Class 01: Getting started

August 28, 2017

Why are you taking this course?

Either:

- You don't know programming but are eager to learn, or
- It's a requirement for your degree

Good news!

- Programming is fun
- Programming improves all aspects of human experience
- Programming will make your life easier

More good news!

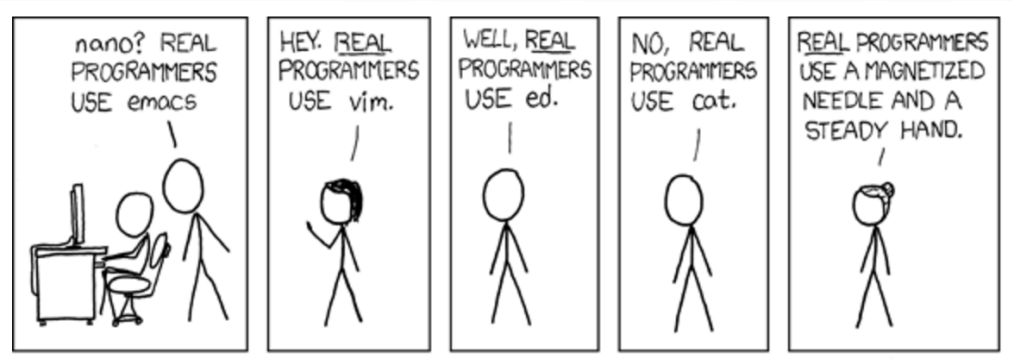
- All the examples in this course are based on linguistic problems

Prerequisites

Stuff you need before you begin:

- A UNIX-compatible system (GNU/Linux, *BSD, Mac/OS)
- A text editor
- An installation of Python – Python 3.0 or higher!

How to choose a text editor:



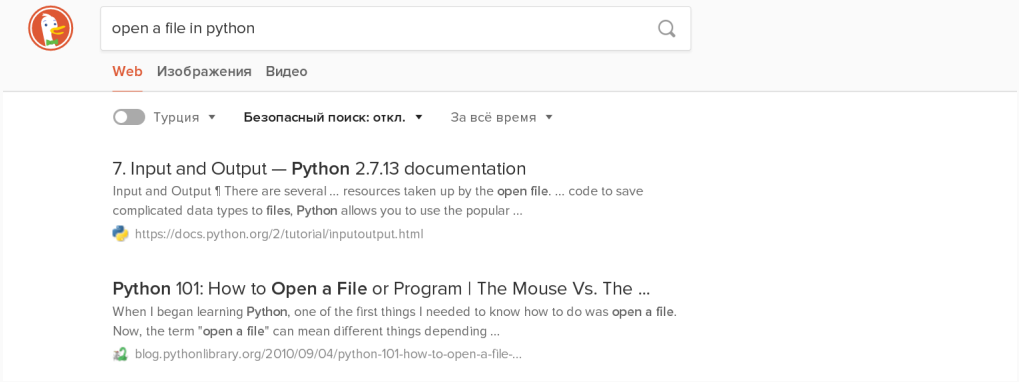
Honestly, use something other people (programmers) you know use.

Argh but what if I have Windows™

I have no idea about Windows

To be safe, install a Virtual Machine (e.g. VirtualBox) and a flavour of GNU/Linux, e.g. Ubuntu.

How to get help



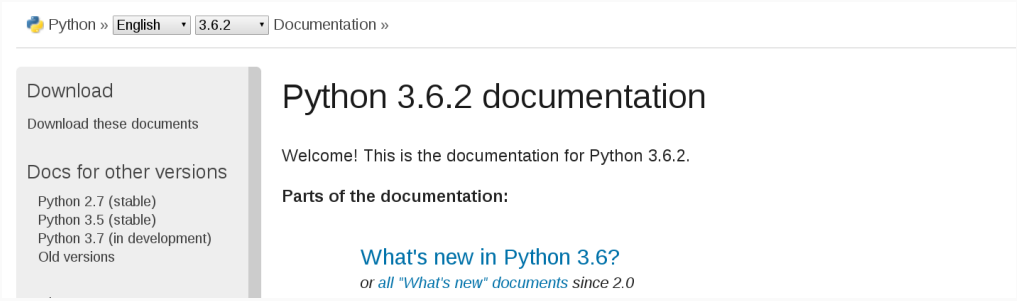
On your own:

- A search engine such as Google™, Yandex™ or DuckDuckGo™
- The fine Python documentation: `http://docs.python.org`
- Internet Relay Chat: `http://webchat.freenode.net`
- Stack Overflow: `https://stackoverflow.com`

Ask me:

In class	(IRL)
`#hseling` on `irc.freenode.net`	(IRC)
`https://vk.com/id138461818`	(VK)
`francis.tyers@gmail.com`	(Hangouts)

How to get help



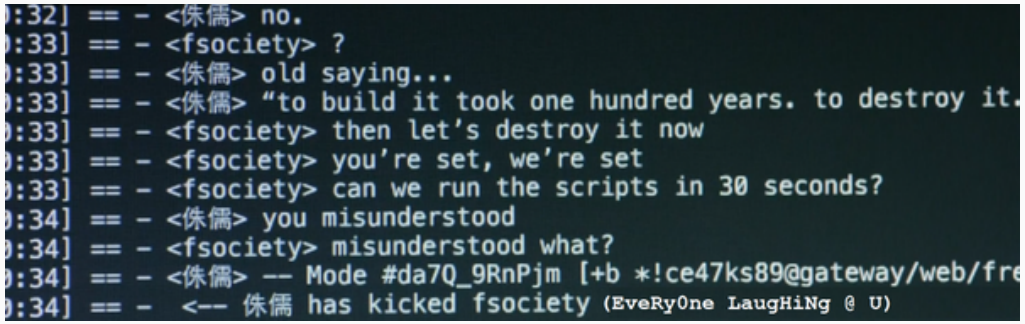
On your own:

- A search engine such as Google™, Yandex™ or DuckDuckGo™
- The fine Python documentation: `http://docs.python.org`
- Internet Relay Chat: `http://webchat.freenode.net`
- Stack Overflow: `https://stackoverflow.com`

Ask me:

In class	(IRL)
`#hseling` on `irc.freenode.net`	(IRC)
`https://vk.com/id138461818`	(VK)
`francis.tyers@gmail.com`	(Hangouts)

How to get help



On your own:

- A search engine such as Google™, Yandex™ or DuckDuckGo™
- The fine Python documentation: `http://docs.python.org`
- Internet Relay Chat: `http://webchat.freenode.net`
- Stack Overflow: `https://stackoverflow.com`

Ask me:	In class	(IRL)
	`#hseling` on `irc.freenode.net`	(IRC)
	`https://vk.com/id138461818`	(VK)
	`francis.tyers@gmail.com`	(Hangouts)

How to get help

On your own:

- A search engine such as Google™, Yandex™ or DuckDuckGo™
- The fine Python documentation: `http://docs.python.org`
- Internet Relay Chat: `http://webchat.freenode.net`
- Stack Overflow: `https://stackoverflow.com`

Ask me:

In class	(IRL)
`#hseling` on `irc.freenode.net`	(IRC)
`https://vk.com/id138461818`	(VK)
`francis.tyers@gmail.com`	(Hangouts)

Structure of the course

`https://ftyers.github.io/079-osnov-programm/index.html`

Class	Topic	Class	Topic
1	Command line	7	*Project work*
2	Segmenter	8	*Project work*
3	Tokeniser	9	*Project work*
4	Transliterator	10	*Project work*
5	Language model	11	*Project work*
6	Tagger	12	*Presentations*

Pipeline

A typical basic NLP pipeline looks like the following:

```
sentence segmenter | tokeniser | tagger | parser
```

- segmenter: takes a paragraph and gives sentences
- tokeniser: takes a sentence and gives list of tokens
- tagger: gives every token a morphosyntactic tag
- parser: takes a tagged sentence and gives a parse tree

During the first six classes you will be implementing basic versions of the first three modules.

Projects

For the remaining six classes you will work on:

- A small software project
- Something that you are excited about

For inspiration, you could:

- Perform some quantitative linguistic experiment
- Implement a program to convert between formats
- Write a *scraper* for some online language data
- Implement a simple machine learning solution to a problem

You will need to decide by the 5th class, if you are unsure, talk to me

Marking scheme

Details on the course page.

Marking

- 40% Project
- 10% Exam
- 20% Practicals
- 20% Homework
- 10% Active participation

Project: The project will encompass all of the class work and homework for the last six classes. You should start thinking from the first class what you might be interested in working on. If you cannot come up with any ideas, then I will give a number of options, or come and talk to me. The project should be substantial and test and expand your knowledge in some way. It should contain an evaluation component, either for efficiency of implementation or in terms of accuracy for some task. It will include a short (maximum ten minute) presentation to be done on the last day of class. One of the most important aspects of programming is learning to use the computer to *scratch an itch* `удовлетворить личное желание' the project will ensure you are able to do that.

Exam: There will be a short exam to test you on what you have learnt in the course. There will be a number of multiple choice questions and a programming assignment to complete on paper. As programming is a subject that is more suited to practical evaluation the exam has a concomitantly low contribution to your final mark.

Practicals: Most of the course will be made up of practical sessions. I will evaluate your progress after each session.

Homework: Homework will be submitted through Github, and will need to be completed before the following lesson. Your Github repository should be called `2017-osnov_programm` and have the following subdirectories: `corpus` for your (sub-)corpus from Wikipedia, and `project` for your project work.

Active participation: Beyond simply showing up, I encourage you to contribute to discussions by asking questions, answering questions, making relevant comments, helping classmates and asking for help with in-class activities, etc. There are no stupid questions — I want to make sure everyone grasps the concepts, and many are not as straightforward as they may first seem (or as I think they are). You are also expected to have read any assigned readings before class.

tl;dr Most of the final mark is from the class work and project.

What we are going to do today

First things first:

- Make sure you have Python installed
- Set up Github accounts
- Install a text editor
- Work with the shell

Then second things:

- Choose a language
  - For purposes of speed, choose one with $<= 500,000$ articles
- Download the Wikipedia in that language
- Extract the text from Wikipedia

Check your Python installation

Open a terminal and type `python3` and press return ⏎ .

```
$ python3
Python 3.5.2+ (default, Aug  5 2016, 08:07:14)
[GCC 6.1.1 20160724] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

If you don't have Python installed, install it now.

Github

All practical work will be stored and submitted through GitHub.

If you don't already have an account:

- Go to `https://github.com/join`
- Fill in the information
- Click "Create an account"
- Choose "Unlimited public repositories for free."
- Skip the next part.

Setup the directory structure

In your browser:

- First make a repository, call it `2017-osnov-programm`
- Choose 'Initialise this repository with a README'
- Click 'Clone or download' and copy the link

In the terminal:

```
$ git clone https://github.com/XXXXXX/2017-osnov-programm.git

$ cd 2017-osnov-programm

$ mkdir corpus project
```

Where XXXXXX is your GitHub username.

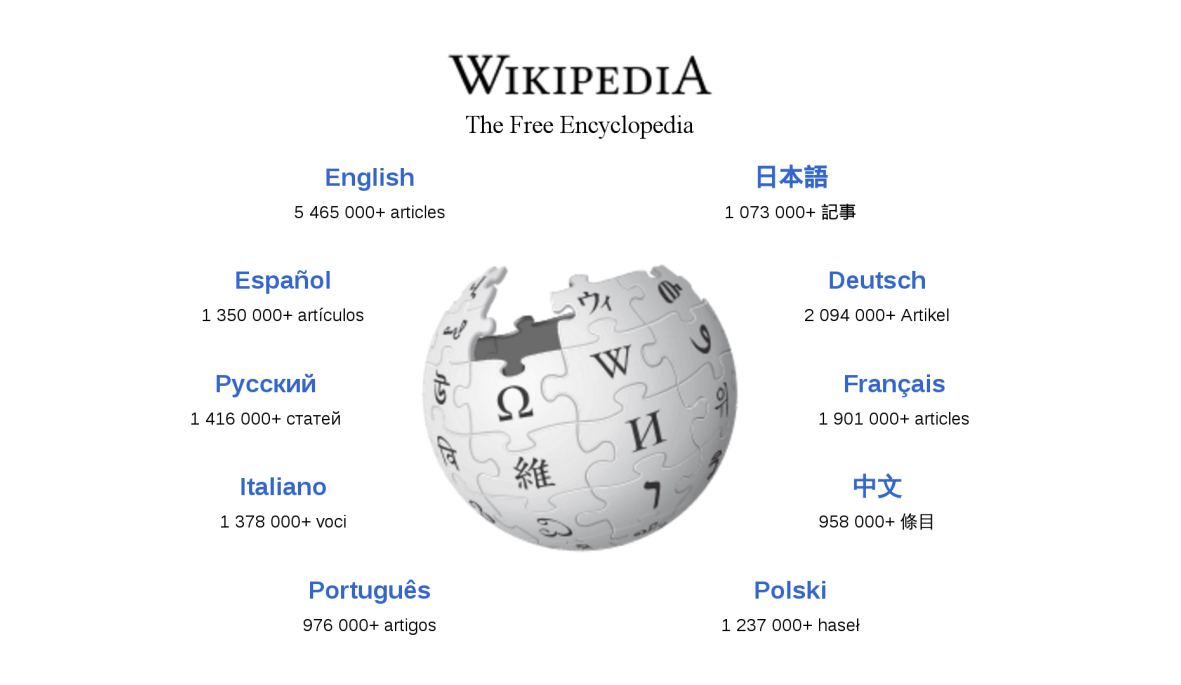
Text editor

I tried to get a definitive answer on which is the best text editor by asking your fellow students I know which one they use ...

- Sublime: +++
- TextWrangler: +
- Vim: +
- Atom: +
- Notepadpp: +
- Emacs:

Unfortunately there were nearly as many favourites as students ...

Wikipedia as a corpus/1



Wikipedia makes a great[1] corpus:

- Free to use and distribute
- Very many languages – 295 at the last count

[1] Well, great in some respects

Wikipedia as a corpus/2

Deliberately vague steps:

- Use your search engine to find where Wikipedia keeps it's 'dumps'.
- Find the language code of the language you are interested in
- Download the dump for the language you are interested in
  - Tip 1: You're looking for a 'Database backup dump'
  - Tip 2: The filename will include `pages-articles.xml.bz2`
- Find WikiExtractor on the Apertium Wiki
- Run WikiExtractor on the dump file you downloaded.

What next

There is an excellent introduction to the shell by Ken Church:

For the remainder of the class we'll be going through this and making sure that you are able to run all of the commands.

Notes:

- We'll use our Wikipedia dump, not Genesis
- Instead of `tr -sc` we'll use `tr -s`
- Instead of `[a-z][A-Z]` we'll use `[,;:!?/." ()]`

For example for Avar:

```
tr -s '[,;:!?/." ()]' '[\n*]' < wiki.txt |
sort | grep '[а-яА-Я]' |
uniq -c > wiki.hist
```

You can get the same output in many different ways:

```
$ head -n 5 wiki.hist
2 100% магIарулал
1 10-го
1 11-абилеб
1 1250 км
1 1369-леб

$ sed 5q wiki.hist
2 100% магIарулал
1 10-го
1 11-абилеб
1 1250 км
1 1369-леб
```