

Week 01 · 데이터 저널리즘

## Introduction to Data Journalism

---

Joonhwan Lee  
human-computer interaction + design lab.

## 오늘 다룰 내용

---

- 수업 소개
- 데이터 저널리즘(data journalism)이란?
- 프로그래밍(programming)이란?
- 파이썬(Python)이란 무엇인가?
- 파이썬(Python)을 사용해 보자.

# 1. 수업 소개

---

---

## 담당 교수 및 TA

- ◆ 담당교수: 이준환
  - ◆ Email: [joonhwan@snu.ac.kr](mailto:joonhwan@snu.ac.kr)
  - ◆ Office: 64동 405호
  - ◆ TA: 이규호 ([artandplay@snu.ac.kr](mailto:artandplay@snu.ac.kr))  
은진수 ([eunjs9071@gmail.com](mailto:eunjs9071@gmail.com))
- ◆ 수업 홈페이지
  - ◆ <http://hcid.snu.ac.kr/courses/data-journalism-2019>

# 수업 개요

- ◆ 데이터 저널리즘은 디지털 사회에서의 정보의 생산과 확산 과정에서 점차 중요해지는 데이터의 역할을 다루는 학문 분야이다. 본 과목에서는 데이터 저널리즘의 이해와 더불어 이를 수행할 수 있는 컴퓨팅 기술을 학습하는 것을 목적으로 한다. 이를 위해 1) 데이터 저널리즘의 개념과 사례를 조사, 학습하고 2) 파이썬 프로그래밍 언어를 통한 데이터 분석 기법을 학습하고 이를 데이터 저널리즘에 활용하는 방법을 다룬다.
- ◆ 또한 본 수업에서는 팀별 프로젝트를 통해 실전적인 데이터 저널리즘에 대한 이해와 해당 분야에 대한 포괄적 경험을 구축하고자 한다.

---

## 강의 내용

- ◆ Python 을 이용한 기초 프로그래밍
- ◆ Python 기반의 라이브러리를 이용한 데이터 처리 프로그래밍
- ◆ 웹 기반 기술(web technology)
- ◆ 소셜 데이터 마이닝(social data mining)을 통한 데이터 분석

---

## 수업진행 계획

- ❖ 강의 계획서 참고
  - ❖ <http://hcid.snu.ac.kr/courses/data-journalism-2019>

---

## 과제 및 평가

- ❖ 강의 계획서 참고
  - ❖ <http://hcid.snu.ac.kr/courses/data-journalism-2019>

## 2. 데이터 저널리즘(Data Journalism) 이란?

---

---

# Data Journalism

- ❖ **Telling stories with data - data as a source**

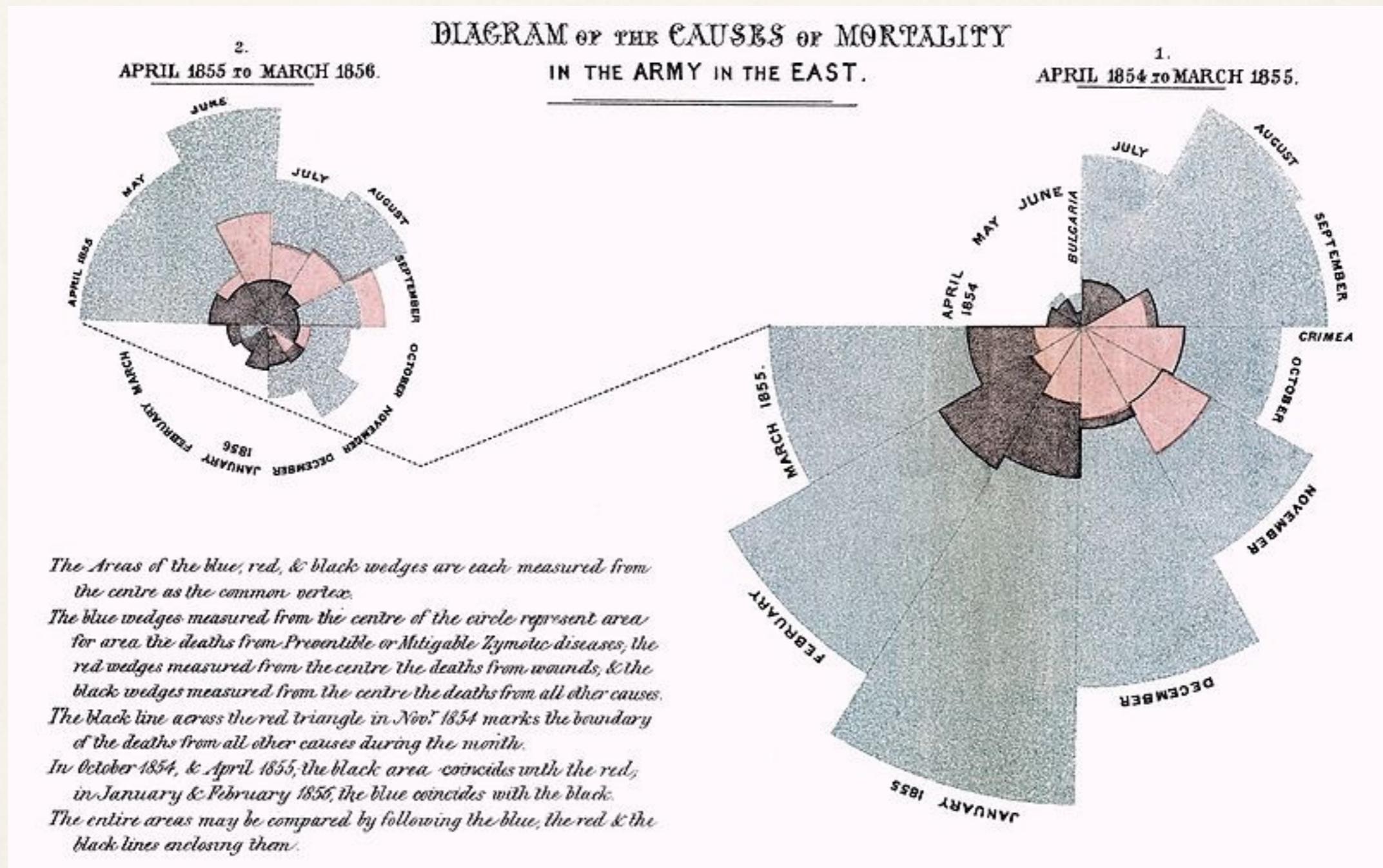
# Data Journalism

- ❖ 저널리즘 (journalism)
  - ❖ 매스미디어를 통해 공공적인 사실이나 사건에 관한 정보를 보도하고 논평하는 활동, 특히 시사적 문제의 보도·논평의 사회적 전달활동을 의미한다
- ❖ 데이터 저널리즘 (data journalism)
  - ❖ (디지털) 데이터를 통해 사실이나 사건을 보도하는 활동

# 데이터 저널리즘의 오래된 사례

- ◆ 나이팅게일 보고서
  - ◆ 1854년 크림 전쟁 당시 위생 상태에 의해 사망하는 사람이 많다는 점을 발견하고, 사람들을 설득하기 위해 데이터를 시각화한 자료를 활용
  - ◆ 위생 개선을 통해 42%에 이르던 사망률을 2%까지 줄일 수 있다는 점을 그래프로 보여 주었음

# 데이터 저널리즘의 오래된 사례



# Visual Thinking through Data

- ◆ 19세기 중반까지 유럽인들은 콜레라가 하수나 폐수에서 발생하는 독기(毒氣), 즉 나쁜 공기에 의해 발생한다고 믿었다. 1854년, 이를 의심하던 영국의 젊은 의사 존 스노우는 런던에서 발생한 콜레라 집단 발병이 눈에 보이지 않는 작은 벌레에 의한 것일 수 있다는 생각으로 이를 추적하기로 한다.
- ◆ 이듬해 런던의 소호에서 콜레라가 다시 집단 발생했을 때, 스노는 가가호호(家家戶戶)를 방문하면서 콜레라 환자를 조사했다. 그리고 환자가 발생한 지역과 환자 수를 지도에 일일이 기입했다.

서울대 자연과학대학 천종식 교수, 조선일보 (2006)

# Visual Thinking through Data



# Visual Thinking through Data



# Visual Thinking through Data

- ♦ 어느 정도 일이 진척되자 슬슬 수수께끼가 풀렸다. 가구당 콜레라 환자의 수는 브로드가(街)의 우물에 가까울수록 많다는 것을 발견한 것이다. 그리고 환자 대부분이 이 우물에서 물을 길어 식수로 이용했다는 사실도 확인했다. 콜레라를 일으킨다고 믿던 독기(毒氣)를 중화시키기 위해 거리에 석회를 열심히 뿌리던 런던 시당국은 그에게 속는 셈치고 우물을 폐쇄했다. 그러자 기적같이 콜레라 전염이 멈추게 된 것이다.

서울대 자연과학대학 천종식 교수, 조선일보 (2006)

데이터의 시각화는 보이지 않는 문제점을 발견하게 해준다.

# KBS, 왜 “데이터 저널리즘”인가?

<http://news.kbs.co.kr/news/view.do?ncd=3028270>



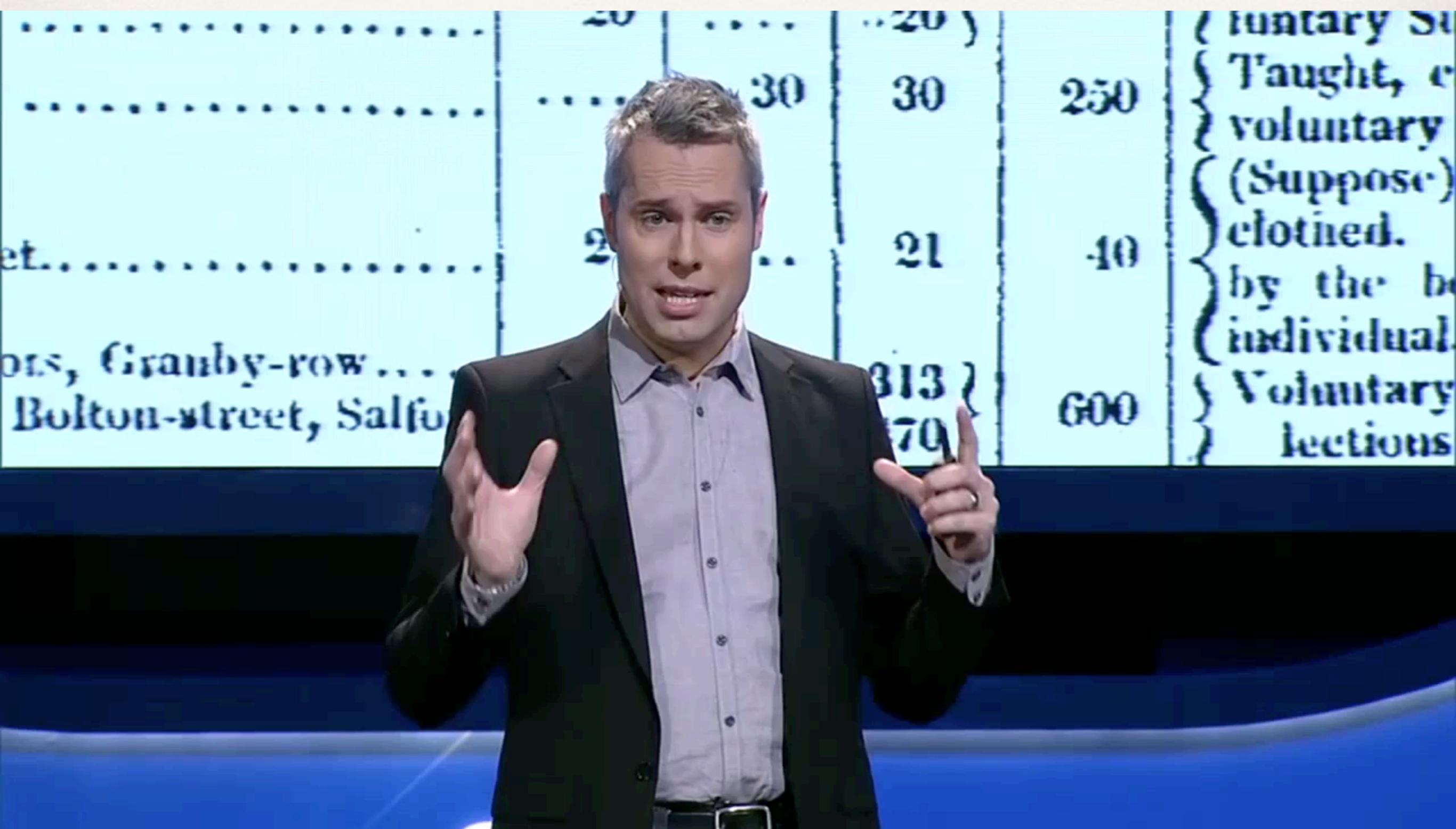
# 뉴스타파: 프로퍼블리카 설립자 인터뷰

<https://www.youtube.com/watch?v=nkqQqQn9SvQ>



## [SDF 2013] 열린 데이터 저널리즘 (사이먼 로저스)

<https://www.youtube.com/watch?v=wLYhrY1qgag>



# New York Times

## New York Times launches data journalism site The Upshot

Publisher's 'conversational' replacement for Nate Silver's FiveThirtyEight takes on Vox and Washington Post's Wonkblog

The screenshot shows the homepage of The Upshot. At the top, there is a navigation bar with links for 'SEARCH', 'The New York Times', 'COLUMBIA UNIVERSITY', 'MASTER OF SCIENCE IN COMMUNICATIONS PRACTICE FULL-TIME GRADUATE PROGRAM', 'Learn More', and a 'SUBSCRIBE' button with arrows. Below the navigation is the 'The Upshot' logo, which consists of a yellow dot followed by the word 'TheUpshot' in a bold, black, sans-serif font. To the right of the logo, it says 'EDITED BY DAVID LEONHARDT' and 'FOLLOW US' with links to Facebook and Twitter. The main visual is a large, stylized illustration of a brain composed of colored dots (yellow, blue, red, grey) with figures in lab coats interacting with it. Below this, the headline 'Who Will Win the Senate?' is displayed. A small note below the headline states: 'According to our statistical model that forecasts the election, it's a tossup.' and 'This document has about a 1% chance of containing a mistake.' At the bottom left, there is a camera icon and the text 'The New York Times's The Upshot website'. On the right side, there is a photo of a woman in a red jacket in a kitchen, with the caption 'Nicole Bengiveno/The New York Times' and the headline 'The American Middle Class Is No Longer the World's Richest'.

URCH

The New York Times

COLUMBIA UNIVERSITY

MASTER OF SCIENCE IN COMMUNICATIONS PRACTICE

FULL-TIME GRADUATE PROGRAM

Learn More

SUBSCRIBE

EDITED BY DAVID LEONHARDT

FOLLOW US [f](#) [t](#)

**TheUpshot**

Who Will Win the Senate?

According to our statistical model that forecasts the election, it's a tossup.  
This document has about a 1% chance of containing a mistake.

Nicole Bengiveno/The New York Times

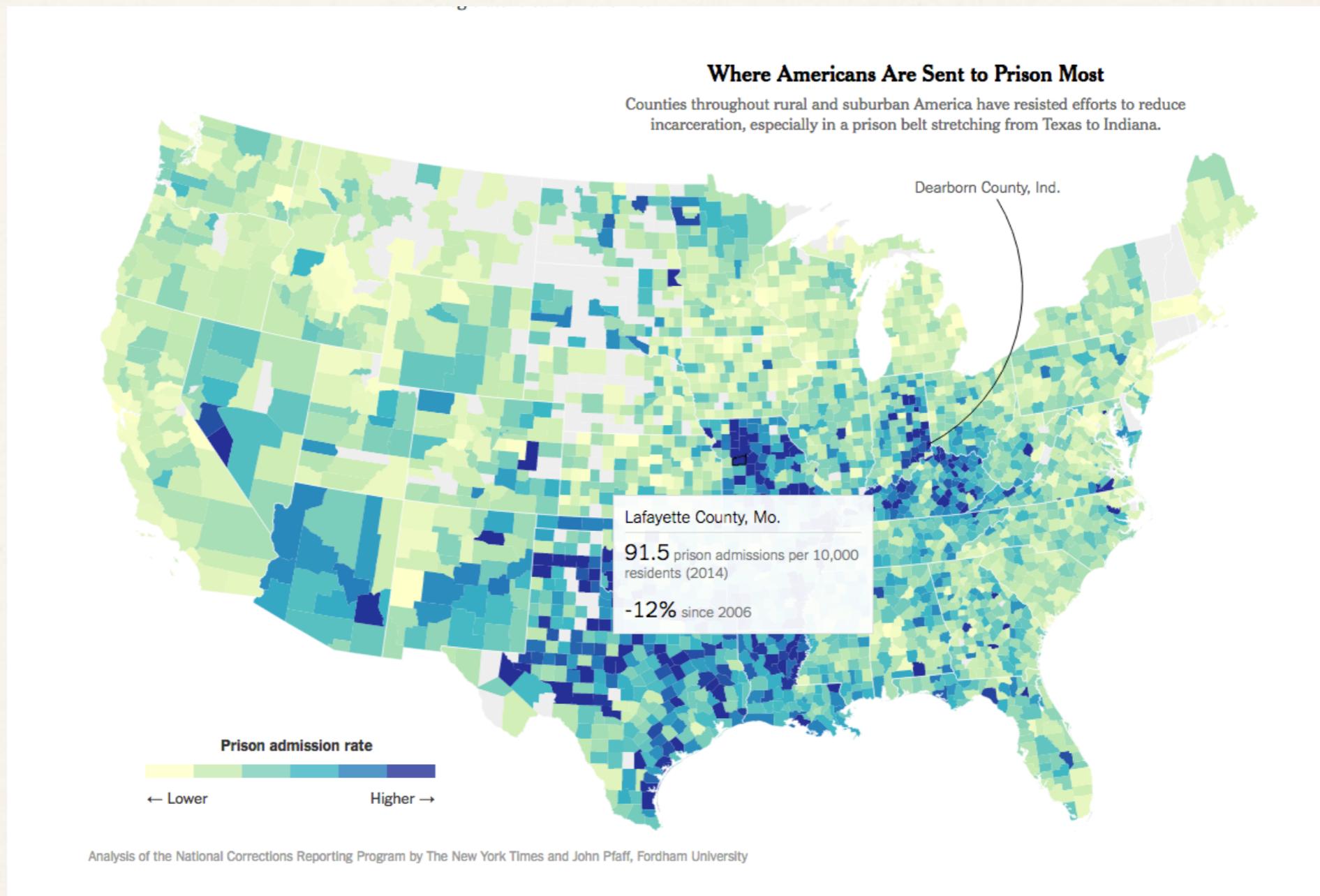
**The American Middle Class Is No Longer the World's Richest**

After three decades of slow growth, median income in the U.S. trails that of Canada. Poor Americans now make less than the poor in several other countries.

[Read more](#) [Email](#) [RSS](#) [Print](#)

The New York Times's The Upshot website

# The Upshot



<http://www.nytimes.com/2016/09/02/upshot/new-geography-of-prisons.html?rref=collection%2Fsectioncollection%2Fupshot&action=click&contentCollection=upshot&region=rank&module=package&version=highlights&contentPlacement=3&pgtype=sectionfront>

# FiveThirtyEight



Politics Sports Science & Health Economics Culture



WEATHER

## We Haven't Seen Many Storms Like Hermine

By Eric Holthaus

2016 ELECTION

### 13 Tips For Reading 2016 Election Polls Like A Pro

By Harry Enten

#### THE LATEST

SEP 3

We Haven't Seen Many Storms Like Hermine

SEP 2

How The MLB Standings Would Look If Teams Stuck With Whoever They Drafted

SEP 2

Internet Tracking Has Moved Beyond Cookies

SEP 2

Trump and Clinton Can Both Spin The Latest Jobs Report

SEP 2

The Ultimate Guide To How Much To Spend On Wedding Gifts

SEP 2

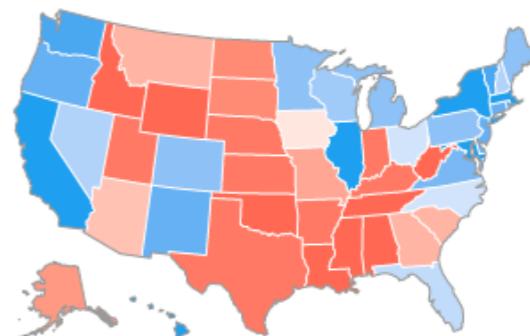
Significant Digits For

New episode: Elections podcast

#### INTERACTIVES

### 2016 Election Forecast

UPDATED 21 HOURS AGO



See polls and forecasts

### U.S. Open Predictions

UPDATED 8 HOURS AGO

#### Women

Serena Williams 60%

Angelique Kerber 18%

#### Men

Novak Djokovic 67%

Andy Murray 16%

<https://fivethirtyeight.com/>

### 3. 프로그래밍(programming) 이란?

---

# Ciao Espresso~



fully automatic • serves 2 shots • \$3,000 or more?

## Ciao Espresso~



fully automatic • serves 2 shots • \$3,000 or more?

---

# Ciao Espresso~



---

# Ciao Espresso~

- ❖ Short Instruction
  - ❖ 그라인더에 간 커피를 portafilter에 넣고 꾹꾹 잘 누른 다음 커피머신에 돌려서 끼우고 버튼만 누르면 됨.

---

## Ciao Espresso~

- ♦ portafilter에 뜨거운 물을 내려부어 따뜻하게 한다
- ♦ 커피를 portafilter에 담는다
- ♦ portafilter를 커피머신에 돌려 끼운다
- ♦ 에스프레소 잔을 올려놓고 버튼을 누른다

# Ciao Espresso~

- ◆ 물이 없다면
  - ◆ 물탱크에 물을 담는다
- ◆ 물이 있다면
  - ◆ portafilter에 뜨거운 물을 내려부어 따뜻하게 한다
- ◆ 커피가 없다면
  - ◆ 커피를 그라인드로 간다
- ◆ 커피가 있다면
  - ◆ 커피를 portafilter에 담는다
- ◆ portafilter를 커피머신에 돌려 끼운다

# Ciao Espresso~

- ◆ 한잔만 만들 경우
  - ◆ 에스프레소 잔을 올려놓고 버튼을 누른다
- ◆ 한잔 이상일 경우
  - ◆ 에스프레소 잔을 올려놓고 버튼을 누른다를 반복한다
- ◆ 아메리카노일 경우
  - ◆ 큰 컵에 뜨거운 물을 붓는다
  - ◆ 에스프레소 샷을 담는다
- ◆ 아이스 아메리카노일 경우
  - ◆ 큰 컵에 얼음을 넣는다
  - ◆ 찬 물을 붓는다
  - ◆ 에스프레소 샷을 담는다

# 프로그래밍 (Programming)

- ◆ 컴퓨터라는 지능을 가지고 있지 못한 기계에게 일의 순서를 알려주는 것
  - ◆ 모든 컴퓨터는 내부에 CPU를 가지고 있는데, 이 CPU는 다양한 명령어 집합(instruction sets)을 가지고 있다
  - ◆ 프로그래밍은 각각의 명령어 집합에 순차적으로 명령(instruction)을 내리는 과정
- ◆ 즉, 컴퓨터가 어떠한 일을 어떻게 처리해야 할지를 알려주는 방법

# 프로그래밍 언어 (Programming Language)

- ❖ A programming language is an **artificial language** designed **to communicate instructions to a machine**, particularly a computer. Programming languages can be used to create programs that control the behavior of a machine and/or to express algorithms precisely.
  - wikipedia ([http://en.wikipedia.org/wiki/Programming\\_language](http://en.wikipedia.org/wiki/Programming_language))

# 프로그래밍 언어 (Programming Language)

- ♦ 어플리케이션을 개발하기 위해서는 프로그램 언어가 필요
  - ♦ 프로그래머 → 프로그래밍 언어 → 프로그램 작성  
(programming) → 해석기 (compiler) → 실행
  - ♦ 프로그래밍 언어의 종류 (언어 표현의 난이도)
    - ♦ 저수준 언어 (Low-level Language)
    - ♦ 고수준 언어 (High-level Language)

# 프로그래밍 언어 (Programming Language)

## ♦ 저수준 언어 (Low-level Language)

- ♦ CPU가 직접 해독하고 실행이 가능하도록 비트 단위로 쓰여진 언어: 기계어 (Machine Language)
  - ♦ 컴퓨터 실행에는 효율적이지만, 사람이 이해할 수가 없어 작성하기에 불편
    - 유지/보수 거의 불가능
  - ♦ Machine Language (기계어) 와 Assembly Language 가 대표적

### Machine Language

```
169 1 160 0 153 0 128 153 0 129 153 130 153 0 131 241 96
```

### BASIC

```
5 FOR I=1 TO 1000: PRINT "A";: NEXT I
```

# 프로그래밍 언어 (Programming Language)

- ♦ 저수준 언어 (Low-level Language)
  - ♦ Assembly Language
  - ♦ 기계어 명령을 어느 정도 해독할 수 있도록 문자화 한 형태의 언어
  - ♦ CPU/OS 에 따라 명령어 구조가 다름
  - ♦ 어셈블리어도 기본적으로 기계어로 번역되는 과정을 거쳐야
  - ♦ 고수준 언어보다 빠른 처리 속도  
→ 임베디드 프로그램 작성 시에 많이 사용

```
section .text
global _start
_start:
    mov edx, len
    mov ecx, msg
    mov ebx, 1
    mov eax, 4
    int 0x80

    mov eax, 1
    int 0x80

section .data
msg db 'Hello, world!', 0xa
len equ $ - msg
```

# 프로그래밍 언어 (Programming Language)

- ❖ 고수준 언어 (High-level Language)
  - ❖ 어셈블리어의 문제점
    - ❖ 사람이 해독하기 여전히 어렵다
    - ❖ 컴퓨터 시스템 (CPU) 마다 서로 다른 명령어를 사용해야 한다
  - ❖ 일상적으로 사용하는 언어와 비슷한 구조의 랭귀지 등장
  - ❖ 컴파일러 통해 기계어로 해석하기 때문에 랭귀지 레벨에서는 시스템에 따라 다른 명령어를 사용할 필요가 없다
  - ❖ Fortran, Cobol, C, Pascal, Basic, C++, Java, Processing, Objective-C, Python, Ruby 등등

# 프로그래밍 언어 (Programming Language)

## #JAVA

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```

## #ANSI C

```
#include <stdio.h>  
  
/* Hello */  
int main(void)  
{  
    printf("Hello, World!");  
    return 0;  
}
```

## #Ruby

```
puts "Hello, World!"
```

## #Objective-C/Cocoa

```
#import <Foundation/Foundation.h>  
  
int main (int argc, const char * argv[ ]) {  
    NSLog(@"Hello, World!");  
    return 0;  
}
```

<http://www.scriptol.com/programming/hello-world.php>

# 컴파일러 (Compiler)

- ❖ 기계어 이외의 다른 언어로 작성된 프로그램 → 기계어로 번역되어야 실행가능
- ❖ 컴파일러
  - ❖ 고수준 언어로 작성된 소스코드 (Source Code)를 기계어로 번역한 후 오브젝트코드 (Object Code)로 저장하는 프로그램
  - ❖ Object Code: 기계어로 변환된 프로그램
    - ❖ .exe, .com, .app (mac os x) 과 같은 실행파일이 이에 해당
    - ❖ 이미 기계어로 변환된 상태라 실행 속도가 빠름
    - ❖ 기계어로 암호화 되어 소스코드를 볼 수 없다 → decompile 이 가능한 경우 도... (decompile: 원 소스 코드 상태로 되돌리는 것)
    - ❖ 오류나 구조 변경 시, 재 compile 과정을 거쳐야

# 인터프리터 (Interpreter)

- ❖ Compiler 와 같이 미리 기계어로 번역된 프로그램을 만드는 것이 아니라 소스코드를 그때 그때 기계어로 해석하여 실행
- ❖ BASIC
- ❖ Ruby
- ❖ Python
- ❖ JavaScript
- ❖ 컴파일된 어플리케이션 보다 실행속도 느림 → CPU 파워의 발전으로 어느 정도 해소
- ❖ 개발/업데이트 용이 → 프로토타이핑 툴에 많이 사용

# 프로그래밍 언어의 발전과정

- ♦ 초기의 프로그래밍 언어
  - ♦ 어셈블리 언어의 등장 → 기계어 대체 언어로 초기에 많이 사용
  - ♦ 기계어 보다 쉬우나 여전히 배우기는 어려운 언어
  - ♦ 반도체나 산업기계 제어 프로그램과 같은 영역에서만 사용
- ♦ 최초의 고수준 프로그래밍 언어
  - ♦ FORTRAN (FORmula TRANslator):  
1954년 공학계산을 위한 프로그래밍 랭귀지로 개발 → 우주선 궤도 계산, 선박 설계 등 과학 계산에 (아직도) 많이 사용 됨.
  - ♦ COBOL: 데이터 처리 목적으로 개발  
구조가 구어체 처럼 간단하고 읽기 쉬워 회계처리, 비즈니스 등에 주로 활용

# 프로그래밍 언어의 발전과정

- ♦ Procedural Programming Language (절차적 프로그래밍 언어)
  - ♦ 소프트웨어 개발이 활발해지면서 보다 체계적인 언어가 필요
  - ♦ 수행할 내용을 순서대로 체계적으로 작성해 주는 절차적 프로그래밍 언어가 등장
    - ♦ 이전의 코드는 필요에 따라 코드 내에서 이리 저리 이동하여 복잡도가 증가
  - ♦ 입력, 계산, 데이터 처리, 출력 등의 과정을 체계적으로 반복
    - ♦ Structured Program (구조적 프로그래밍) → if/else, for, while 등 제어문 사용
    - ♦ 프로그램 구조가 인덱스 페이지와 챕터처럼 구성
  - ♦ 대표적인 절차적 프로그래밍 언어: Pascal, C, BASIC, PL/1, Ada 등

# 프로그래밍 언어의 발전과정

- ❖ C Language
  - ❖ 1970년대 초반 AT&T Bell Lab
  - ❖ UNIX 운영체제의 프로그램을 개발하기 위해 개발
  - ❖ 현재 대부분의 어플리케이션을 개발하는데 사용
  - ❖ 객체지향형의 C++ 과 Objective-C 등 많은 프로그래밍 언어로 발전
  - ❖ 성능이 좋아 다양한 분야에 사용
    - ❖ 어플리케이션 개발, 반도체 제어, 산업기계 제어 등

# 프로그래밍 언어의 발전과정

- ❖ Object Oriented Programming Language (객체지향 프로그래밍 언어)
  - ❖ 절차적 프로그래밍 언어 → 컴퓨터가 수행할 작업 순서대로 프로그램 작성
  - ❖ OOP
    - ❖ 컴퓨터 프로그램: 명령어(함수)의 집단 → 객체(Object)의 모임
    - ❖ 각각의 객체는 데이터를 포함하고 있고, 그 데이터를 처리하기 위한 방법(method)들로 구성되어 있다 → Class
    - ❖ 해당 객체의 데이터 처리는 객체 내에서만 이루어 진다 → 전체 프로그램에 영향 X
    - ❖ 프로그램을 객체별로 분리하여 취급하므로 재사용과 수정이 용이
    - ❖ Objective-C, SmallTalk, C++, Java, C# 등이 대표적 언어

# 프로그래밍 언어의 발전과정

- ❖ C++
  - ❖ 1980년대 AT&T Bell Lab
  - ❖ C 언어를 확장하여 클래스, 객체, 이벤트 등의 객체지향 개념을 적용
- ❖ Java
  - ❖ 1980년대 후반 Sun Microsystems
  - ❖ 웹어플리케이션, 일반 어플리케이션, 소형 가전기기의 임베디드 OS에 사용
  - ❖ 네트워크 플랫폼을 겨냥 → 서버 어플리케이션용 랭귀지로 발전
  - ❖ 컴파일러 → (중간단계의) Byte Code 생성 → JavaVM에서 어플리케이션 실행 (Platform independent)

# 왜 “프로그래밍”을 배워야 하는가?

*“We believe every child should have the opportunity to learn computer science, from primary school up to and including further education. We teach elementary physics to every child, not primarily to train physicists but because each of them lives in a world governed by physical systems. In the same way, **every child should learn some computer science from an early age because they live in a world in which computation is ubiquitous**. A crucial minority will go on to become the engineers and entrepreneurs who drive the digital economy, so there is a complementary economic motivation for transforming the curriculum.”*

The Guardian, March 31, 2012

(<http://www.guardian.co.uk/education/2012/mar/31/manifesto-teaching-ict-education-minister>)

# 왜 “프로그래밍”을 배워야 하는가?

- ♦ 컴퓨터에 의해 움직이는 네트워크 사회에 살고 있기 때문
  - ♦ 컴퓨터가 만들어 낸 다양한 컨셉들에 대해 이해할 필요
- ♦ 문제 해결을 위한 새로운 사고가 도래: computational thinking
  - ♦ understanding difference between human and artificial intelligence
  - ♦ 해결하지 못하던 수많은 문제를 해결
- ♦ 이로 인해 새로 등장한 개념들
  - ♦ algorithms, cryptography, machine intelligence, search, computational X 등

# 왜 “프로그래밍”을 배워야 하는가?

- ◆ 사회과학자의 입장

- ◆ 디지털미디어의 급속한 확산 → 전통적인 매스미디어 모델에서 벗어나 개인 미디어나 소셜 미디어를 통한 정보의 생성, 전달, 소비가 활발하게 이루어짐.
- ◆ 소수에 의해 정보가 생산되는 매스미디어에 비해 개인 미디어나 소셜 미디어를 통해서 생산되는 정보의 양은 엄청나며, 이를 분석하기 위해서는 기존 연구방법을 뛰어 넘는 새로운 데이터 중심의 연구(data driven research) 패러다임이 요구되고 있음.
- ◆ 소셜 미디어와 그 안에서 데이터가 생성되고 공유되는 방식을 이해하기 위해 기술적인 접근이 필요.
- ◆ 생산되는 엄청난 양의 데이터를 수집, 분석하기 위해 관련 프로그래밍 기술의 습득이 필수.

# 왜 “프로그래밍”을 배워야 하는가?

- ◆ 사회과학자의 입장
  - ◆ 반면, 컴퓨터 공학 분야에서 이루어지는 소셜 미디어 연구는 사회적 함의를 가지는 질문 보다는 대부분 네트워크 자체의 형태적, 구조적 특성에 대한 묘사적 연구가 주류.
  - ◆ 전체 구조에 대한 콘텍스트는 제공하나 그 안에서 전달되는 콘텐츠에 대해서는 의미있는 방법론을 제시하지 못하고 있음.
  - ◆ 사회과학자들이 프로그래밍과 관련 기술의 습득을 통해 소셜 미디어에서의 빅 데이터 분석에 있어 기존 연구전통과 연결되는 새로운 방법론을 만들어 내야할 시점.

## 4. 파이썬(Python)이란 무엇인가?

---

# 파이썬(Python)이란 이름의 언어

- ◆ 1991년 프로그래머 Guido van Rossum 이 발표한 언어
- ◆ 텍스트 관련 처리 능력이 매우 뛰어남
- ◆ 문법이 간단하고 이해하기 쉬움
- ◆ OOP 지향 언어
  - ◆ 개발효율, 유지, 보수, 재이용성, 신뢰성을 높일 수 있음
- ◆ 인터프리터 언어
  - ◆ 실행할 때 소스코드를 기계어로 번역하기 때문에 편리하게 코딩 가능
- ◆ 오픈소스! 멀티플랫폼(mac, windows, linux 등)
- ◆ 과학 계산과 관련한 라이브러리가 많아 최근 데이터 사이언스에 많이 사용

# Python 2 or Python 3?

- ◆ Python 2
  - ◆ 라이브러리의 호환성이 좋다.
  - ◆ 기존의 코드(샘플코드)가 대부분 Python 2로 되어 있어 학습에 편리
  - ◆ 한글 인코딩이 복잡
- ◆ Python 3
  - ◆ 라이브러리의 호환성이 상대적으로 떨어진다  
(하지만 최근엔 대부분의 주요 라이브러리가 Python 3 지원)
  - ◆ 기존의 코드와 호환이 안되는 경우가 있다
    - ◆ `print("문자출력")` vs `print "문자출력"`
    - ◆ `for i in range(10):` vs `for i in xrange(10):`
  - ◆ 한글 인코딩 문제가 해결

# Python 2 or Python 3?

- ♦ Python 2
  - ♦ 라이브러리의 호환성이 좋다.
  - ♦ 기존의 코드(샘플코드)가 대부분 Python 2로 되어 있어 학습에 편리
  - ♦ 한글 인코딩이 복잡
- ♦ Python 3 **← 수업에서는 Python 3을 사용할 계획**
  - ♦ 라이브러리의 호환성이 상대적으로 떨어진다  
(하지만 최근엔 대부분의 주요 라이브러리가 Python 3 지원)
  - ♦ 기존의 코드와 호환이 안되는 경우가 있다
    - ♦ `print("문자출력")` vs `print "문자출력"`
    - ♦ `for i in range(10):` vs `for i in xrange(10):`
  - ♦ 한글 인코딩 문제가 해결

# Python의 설치

- ◆ 설치 방법은 매우 다양해서 인터넷에서 검색해서 자신에게 맞는 방법을 선택할 것
- ◆ 표준 설치 방법: <http://kybin.github.io/translateDiveIntoPython3korean/installing-python.html>

## 4. 파이썬(Python)을 사용해 보자

---

# CMD & Terminal

- Windows 나 Mac OS X 에는 키보드로 명령을 입력하기 위한 command line tool 이 있음
  - Windows: 시작 → 실행 → cmd 입력
  - Mac: Applications → Utilities → Terminal 실행
  - Python은 이들 창을 통해 실행 됨

```
C:\WINDOWS\system32\cmd.exe

ruby:
  bin:      C:/RailsInstaller/Ruby1.9.2/bin/ruby.exe
  version:   ruby 1.9.3p125 (2012-02-16) [i386-mingw32]

rails:
  bin:      C:/RailsInstaller/Ruby1.9.2/bin/rails.bat
  version:   Rails 3.2.1

ssh:
  public_key_location: C:\Documents and Settings\joonhwan/.ssh/id_rsa.pub
  public_key_contents: ssh-rsa AAAAB3NzaC1yc2EAAQEA0dL6+jXJEyu2d3woE2Rtc
whfy5TeEoJnH4hErxFh738m0kBLRzA6wE0o1cDr99jA3Unt6vzjDkYm+wQa44QpwRBoi5Zhu,jjUVCr68
YEgI+u/VqxxkmbGdLuF7/dcsJNzEfCKjWFw8+s1pnTYOU7WguIAkb7TDWUFjaD0ysJb81bDj91foXC
5tpwInwgtUBGWILPX63gsPYpJnU4t1pSu/qPSSsJ5i+NyNrpPQRDiTUofOAQYP+JKRwzenNz4hZRJC0
Ui5lqo1HmL2FIBoRcBhDKW+ILbZFBkBCeGAiEEPOKHYu5is1Yd5/q4ct8pez0TyjtMFb12/Z60KnZmEm
w== Joonhwan Lee <joonhwan@gmail.com>

C:\WSites>
C:\WSites>
C:\WSites>
C:\WSites>
```

명령프롬프트

```
var — bash — 100x30
bash

== LICENSE:
Copyright (c) 2009-2011 Wayne E. Seguin
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at
http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

monoair:var joonhwan$ rvm list
monoair:var joonhwan$ rvm list
rvm rubies
  jruby-1.6.7.2 [ x86_64 ] ruby:
  ruby-1.9.2-p318 [ x86_64 ] bin:      C:/RailsInstaller/Ruby1.9.2/bin/ruby.exe
  ruby-1.9.2-p328 [ x86_64 ] version:   ruby 1.9.3p125 (2012-02-16) [i386-mingw32]
*=* ruby-1.9.3-p194 [ x86_64 ]
                               rails:
                               # => - current           bin:      C:/RailsInstaller/Ruby1.9.2/bin/rails.bat
                               # * - current && default  bin:      C:/RailsInstaller/Ruby1.9.2/bin/rails.bat
                               # * - default            version:   Rails 3.2.1
monoair:var joonhwan$ ssh:
```

Terminal

# CMD & Terminal

## ♦ 간단한 명령어 리스트

	CMD	Terminal	example
디렉토리 이동	cd	cd	cd .. cd Documents cd ~/Desktop
디렉토리 내 파일 리스트 보기	dir	ls	-
디렉토리 생성/삭제	mkdir rmdir	mkdir rmdir	mkdir temp rmdir temp
디렉토리 이름 변경	rename	mv	rename temp1 temp2 mv temp1 temp2
파일 카피	copy	cp	copy temp1 temp2 cp temp1 temp2

# Text Editors

- ❖ 프로그램의 소스코드를 작성하기 위해 자신에 맞는 텍스트 에디터를 선택해야 함.
- ❖ 텍스트 에디터 = 일종의 워드프로세서
  - ❖ 워드프로세서처럼 텍스트의 속성이 저장되지 않는다.  
(예: 볼드 x, 칼라 x, 밑줄 x, 글자크기 x)
- ❖ 그러나 프로그래밍 전용 텍스트 에디터는 키워드, 변수 등을 다른 색으로 표시하여 손쉽게 코드를 이해할 수 있음
  - ❖ 라인넘버 제공으로 특정 코드의 위치를 표시 → 에러 발생시 유용하게 쓰임
  - ❖ Atom, MS Visual Studio Code 등

The screenshot shows a code editor window with a dark theme. The file is named 'crawl\_comments.rb'. The code is a Ruby script with several methods defined:

```
# slice(.,.*\\/) : returns ",28218468,134057910"
# - slice(1..-1) : remove the first character -> returns "28218468,134057910"
# - chop : remove the last character -> returns "28218468,134057910"
#
def getArticleCommentID(doc)
  ids = Array.new
  doc.xpath('//dd[@class="action"]/a@onclick').each do |method_span|
    ids << method_span.text.slice(.,.*\\/).slice(1..-1).chop
  end
  return ids
end

# get the number of a comment's child comment
def getChildCount(doc)
  counts = Array.new
  doc.xpath('//a[@class="reply_count"]').each do |method_span|
    counts << method_span.content.gsub!("답글", "").strip
  end
  return counts
end

# clean up json file and parse
def parseJSON(content)
  orig_title = Array.new
  clen_title = Array.new
  orig_comment = Array.new
  clen_comment = Array.new

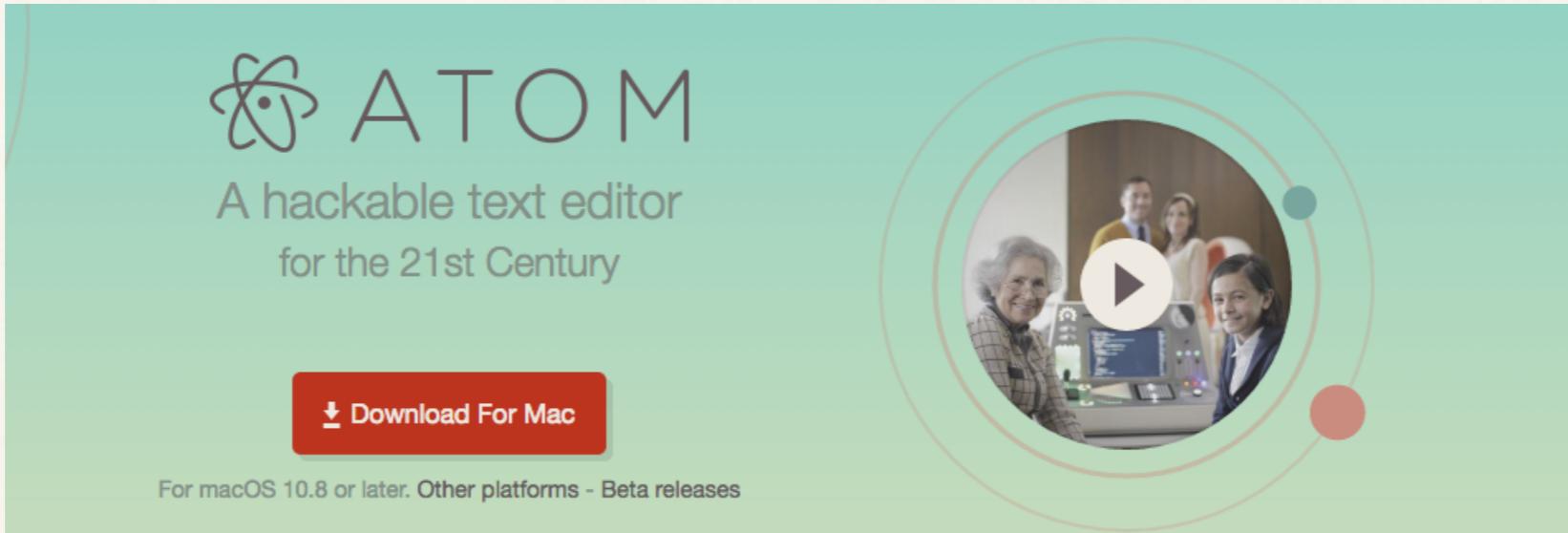
  puts "Cleaning up JSON..."
  @logfile << "Cleaning up JSON...\n"

  # scan cleannning area
  orig_title = content.scan(/articleSubject.+?articleUrl/)
  orig_comment = content.scan(/commentContent.+?rheaAdminTool/)

  # clean up title
  orig_title.each do |method_span|
    clen_title << method_span.gsub(':"', ' "Q@T").gsub(',"', ' "Q@T").gsub('","', ' "Q@T")
  end
end
```

At the bottom of the window, there are status bars showing 'Line: 160 Column: 29 | Ruby' and 'Soft Tabs: 2 Symbol'.

# Atom



The screenshot shows the official Atom website homepage. At the top left is the Atom logo (a stylized atom symbol) and the word "ATOM". Below it is the tagline "A hackable text editor for the 21st Century". A red "Download For Mac" button is prominently displayed. To the right is a video player showing three people (two men and one woman) smiling at a computer screen. Below the video is a large screenshot of the Atom code editor interface, showing a file named "atom.coffee" with CoffeeScript code. The sidebar on the left lists various project files like build, docs, and src.

For macOS 10.8 or later. Other platforms - Beta releases

atom.coffee

```
18 # Essential: Atom global for dealing with packages, themes, menus, and the window system.
19 #
20 # An instance of this class is always available as the `atom` global.
21 module.exports =
22   class Atom extends Model
23     @version: 1 # Increment this when the serialization format changes
24     # Load or create the Atom environment in the given mode.
25     #
26     # Returns an Atom instance, fully initialized.
27     @loadOrCreate: (mode) ->
28       startTime = Date.now()
29       atom = @deserialize(@loadState(mode)) ? new this({mode, @version})
30       atom.deserializeTimings.atom = Date.now() - startTime
31
32       atom
33
```

src/atom.coffee\* 31,17

UTF-8 CoffeeScript master

<https://atom.io/>

# My First Python Program!

- ❖ text editor 실행 후 다음과 같이 입력

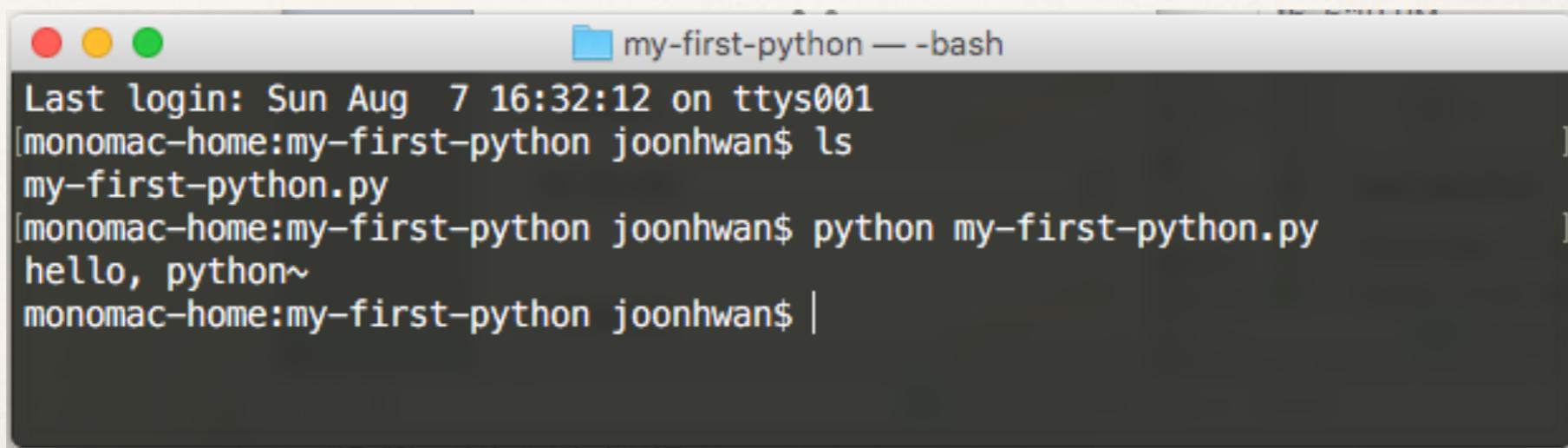


A screenshot of a Mac OS X TextEdit window. The title bar says "my-first-python.py — /Users/joonhwan/Desktop/my-first-python". The left pane shows a folder "my-first-python" containing a file "my-first-python.py". The right pane displays the code:

```
1 print("hello, python~")  
2
```

The status bar at the bottom shows "my-first-python.py 1:22" and "LF UTF-8 Python 2 updates".

- ❖ 해당 파일이 저장된 디렉토리로 이동 (cd <디렉토리>)
- ❖ python <파일명> 으로 python 프로그램 실행

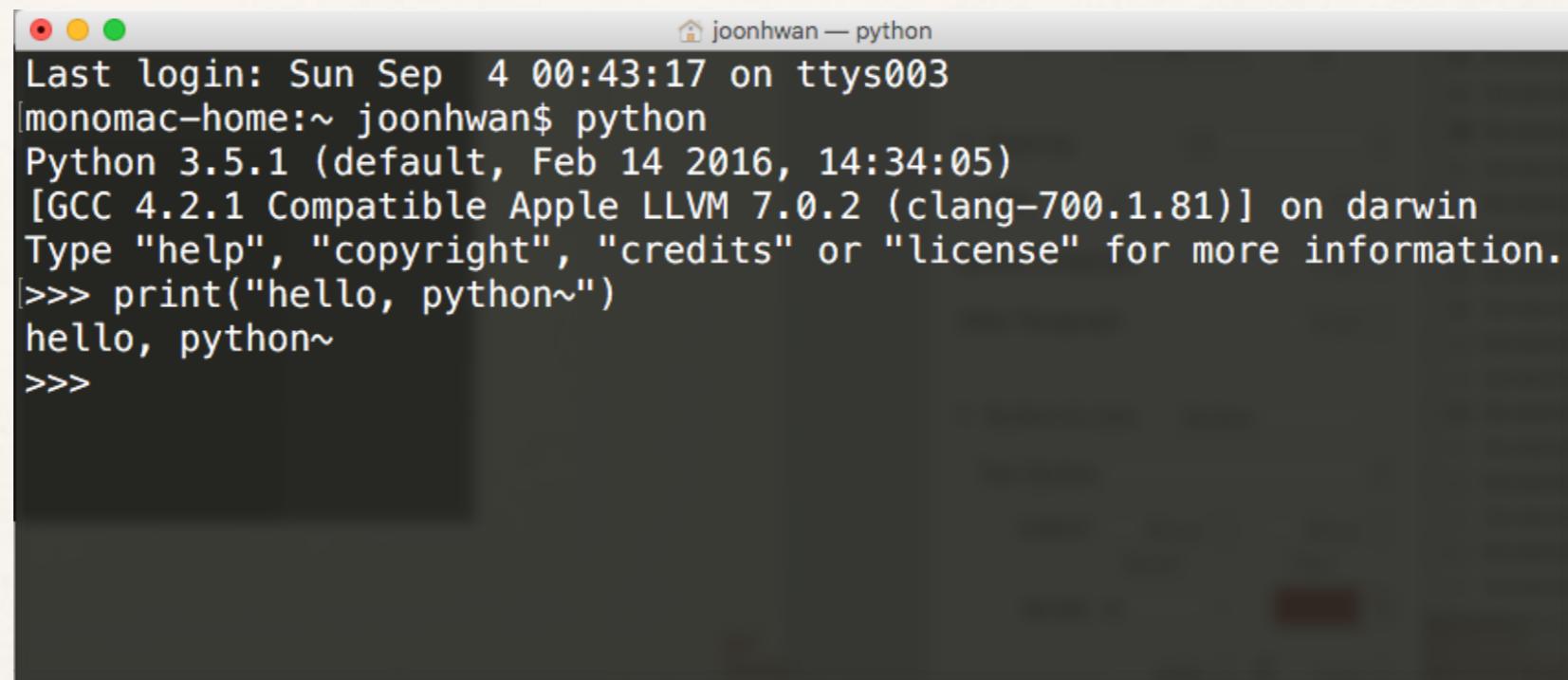


A screenshot of a Mac OS X terminal window. The title bar says "my-first-python — -bash". The terminal output is:

```
Last login: Sun Aug  7 16:32:12 on ttys001  
[monomac-home:my-first-python joonhwan$ ls  
my-first-python.py  
[monomac-home:my-first-python joonhwan$ python my-first-python.py  
hello, python~  
monomac-home:my-first-python joonhwan$ |
```

# My First Python Program!

- ❖ 파이썬을 실행하는 또 다른 방법
  - ❖ interactive shell mode

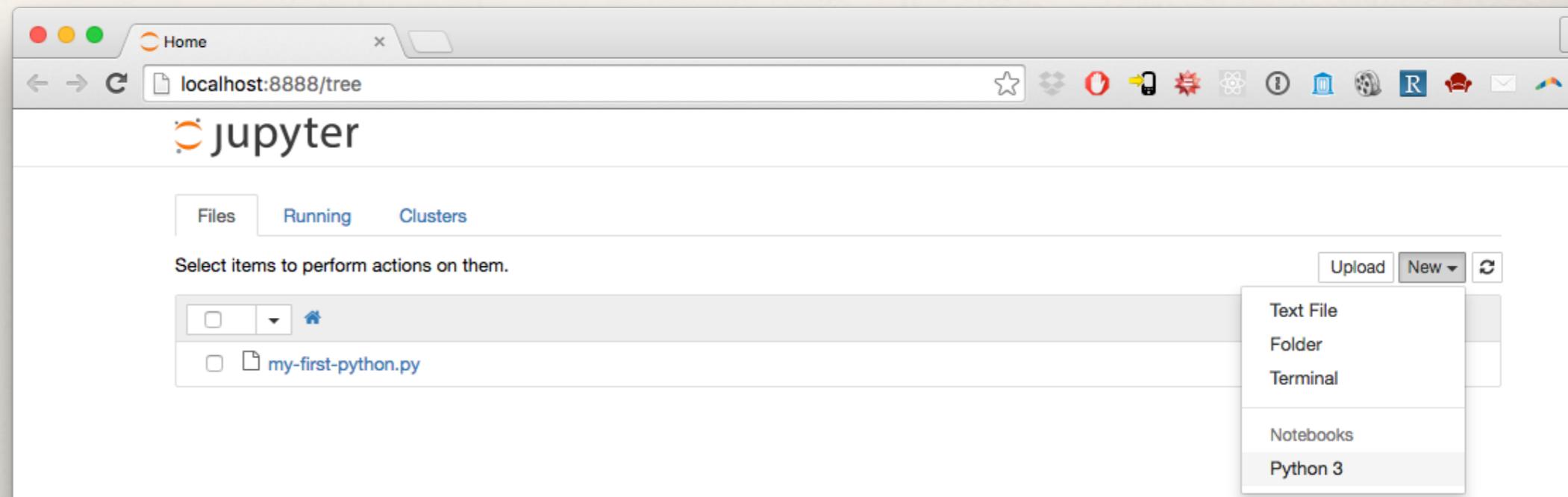


A screenshot of a Mac OS X terminal window titled "joonhwan — python". The window shows the following text:

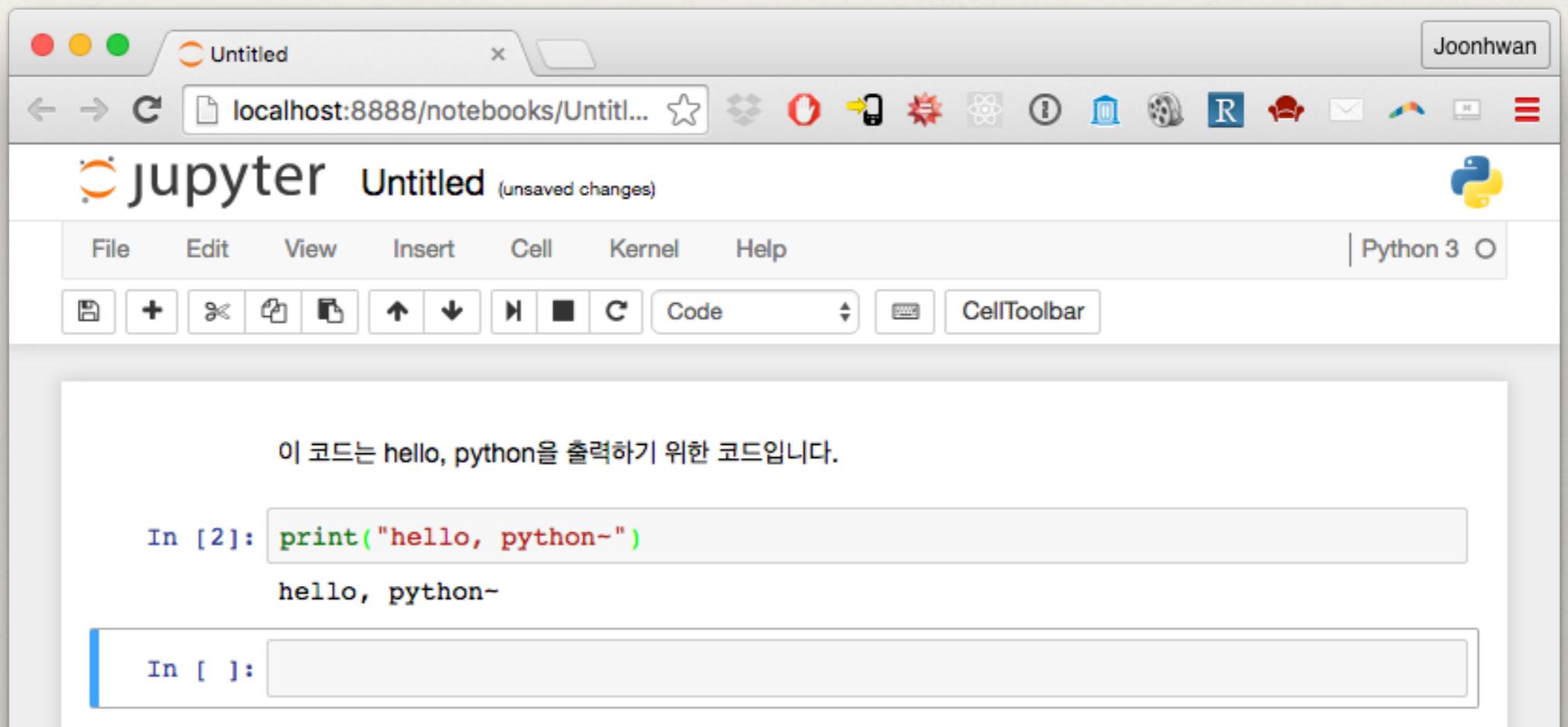
```
Last login: Sun Sep  4 00:43:17 on ttys003
monomac-home:~ joonhwan$ python
Python 3.5.1 (default, Feb 14 2016, 14:34:05)
[GCC 4.2.1 Compatible Apple LLVM 7.0.2 (clang-700.1.81)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> print("hello, python~")
hello, python~
>>>
```

# My First Python Program!

- ❖ 파이썬을 실행하는 근사한 방법 - Jupyter Notebook
  - ❖ 설치: pip install jupyter
  - ❖ 실행: (터미널에서) jupyter notebook



# Jupyter Notebook



# 유용한 자료

- ♦ 참고 교재
  - ♦ 점프 투 파이썬: <https://wikidocs.net/book/1>
  - ♦ CodeCademy: <https://www.codecademy.com/ko/tracks/python-ko>
  - ♦ Coursera: <https://www.coursera.org/learn/interactive-python-1>
  - ♦ Learn Python: <http://www.learnpython.org/>
- ♦ reference
  - ♦ <https://docs.python.org/3/>

# Codecademy

The screenshot shows the Codecademy Python editor interface. The top navigation bar includes the Codecademy logo, a 'Python' dropdown, and user icons. The left sidebar displays a 'Conditions & Control Flow' track and a 'Go With the Flow' section. This section contains text about control flow and a sample Python script. The main workspace shows the script file 'script.py' with the following code:

```
1 def clinic():
2     print "You've just entered the clinic!"
3     print "Do you take the door on the left or the right?"
4     answer = raw_input("Type left or right and hit 'Enter'").lower()
5     if answer == "left" or answer == "l":
6         print "This is the Verbal Abuse Room, you heap of parrot droppings!"
7     elif answer == "right" or answer == "r":
8         print "Of course this is the Argument Room, I've told you that already!"
9     else:
10        print "You didn't pick left or right! Try again."
11    clinic()
12
13 clinic()
```

The bottom of the editor features buttons for 'Save & Submit Code' and 'Reset Code'. Navigation links for 'Q&A Forum' and 'Glossary' are also present.

<https://www.codecademy.com/learn/python>

<https://www.codecademy.com/ko/tracks/python-ko>

# TODOs

- ◆ 파이썬 설치
  - ◆ 자신의 컴퓨터에 파이썬 설치
  - ◆ 강의실 컴퓨터는 실습이 들어가기 전에 설치를 해 놓을 예정
- ◆ 에디터 선택
  - ◆ 맘에 드는 에디터 선택
  - ◆ Command Shell 명령어에 익숙해지기
  - ◆ CMD, Terminal에서 쓰는 도스/유닉스 기본 명령어 찾아 공부하기

# Assignment 1

- ❖ CodeCademy에서 Python 강좌 수강
  - ❖ 강좌 수강 증명 제출 (due: 9/30)
  - ❖ 과제 면제 요청 가능 (파이썬, 프로그래밍 경험자)

<https://tinyurl.com/ydxog5yk>

The screenshot shows the Codecademy website for the Python course. At the top, there's a navigation bar with the Codecademy logo, a 'Upgrade to Pro' button, a 'Catalog' link, a notification bell icon, and a user profile icon. Below the navigation, there's a 'Back to Courses' link. The main title is 'Python', with a subtitle: 'Learn to program in Python, a powerful language used by sites like YouTube and Dropbox.' To the right of the title is a large circular progress indicator. Inside the circle is a pink apple icon on a hexagonal background, with the word 'Lessons' below it and a progress bar at the bottom indicating '28%'. Below the progress circle is a green 'CONTINUE' button. At the bottom of the page, there's a teal footer bar with text: 'Want more practice and review? Upgrade for the complete experience.', '8 Projects', '9 Quizzes', '1 Final Project', and a 'Get Instant Access »' button. The 'hcilab.' logo is located on the far left edge of the slide.

# Questions?

---