



CPOUTA ENVIRONMENT IN GEOIFORMATICS TEACHING

DIGITAL GEOGRAPHY LAB
UNIVERSITY OF HELSINKI





CONTENTS

cPouta use cases from geoinformatics courses at the University of Helsinki:

- Open data course: **GeoServer**
- Advanced GIS course: **PostGIS**
- Python course: **Jupyter Lab**



GEOSERVER

Challenge:

- Teach the basics of **Geoserver** and sharing geospatial data.
- Use shared datasets in students' custom web maps

The screenshot displays the GeoServer web interface. The top navigation bar includes the GeoServer logo and the text "GeoServer". The main content area is titled "Welcome" and contains the following information:

- Server Status: 80 Layers (Add layers)
- GeoServer Logs: 74 Stores (Add stores)
- Contact Information: 63 Workspaces (Create workspaces)
- About GeoServer: Strong cryptography available

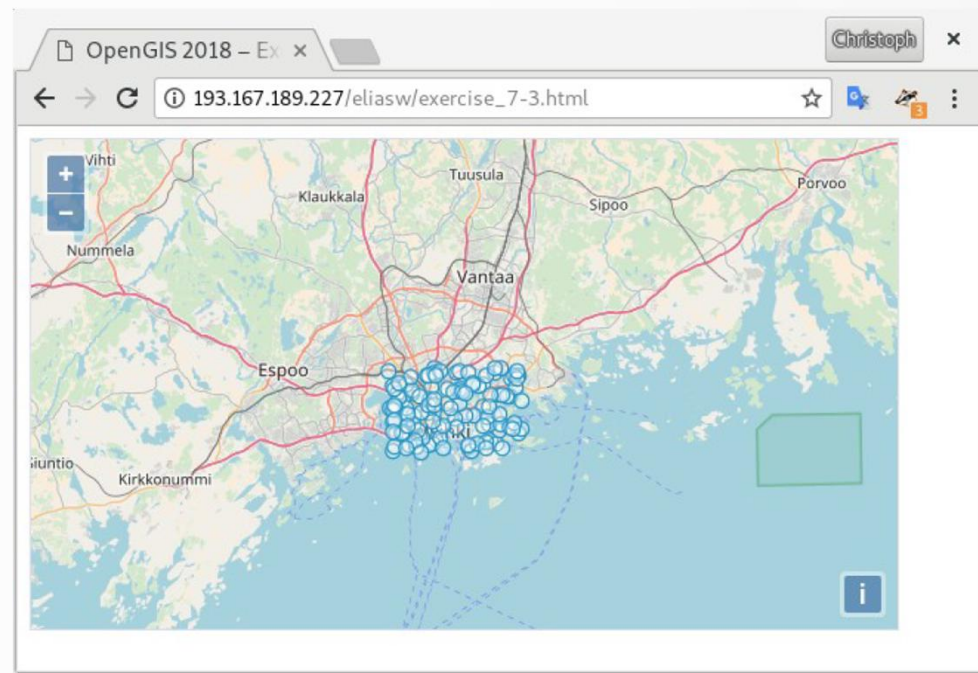
At the bottom, it states: "This GeoServer instance is running version 2.12.1. For more information, see the administrator's manual."



GEOSERVER

Solution

- Remote CSC server to run course Geoserver
- All students used the same account
- Datasets were imported using QGIS GeoServer Explorer plugin
- Shared datasets were used in custom OpenLayers web maps

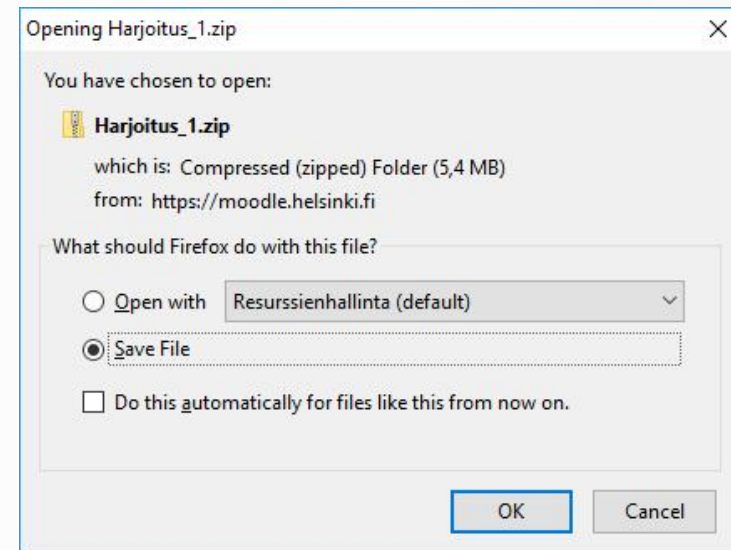




POSTGIS

Challenge:

- Teach the basics of **PostgreSQL/PostGIS**
- Distribute exercise data in a smart way



In stead of this...



POSTGIS

Challenge:

- Teach the basics of **PostgreSQL/PostGIS**
- Distribute exercise data in a smart way

Solution: PostGIS (running on CSC server) used via QGIS



Create a New PostGIS connection

Connection Information

Name: socialmedia
Service:
Host: 193.166.25.204
Port: 5432
Database: postgres
SSL mode: prefer

Authentication: **Configurations**

Username: Save
Password: Save

Test Connection

Only show layers in the layer registries
 Don't resolve type of unrestricted columns (GEOMETRY)
 Only look in the 'public' schema
 Also list tables with no geometry
 Use estimated table metadata

OK Cancel Help



SETTING THINGS UP

- **Collaboration** between the University and CSC:
- **CSC**
 - Provides cPouta infrastructure
 - Gives advice about installations. **Instructions** to setup popular server software (GeoServer, PostGIS) in CSC's cPouta environment are available [on GitHub](#).
- **University** (the teacher and/or local IT support):
 - **Sets up the virtual machine**
 - **Installs the required software** (eg. Geoserver or PostGIS)
 - Creates users accounts for students
 - **Monitors** the virtual machines and software usage





AUTOMATING GIS PROCESSES

Challenge:

- Organizing interactive **python teaching** for 80 + students!
- Geospatial Python libraries





AUTOMATING GIS PROCESSES

Challenge:

- Organizing interactive **python teaching** for 80 + students!
- Geospatial python libraries

Solution: JupyterLab & notebook (docker) hosted by CSC

State	Name	Time Left	Access	Actions	Details
✓	pb-clifford-the-witty	5h 59m	▶ Open in browser	✕ Destroy	Details

Notebooks by CSC Dashboard

2018

COURSE INFORMATION

- General info
- Learning goals
- Grading
- Installing Python
- Vocabulary - Basic terms
- Resources
- License and terms of usage

LESSON 1

- Motivation for the course
- Lesson overview
- Course environment
- Communicating with Slack
- A taste of Python
- Exercise 1

LESSON 2

- Lesson overview
- Introduction to Version Control and GitHub
- Meet Git
- Using Classroom for Github
- Basic elements of Python
- Exercise 2

LESSON 3

Other Versions v: master

View page source

This page was generated from source/notebooks/L4/modules.ipynb.

[launch full binder](#) [launch student binder](#) [launch CSC notebook](#)

Loading and using modules

What is a module?

A *module* in Python is simply a Python `.py` file that contains a list of related functions that can be loaded and used. Modules are similar to what are more generally called libraries in programming languages, which again contain code related to a specific task such as mathematical operations. There are a *HUGE* number of Python modules, and many of them greatly extend what can be done in a normal Python program. In fact, the abundance of free Python modules is one of the best reasons to learn and start using Python.

How can modules be loaded?

Python modules can be loaded in a number of different ways.

Loading a module

Let's start simple with the math module. Here, we'll load the math module using the `import` statement.

```
In [1]: import math
In [2]: math.sqrt(81)
Out[2]: 9.0
```

Here we have loaded the math module by typing `import math`, which tells Python to read in the functions in the math module and make them available for use. In our example, we see that we can use a function within the math library by typing the name of the module first, a period, and then the name of function we would like to use afterward (e.g., `math.sqrt()`). Built-in functions such as `print()` do not require the name of the module first since nothing is explicitly imported.

Static webpage at: <https://geo-python.github.io/>

The screenshot shows a Jupyter Notebook interface with a sidebar on the left containing a file explorer and a list of notebooks. The main area displays the notebook content for 'modules.ipynb'. The content includes a title 'Loading and using modules', a section 'What is a module?', and a code cell with the text 'import m'. A large red 'DEMO' watermark is overlaid on the notebook content.

DEMO

- Used during lessons and exercises
- Lesson materials are readily available
- Students clone exercise repo from GitHub

Interactive version of the lesson at: <https://notebooks.csc.fi/>



KIITOS!

helsinki.fi/digital-geography

geo-python.github.io