

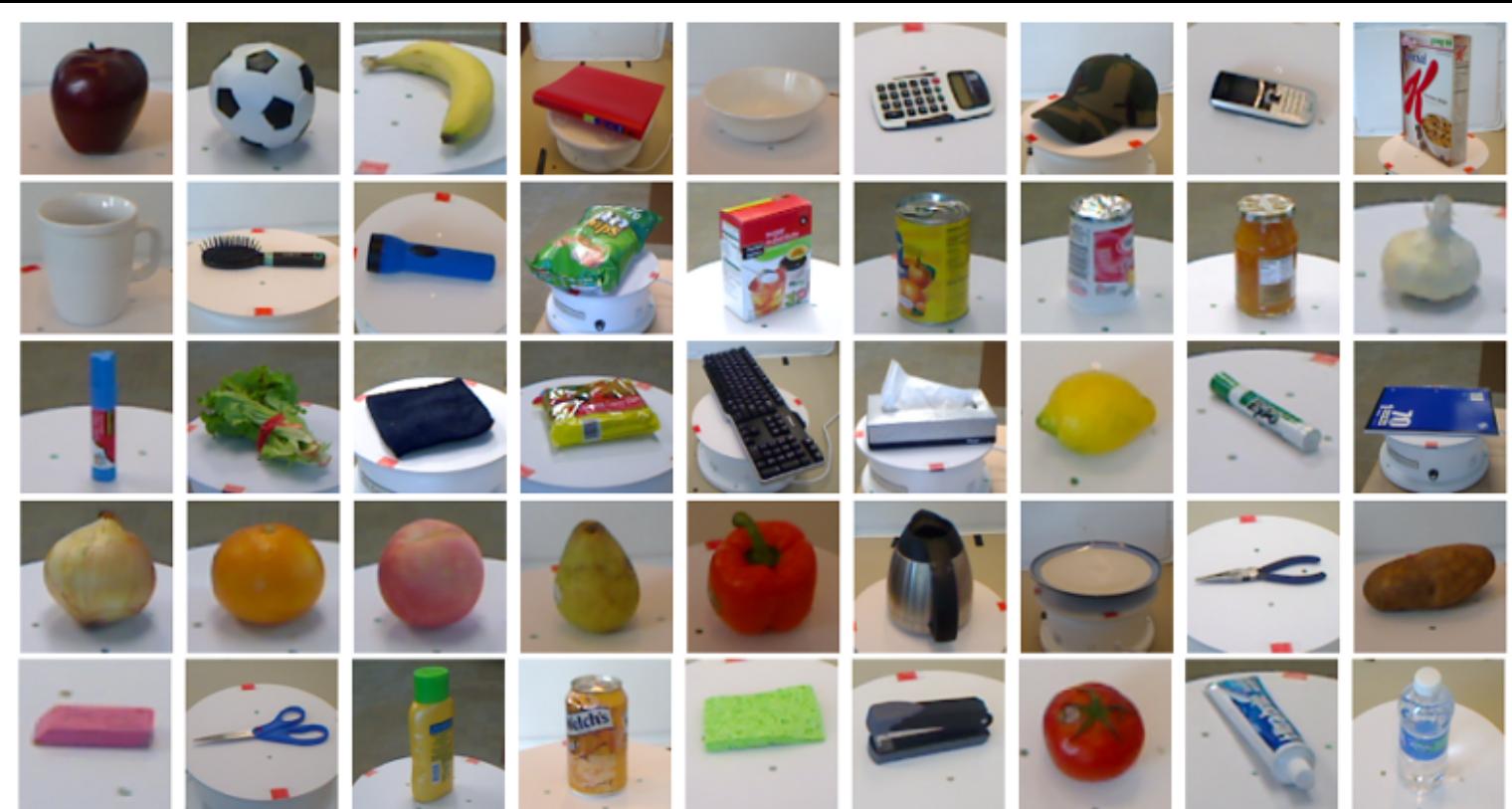
# What is IT & Cognition?

What is Scientific Programming?





# Texts

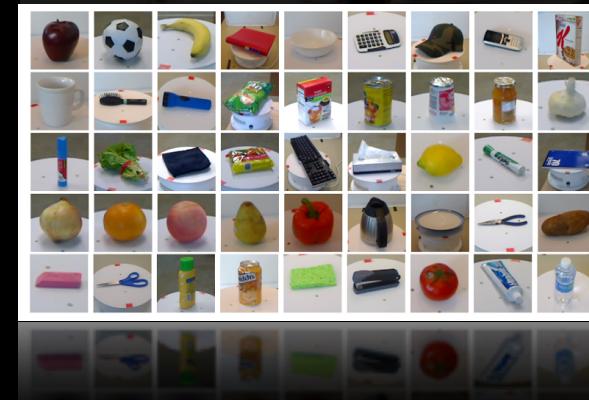
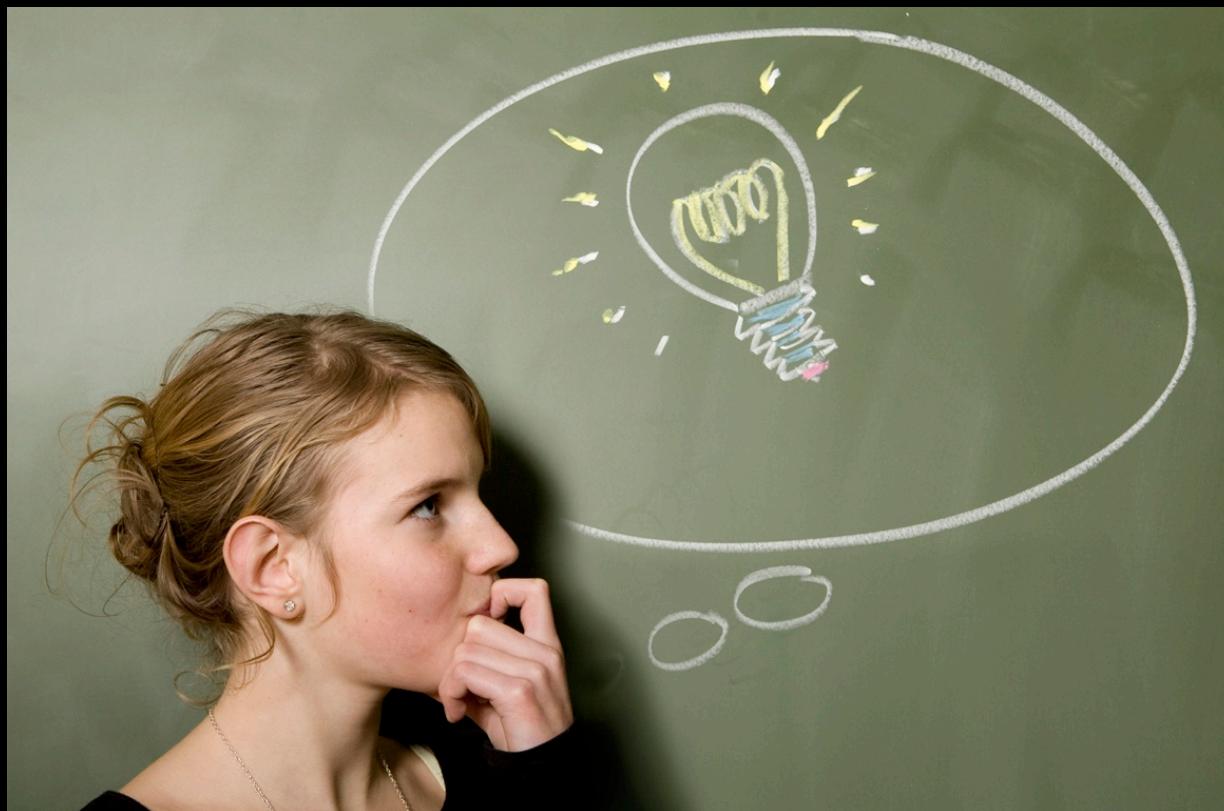


# Images



Both

In two years you will be an expert  
in automatic extraction of  
information from texts and images...



Perspective #3



PLATINUM LEVEL SPONSOR



GOLD LEVEL SPONSORS



Microsoft:



SILVER LEVEL SPONSORS



ممىز قطر لبحوث الحوسبة  
Qatar Computing Research Institute  
Member of Qatar Foundation



Xerox Research Centre Europe



BEST STUDENT PAPER AWARD

**IBM Research**

STUDENT VOLUNTEER

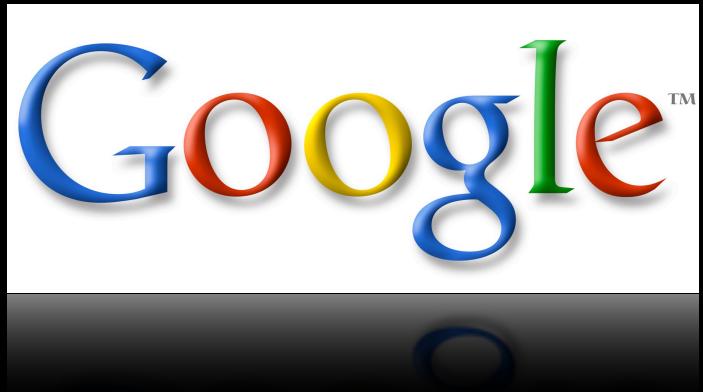
**facebook**

CONFERENCE BAG SPONSOR



CONFERENCE BAG SPONSOR

# Companies (ACL'13)



meltwater  
group  
@Lonb

POLFOTO



textkernel

The textkernel logo consists of the brand name in a large, bold, teal sans-serif font. Below it is a horizontal bar with a dark gradient, and a reflection of the text is visible underneath.

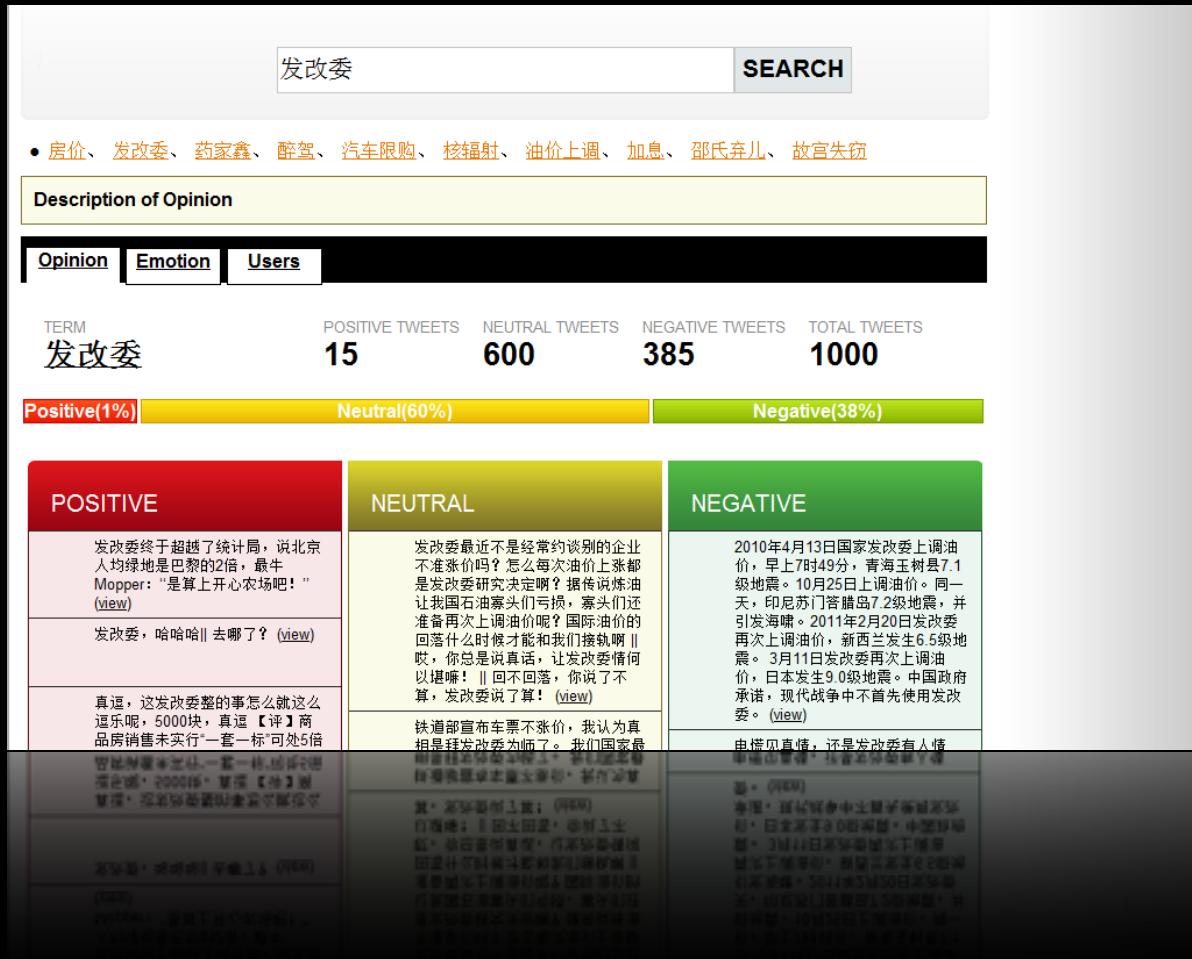
INFOMEDIA

IT & Cognition Advisory Board

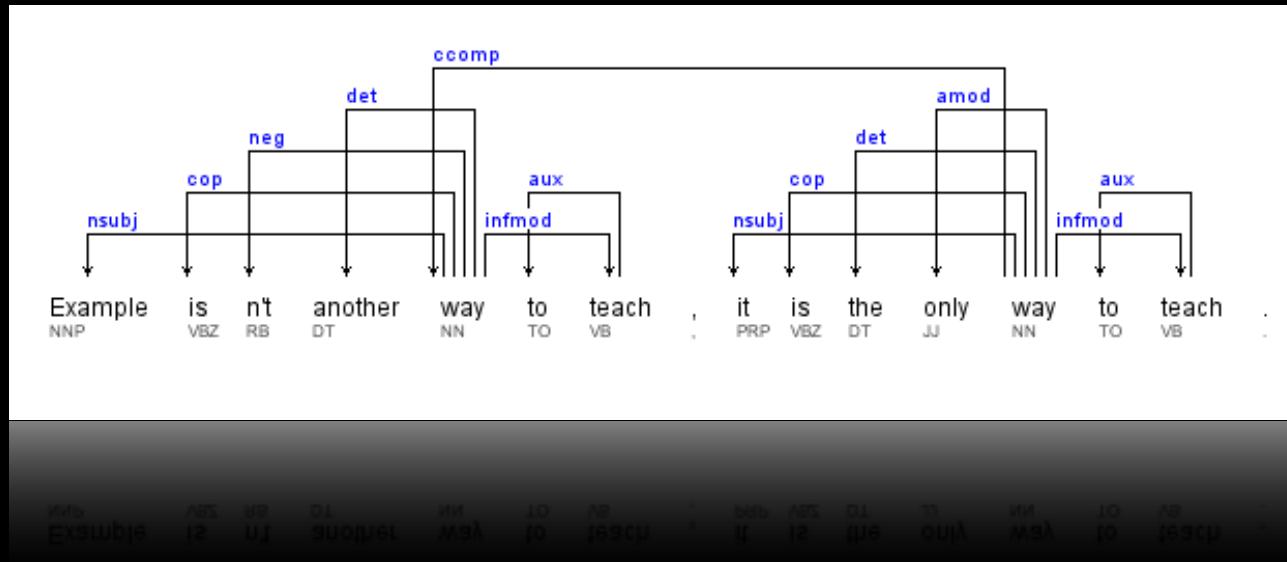
# Learning goals

- Manage large collections of texts and images
- Feature extraction
- Data analysis and visualization
- Model induction from data (machine learning)
- Scientific evaluation and statistical analysis
- Scalability
- Web crawling and data collection from Twitter, Facebook, etc.
- **Creative cognitive technologies**

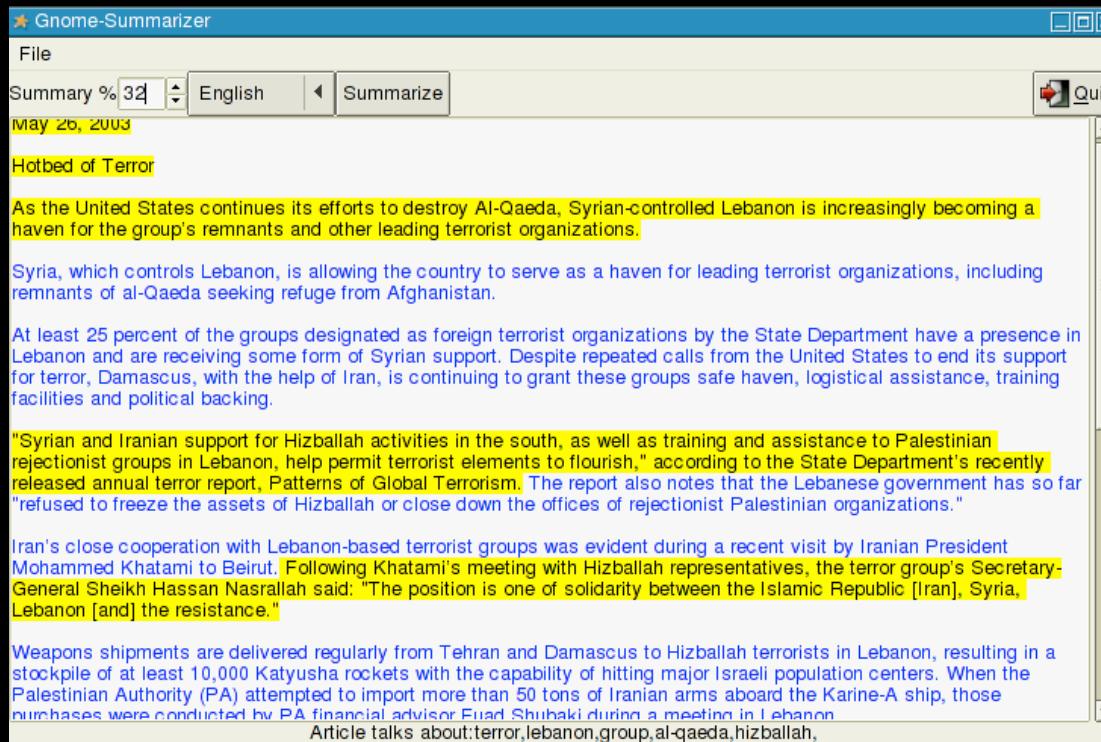
# Text processing - Preview



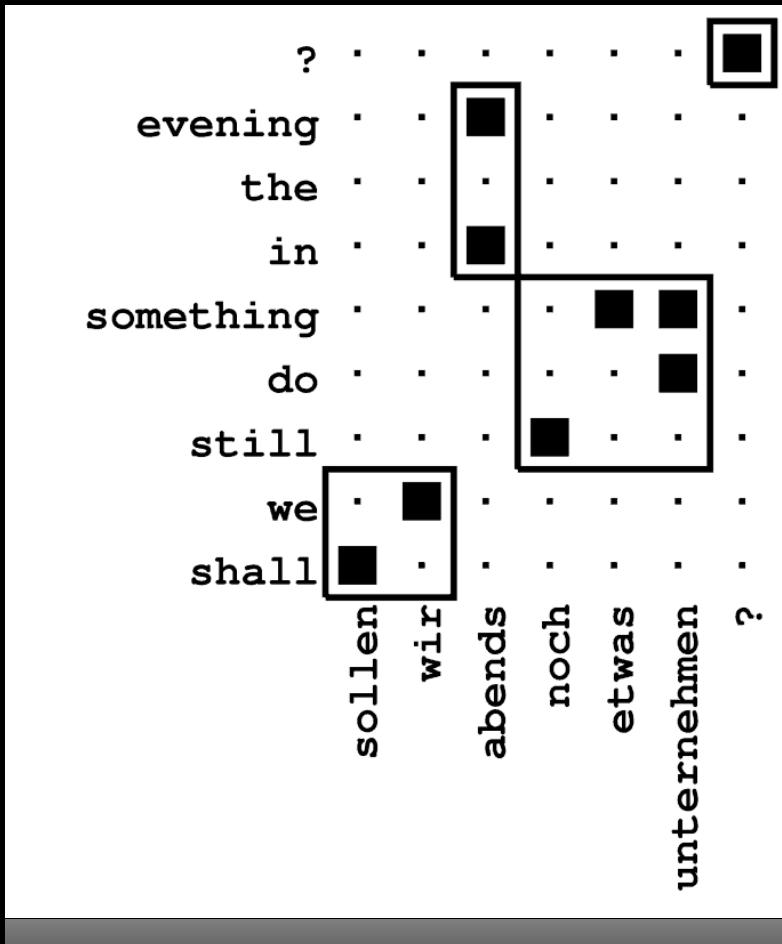
# Sentiment analysis



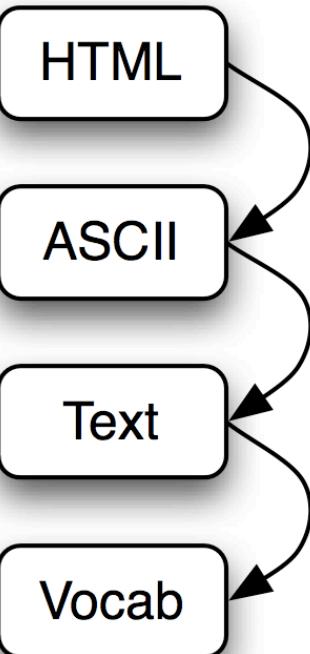
# Syntactic parsing



# Summarization



# Translation



```
html = urlopen(url).read()  
raw = nltk.clean_html(html)  
raw = raw[750:23506]  
  
tokens = nltk.wordpunct_tokenize(raw)  
tokens = tokens[20:1834]  
text = nltk.Text(tokens)  
  
words = [w.lower() for w in text]  
vocab = sorted(set(words))
```

Download web page,  
strip HTML if necessary,  
trim to desired content

Tokenize the text,  
select tokens of interest,  
create an NLTK text

Normalize the words,  
build the vocabulary

```
vocab = sorted(set(words))  
vocab = [w for w in text]
```

Build the wordlist  
from the vocabulary

# NLTK - website wordlist

# Image processing - Preview

Writing an array to a file:

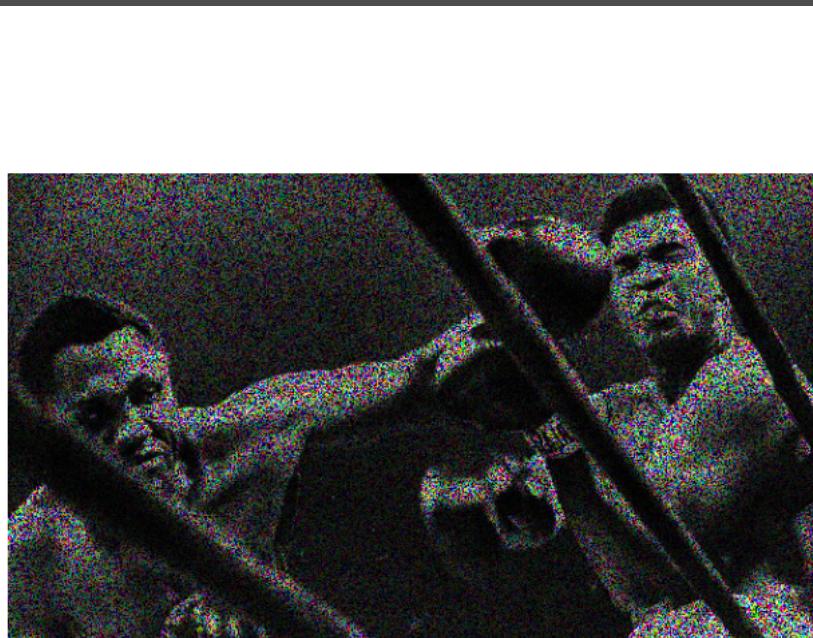
```
from scipy import misc
l = misc.lena()
misc.imsave('lena.png', l) # uses the Image module (PIL)

import matplotlib.pyplot as plt
plt.imshow(l)
plt.show()
```

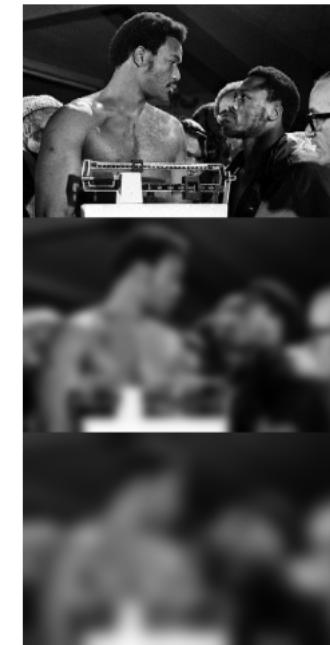


# imread, imshow

# Simple image processing



Add noise to images

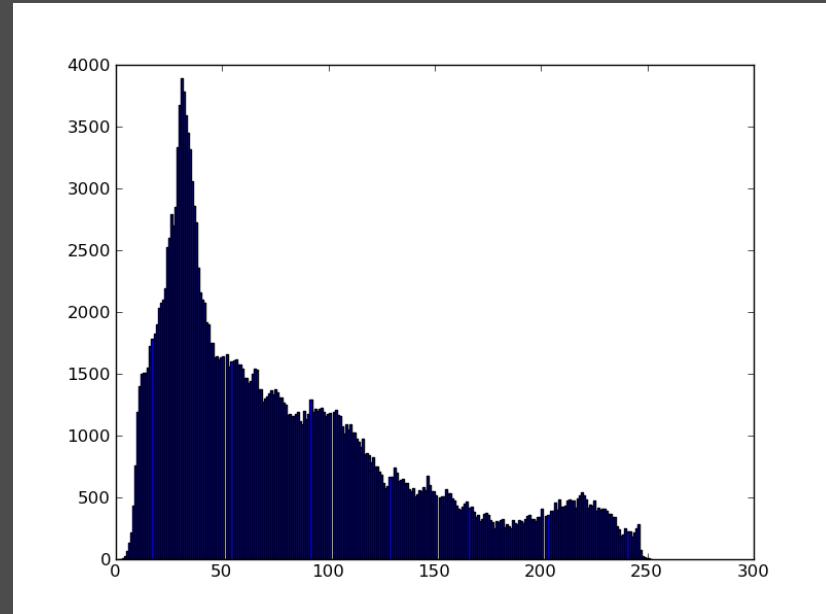


Blur images

# Simple image processing



Merge images



Extract image statistics

Thresholding  
Otsu's Method

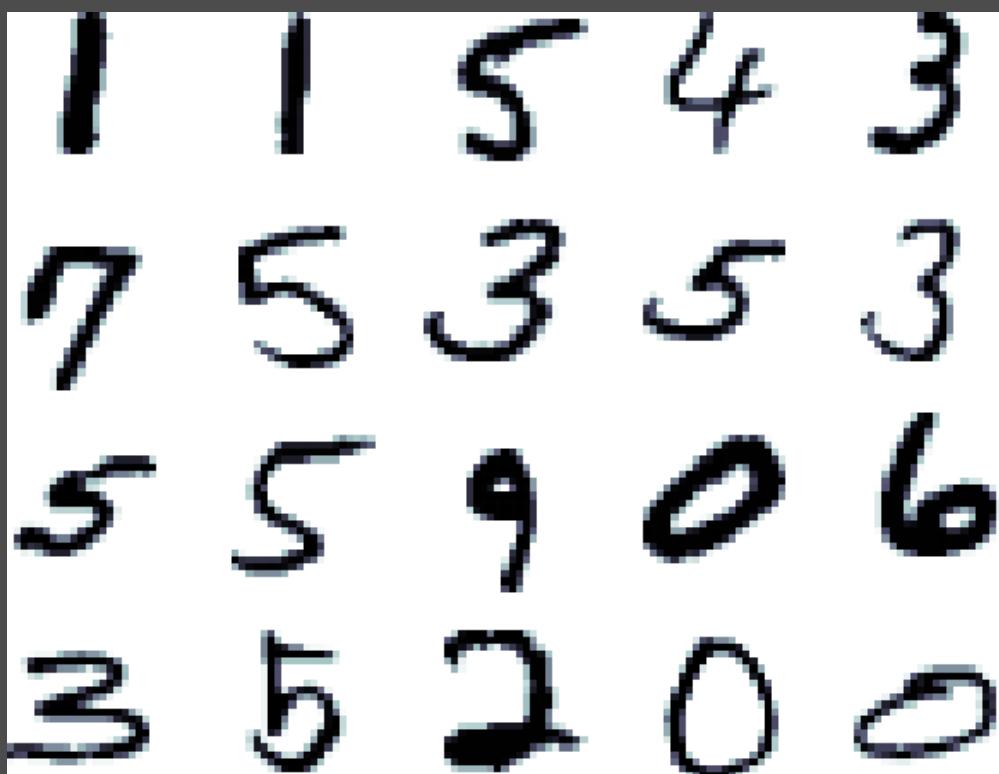


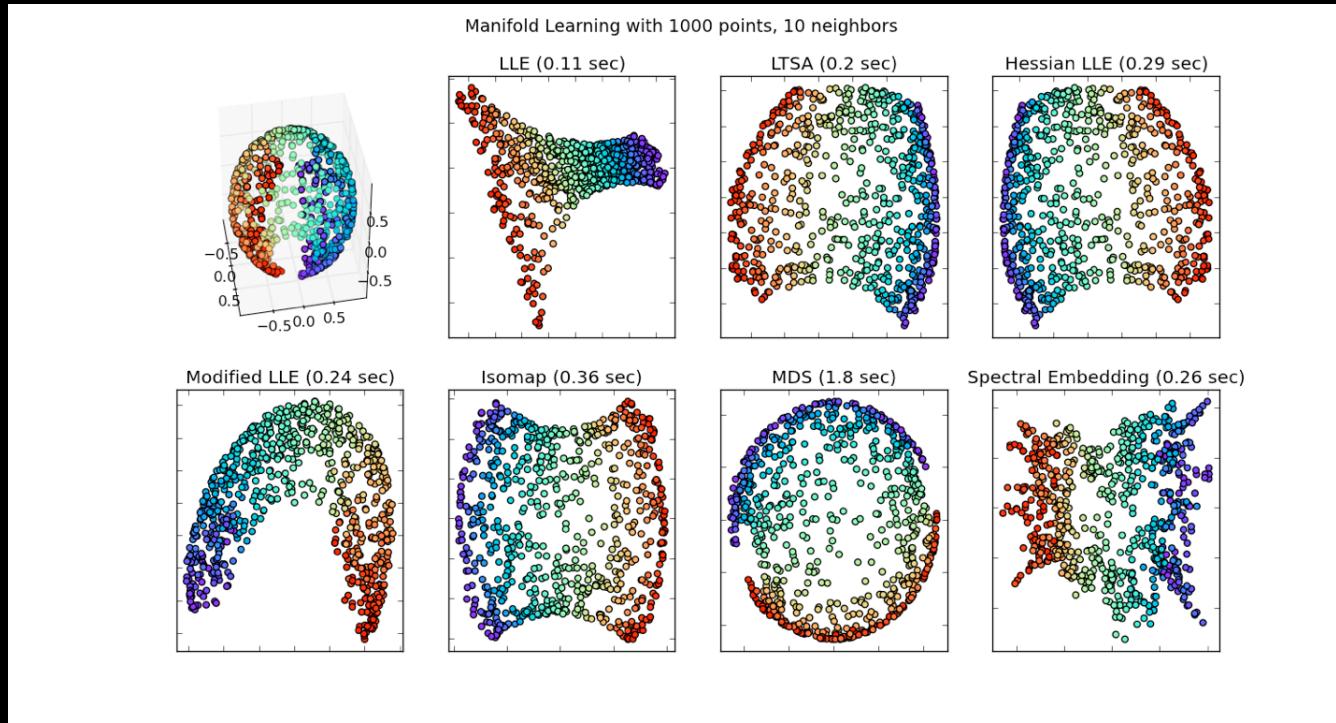
Hm...

Swirling



More seriously  
Digit recognition





# Visualization

# Crash courses

**Today and tomorrow**

Python Programming

**Thursday**

Basic calculus: Vectors, matrices,  
optimization



# Today and tomorrow

- Running a Python script
- Loading and printing a text
- Loading and printing an image
- Your first meaningful program
- Data types
- The building blocks of simple Python programs
- Modules

# Literature

- Reading material
  - [http://upload.wikimedia.org/wikipedia/commons/9/91/  
Python\\_Programming.pdf](http://upload.wikimedia.org/wikipedia/commons/9/91/Python_Programming.pdf)
  - [http://programmingcomputervision.com/downloads/  
ProgrammingComputerVision\\_CCdraft.pdf](http://programmingcomputervision.com/downloads/ProgrammingComputerVision_CCdraft.pdf)
- References: <http://docs.python.org/2/tutorial/> and [http://scikit-learn.org/  
stable/](http://scikit-learn.org/stable/)
- Video tutorials
  - <https://developers.google.com/edu/python/>
  - [http://marakana.com/s/post/1090/2012\\_pydata\\_workshop](http://marakana.com/s/post/1090/2012_pydata_workshop)

# Why Python?

- Python is an open-source object-oriented programming language with a supportive community.
- Its main advantage is that it is easy to learn and read; i.e., it reads much like pseudo-code.
- It is very suitable as a first programming language,
- but it is also suitable as your last programming language.
- It is extremely portable and works on Mac, Windows, Unix, Linux, PalmOS, PlayStation, etc.
- Compilation is implicit.
- Memory management is automatic.

**Note:** Compared to Java code is up to five times shorter. There is dynamic typing, which means quicker development, and Python uses less memory (a major issue on laptops).



# Coding philosophy

Documentation is at your own  
responsibility.  
Speed is nice, but secondary.

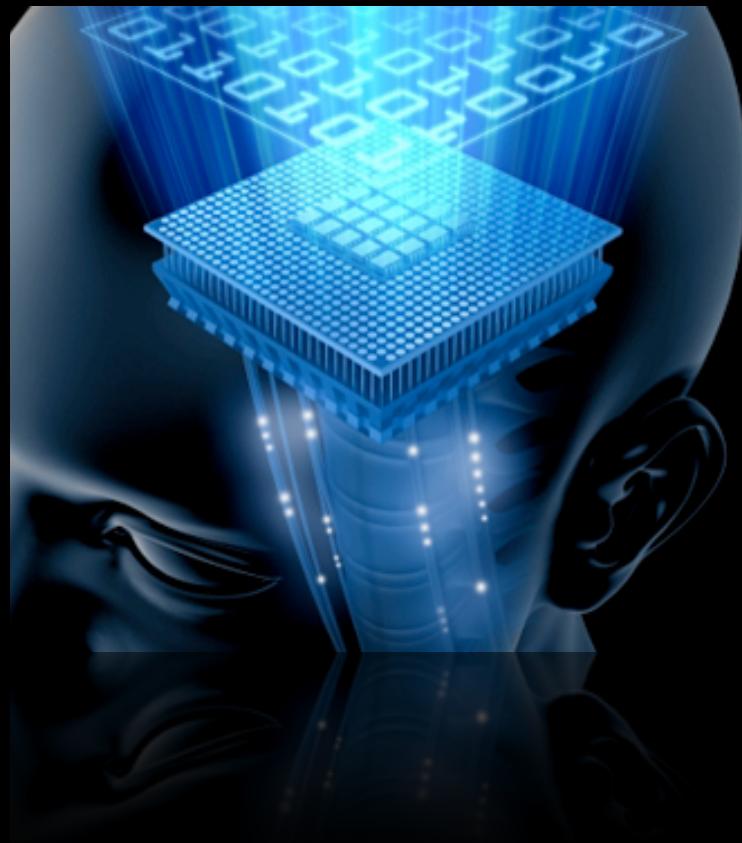


# Show cases

[http://www.youtube.com/watch?  
v=bh9\\_uOdz-bU](http://www.youtube.com/watch?v=bh9_uOdz-bU)

[http://www.youtube.com/watch?  
v=b58RpVvsfpM](http://www.youtube.com/watch?v=b58RpVvsfpM)

[http://www.sentiment140.com/  
search?query=snowden&hl=en](http://www.sentiment140.com/search?query=snowden&hl=en)



# Today and tomorrow

- Running a Python script
- Loading and printing a text
- Loading and printing an image
- Your first meaningful program
- Data types
- The building blocks of simple Python programs
- Modules

## Interactive mode, executables, and libraries

- For interactive mode, type **python**
- To run a script, type **python script.py [args]**
- Or make it *executable*:
  - **chmod +x script.py**
  - **#!/usr/bin/python**
  - **/script.py**
- Scripts/modules:
  - **if \_\_name\_\_ == '\_\_main\_\_':**
  - **main()**

## Check list for Day 1

- ↗ open, readlines, strip, sys.argv (module: sys)
- ↗ str, int, float, list, dict (Try to use '+' and 'len' with different types)
- ↗ slices, .append(x), .insert(i,x), .count(x), .reverse(), max(), sum(),
- ↗ for, in, range, if, ==, else, Booleans (not, or, and),
- ↗ dictionaries (.iterkeys(),.itervalues(),.iteritems())
- ↗ <, >, +, -, \*, /, \*\*, math.log, math.sqrt (module: math)
- ↗ def, return