahmed	0
Set Visibility: HIDDEN	0
Set Container: Numerical CPU with 320mb memory for 1 minute	0
Total	0

Run for Free

Tool Creator

Salah Ahmed (0  
credits / run)

Tool Created

20171204-1455 (9  
days ago)

Tool Notebook

Calculate Federal  
Income Tax for Grad  
Students under the  
House Tax Plan



## Calculate Federal Income Tax for Grad Students under the House Tax Plan

Based on [excel sheet](#) from [this npr article](#)

{total\_income\_int: Total Income ? Enter the total amount of your annual take-home income (i.e. money that you get as a paycheck)}

{health\_insurance\_plan\_int: Health Insurance Plan ? Enter the amount of your health insurance plan if your institution gives you one as part of your appointment.}

{tuition\_int: Tuition ? Enter the amount of tuition you have had waived}

{fellowship\_int: Fellowship? ? no = 0}

```
In [1]: # Crosscompute
total_income_int = 35000
health_insurance_plan_int = 4304
tuition_int = 17500
fellowship_int = 0
target_folder = '.'

In [2]: fellowship = 1 if fellowship_int != 0 else 0

In [3]: def get_taxable_income(income, std_deduction, personal_exemption):
    return income - std_deduction - personal_exemption

In [4]: # current law
taxable_income = get_taxable_income(total_income_int, 6350, 4050)

In [5]: taxable_income
Out[5]: 24600

In [6]: if taxable_income >= 0 and taxable_income <= 9325:
    curr_tax = .1 * taxable_income
elif taxable_income > 9325 and taxable_income <= 37950:
    curr_tax = 932.5 + (0.15 * (taxable_income - 9325))
elif taxable_income > 37950 and taxable_income <= 91900:
    curr_tax = 5526.25 + (0.25 * (taxable_income - 37950))

In [7]: text = ['current taxable income = %d' % taxable_income]
text.append('current tax = %d' % curr_tax)

In [8]: # tax cuts and jobs act
income = total_income_int + health_insurance_plan_int + (tuition_int *
new_taxable_income = get_taxable_income(income, 12200, 0)

In [9]: new_taxable_income
Out[9]: 44604

In [10]: if new_taxable_income >= 0 and new_taxable_income <= 45000:
    new_tax = .12 * new_taxable_income
elif new_taxable_income > 45000 and new_taxable_income <= 200000:
    new_tax = 5400 + (0.25 * (new_taxable_income - 45000))

In [11]: text.append('new taxable income = %d' % new_taxable_income)
text.append('new tax = %d' % new_tax)

In [12]: tax_incr = new_tax - curr_tax

In [13]: percent_incr = (tax_incr / curr_tax) * 100

In [14]: percent_incr
Out[14]: 66.03272586273748

In [15]: text.append('tax increase (dollars) = %d' % tax_incr)
text.append('tax increase (percentage) = %.2f%%' % percent_incr)

In [16]: from os.path import join
path = join(target_folder, 'results.txt')
with open(path, 'w') as f:
    f.writelines([x + '\n' for x in text])
print('results_text_path = %s' % path)
results_text_path = ./results.txt
```

You do not have to create a tool from scratch -- you can improve a tool that someone else has made.

# Build a Human Trafficking Dataset from Court Cases and News Articles 20171214



Which environment would you like to run?

Numerical Computational

How much memory would you like to reserve?

Tiny Small Medium Large Huge

How much time would you like to reserve?

10 minutes 30 minutes 60 minutes 180 minutes 360 minutes

Would you like to save changes made during the session?

☒ Save changes

How much will this session cost?

Pay Notebook Creator: Yourself	0
Reimburse Yourself	0
Set Container: Computational CPU with 4000mb Memory for 10 minutes	9
Total	0

Run for 9 Credits or \$0.07

Notebook Creator

[Roy Hyunjin Han \(0 credits / run\)](#)

Notebook Created

[20170726-0458 \(6 months ago\)](#)

Notebook Modified

[20180126-1635 \(8 hours ago\)](#)

Notebook Contents

[Explore USA DOJ Court Cases](#)

[Extract Human Trafficking Incidents from Court Cases using NLTK](#)

[Extract Human Trafficking Incidents from Court Cases using spaCy](#)

[Extract Text from a Webpage](#)

[Guess Gender of a Name from the USA](#)

[Normalize Raw Text](#)

If you join us for the workshop, we will work in pairs to create a tool that extracts human trafficking incidents from court cases and news articles.



YOU can influence policy by creating tools that let policymakers explore the impact of their decisions.

If you know how to write code in Jupyter Notebook, you can create tools on the CrossCompute website that thousands of people around the world can run.



I hope we have inspired you to create tools to fight for freedom and influence policy.



# Workshop Exercises

[bit.ly/cc-text](https://bit.ly/cc-text)

Thank you to [Civic Hall](#), [NYC PyLadies](#)  
and the [Python Software Foundation](#) for  
making [this event](#) possible