

lecture 0 - Introduction



Programming in geoinformatics
Autumn 2015

VISION - “Not everyone can be a programmer” ?

strategies. An experiment found two distinct populations of students: one could build and consistently apply a mental model of program execution; the other appeared either unable to build a model or to apply one consistently. The first group performed very much better in their end-of-course examination than the second in terms of success or failure. The test does not vary

Despite the enormous changes which have taken place since electronic computing was invented in the 1950s, some things remain stubbornly the same. In particular, most people can't learn to program: between 30% and 60% of every university computer science department's intake fail the first programming course.

Maybe this is anecdotal, but when I taught intro programming to a few hundred liberal arts students, I found no such "double hump". It seemed to me they were all quite capable, though some worked harder than others, and a very few tried to bluff their way through.

A lot has to do with how it is taught.

<http://www.makeuseof.com/tag/6-signs-meant-programmer/> - Final Thoughts

<http://programmers.stackexchange.com/questions/163631/has-not-everyone-can-be-a-programmer-been-studied>

VISION - objectives

IT - a place for everyone?

outside of IT:

- science
- journalism
- data analyses
- services
- mechanical engineering
- ...

programming in geoinformatics - a basic course?

PROGRAMMING LANGUAGES (IN GEOINFORMATICS)

- why python?
- “Is it true that once you learn one language most of the rest come easy?”
(<http://programmers.stackexchange.com/questions/108578/is-it-true-that-once-you-learn-one-language-most-of-the-rest-come-easy>)
- what is next?

PROGRAMMING LANGUAGES IN GEOINFORMATICS

C / C++

extremely fast

low-level

most of software written in C++ (ArcGIS, QGIS)

big desktop applications

not so easy for “everyday-use” problems



PROGRAMMING LANGUAGES IN GEOINFORMATICS

Java / C#

C# - “Microsoft version of Java”

mostly desktop application, servers, databases

Java - Android

JTS, GeoServer



GeoServer is an open source server

Designed for interoperability, it publishes data from a



OpenStreetMap Data © OpenStreetMap contributors, ODbL

The image shows a screenshot of the GeoServer interface. At the top, there is a logo for GeoServer, which consists of a stylized globe icon followed by the text "GeoServer". Below the logo, there is a blue horizontal bar. The main content area features a map of Montreal, Quebec, Canada. The map is overlaid with a network of red and blue lines representing traffic flow or travel times. Various travel time labels are visible on the map, such as "15:40", "15:20", "10:15:20", "720", "10", and "134". The map also shows major roads and landmarks, including the St. Lawrence River and the city of Montreal. The text "GeoServer is an open source server" and "Designed for interoperability, it publishes data from a" is displayed above the map. At the bottom of the map, there is a small text attribution: "OpenStreetMap Data © OpenStreetMap contributors, ODbL".

PROGRAMMING LANGUAGES IN GEOINFORMATICS

Javascript

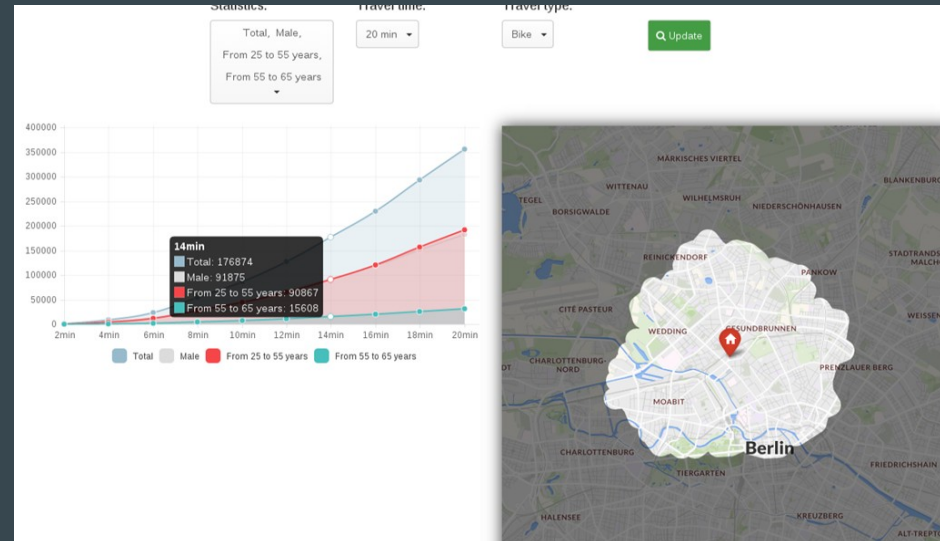
primary client-side web applications

server side (node.js, io.js), desktop, mobile (phonegap, react.native), ...

cartography - visualization and sharing

- leaflet.js
- turf.js
- openlayers
- google maps

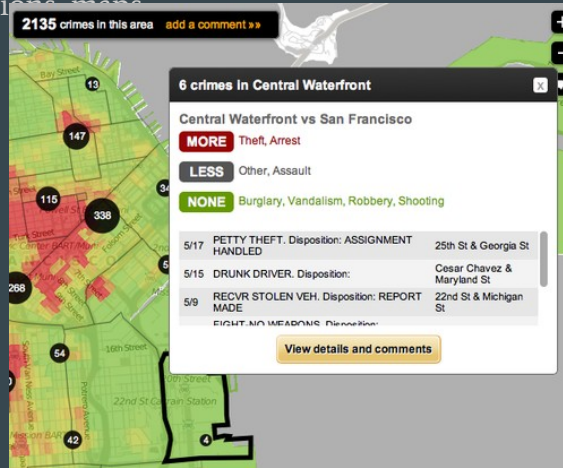
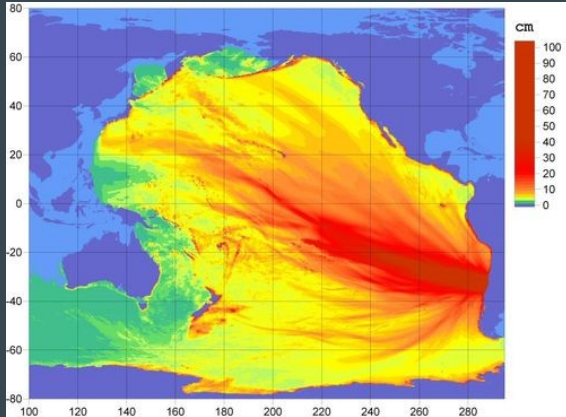
no-sql databases (mongodb, couchdb)



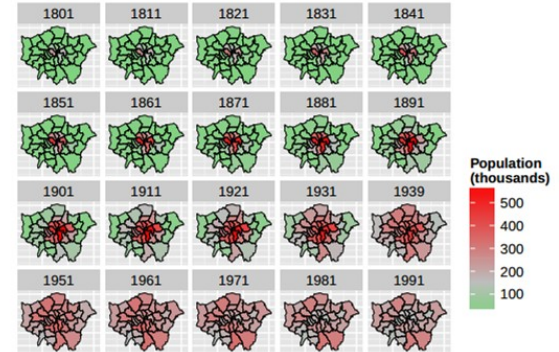
PROGRAMMING LANGUAGES IN GEOINFORMATICS

R

- designed by statisticians, for statisticians (?)
- created primarily for (big) data analysis
- implemented statistical methods
- creating graphs, visualization



```
ggplot(data = lnd_f, # the input data
  aes(x = long, y = lat, fill = pop/1000, group = group)) + # define variables
  geom_polygon() + # plot the boroughs
  geom_path(colour="black", lwd=0.05) + # borough borders
  coord_equal() + # fixed x and y scales
  facet_wrap(~ date) + # one plot per time slice
  scale_fill_gradient2(low = "green", mid = "grey", high = "red", # colors
  midpoint = 150, name = "Population\n(thousands)") + # legend options
  theme(axis.text = element_blank(), # change the theme options
  axis.title = element_blank(), # remove axis titles
  axis.ticks = element_blank()) # remove axis ticks
# ggsave("figure/facet_london.png", width = 9, height = 9) # save figure
```



PROGRAMMING LANGUAGES IN GEOINFORMATICS

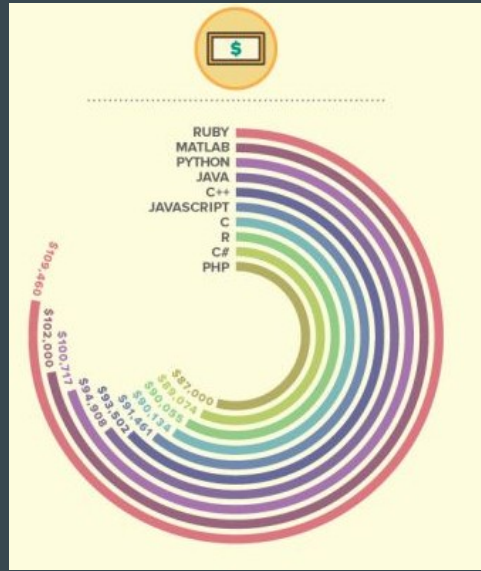
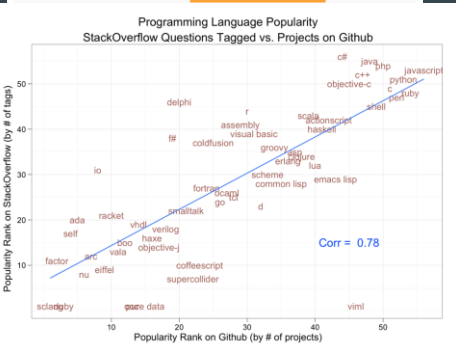
Python

- universal language - lot of possibilities
 - in geoinformatics:
 - ArcGIS scripting - automatization of processes
 - visualization and mapping
 - processing data
 - ...
- easy to learn
- “scientific language”
- web applications, servers, desktop

PROGRAMMING LANGUAGES IN GEOINFORMATICS

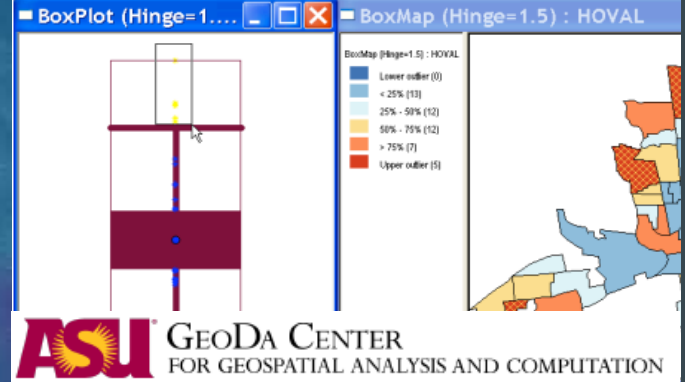
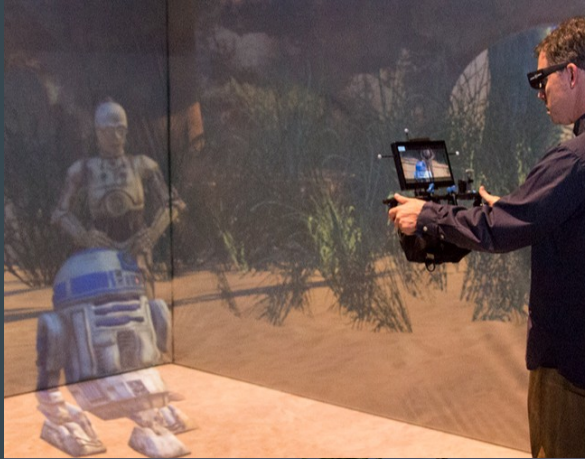
Summary

Language Rank	Types	Spectrum Ranking
1. Java	☉ 📱 🖨	100.0
2. C	📱 🖨	99.9
3. C++	📱 🖨	99.4
4. Python	☉ 📱 🖨	96.5
5. C#	☉ 📱 🖨	91.3
6. R	📱 🖨	84.8
7. PHP	☉ 📱 🖨	84.5
8. JavaScript	☉ 📱 🖨	83.0
9. Ruby	☉ 📱 🖨	76.2
10. Matlab	📱 🖨	72.4
11. SQL	📱 🖨	71.3
12. Shell	📱 🖨	70.7
13. Go	☉ 📱 🖨	67.3
14. Perl	☉ 📱 🖨	66.3
15. Swift	📱 🖨	64.7
16. Visual Basic	📱 🖨	64.4
17. Scala	☉ 📱 🖨	63.3
18. Objective-C	📱 🖨	60.5



- <http://gis.stackexchange.com/questions/61221/what-programming-languages-are-used-by-different-gis-platforms>
- <http://www.atlefen.net/post/2015/04/gis-programming-languages-breakdown/>
- <http://www.quora.com/Which-programming-language-is-best-for-geographical-information-system-GIS-software>
- <http://cdn-media-2.lifehack.org/wp-content/files/2015/07/e25a76a815e29c38d88f1035629b7eaa-2.jpg>

MOTIVATION - Python is everywhere



MOTIVATION - Solving problems

- automatization of work
 - tasks that could be repeated
 - tasks that are boring...
 - creating interactive tool for solving task
- diploma thesis,
Calculate the border between Thailand and Myanmar
- getting data,
- parsing data,
visualization,
(WESTRA - Python Geospatial Development)

```
import os,os.path,shutil
import osgeo.ogr
import shapely.wkt

shapefile = osgeo.ogr.Open("TM_WORLD_BORDERS-0.3.shp")
layer = shapefile.GetLayer(0)
thailand = None
myanmar = None

for i in range(layer.GetFeatureCount()):
    feature = layer.GetFeature(i)
    if feature.GetField("ISO2") == "TH":
        geometry = feature.GetGeometryRef()
        thailand = shapely.wkt.loads(geometry.ExportToWkt())
    elif feature.GetField("ISO2") == "MM":
        geometry = feature.GetGeometryRef()
        myanmar = shapely.wkt.loads(geometry.ExportToWkt())

commonBorder = thailand.intersection(myanmar)

if os.path.exists("common-border"):
    shutil.rmtree("common-border")
    os.mkdir("common-border")
    spatialReference = osgeo.osr.SpatialReference()
    spatialReference.SetWellKnownGeogCS('WGS84')
    driver = osgeo.ogr.GetDriverByName("ESRI Shapefile")
    dstPath = os.path.join("common-border", "border.shp")
    dstFile = driver.CreateDataSource(dstPath)
    dstLayer = dstFile.CreateLayer("layer", spatialReference)
    wkt = shapely.wkt.dumps(commonBorder)

    feature = osgeo.ogr.Feature(dstLayer.GetLayerDefn())
    feature.SetGeometry(osgeo.ogr.CreateGeometryFromWkt(wkt))
    dstLayer.CreateFeature(feature)
    feature.Destroy()
    dstFile.Destroy()
```

MOTIVATION - Getting jobs

jobs **CZ**

Nabídky práce

Brigády

Kurzy

Vysoké školy

Inspirace a rady

Přihlásit

Vstup pro firmy

Jakou práci hledáte?

gis x přidat další...

Kde?

např. Jihlava



Benefity ▾ Čas zveřejnění ▾ Plat ▾ Typ úvazku ▾ Vzdělání ▾ Jazyky ▾ Zadavatel ▾ Vhodné pro ▾

Našli jsme **10** nabídek

Jakou práci hledáte?

python x přidat další...

Kde?

např. Jihlava



Benefity ▾ Čas zveřejnění ▾ Plat ▾ Typ úvazku ▾ Vzdělání ▾ Jazyky ▾ Zadavatel ▾ Vhodné pro ▾

Našli jsme **173** nabídek

COURSE - conditions and grading system

7 short exams (+- 5 questions) for -2 to +1 points

2 tests (with internet) for 0 to 10 points and 0 to 10 points

1 team project for 0 to 20 points

grading (required min. 5 points from team project):

0 - 17 F

18 - 24 E

25 - 28 D

29 - 32 C

33 - 36 B

37+ A

COURSE - organization

Monday 9:00 - 13:00 Z7

- I. 05. 10. 2015 - Variables, basic commands, basic data structures
- II. 12. 10. 2015 - Conditions, lists, cycles (+ exam)
- III. 19. 10. 2015 - Tuples, dictionaries, sets, functions (+ exam)
- IV. 26. 10. 2015 - Reading/writing files, exceptions, repeating (+ exam)
- V. 02. 11. 2015 - Libraries - numpy, PyQt, request (+ exam)
- VI. 09. 10. 2015 - Test 1
- VII. 16. 10. 2015 - Visualization (+ exam)
- VIII. 23. 10. 2015 - Handling data (+ exam)
- IX. 30. 10. 2015 - QGIS / arcGIS scripting (+ exam)
- X. 07. 12. 2015 - Test 2
- XI. 14. 12. 2015 - Project presentations

COURSE - organization

Homeworks

- not graded, optional
- solutions in IS later

Team project

- team ~ 3 people
- presentation
- protocol
- documented code

Absence

- send mail before

Short exams

- at the beginning of the lesson

WHERE TO LEARN Online

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

learn python - <http://www.learnpython.org/>

learn python the hard way -

<http://learnpythonthehardway.org/book/index.html>

python scientific lecture notes - <http://scipy-lectures.github.io/index.html>

pyQGIS -

http://docs.qgis.org/testing/en/docs/pyqgis_developer_cookbook/

tutorialspoint - <http://www.tutorialspoint.com/python/>

Python course - <http://www.python-course.eu/course.php>

Functions

What are Functions?

Functions are a convenient way to divide your code into useful blocks, allowing us to order our code, make it more readable, reuse it and save some time. Also functions are a key way to define interfaces so programmers can share their code.

How do you write functions in Python?

As we have seen on previous tutorials, Python makes use of blocks.

A block is a area of code of written in the format of:

```
block_head:  
1st block line  
2nd block line  
...
```

Code Window

Run

Reset

Solution

Output Window

Expected Output

```
1 # Modify this function to return a list of strings as defined above  
2 def list_benefits():  
3     pass  
4  
5 # Modify this function to concatenate to each benefit - " is a benefit"  
6 def build_sentence(benefit):  
7     pass  
8  
9 def name_the_benefits_of_functions():  
10    list_of_benefits = list_benefits()  
11    for benefit in list_of_benefits:  
12        print build_sentence(benefit)  
13
```

```
Traceback (most recent call last):  
File "prog.py", line 14, in <module>  
File "prog.py", line 11, in name_the_benefits_of_  
TypeError: 'NoneType' object is not iterable
```

<http://www.learnpython.org/>



<https://codecombat.com/play>

WHERE TO LEARN Ebooks

SWAROOP - A Byte of python (<http://www.mybridge.co/view/5668>)

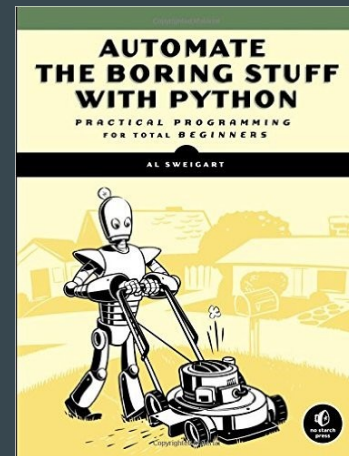
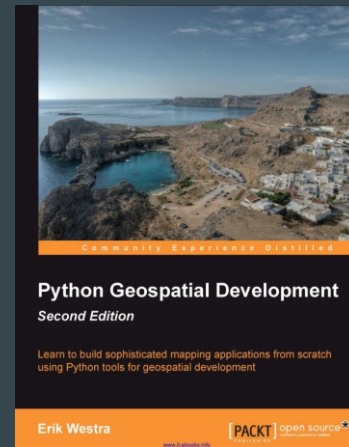
SWEIGART - Automate the boring stuff with Python (<https://automatetheboringstuff.com/>)

SEVERANCE - Python for Informatics (<http://do1.dr-chuck.com/py4inf/EN-us/book.pdf>)

LAWHEAD - Learning Geospatial Analysis with Python

WESTRA - Python Geospatial Development (<http://www.allitebooks.com/python-geospatial-development>)

MCKINNEY - Python for Data Analysis
(<http://www.cin.ufpe.br/~embat/Python%20for%20Data%20Analysis.pdf>)



WHERE TO LEARN

Blogs, Social, Communities

<http://stackoverflow.com> / gis.stackexchange.com

<http://www.python.cz>

<http://geospatialpython.com/> - Points, Polylines, Polygons, Pixels, Python!

<http://jakevdp.github.io/> - Musings and ramblings through the world of Python and beyond

<http://www.danielforsyth.me/>

<http://nbviewer.ipython.org/> - A simple way to share Jupyter Notebooks

<http://www.analyticsvidhya.com/blog/category/python-2/> - Learn everything About Analytics

...

```
import matplotlib
import matplotlib.pyplot as plt
#Inline Plotting for Ipython Notebook
%matplotlib inline

pd.options.display.mpl_style = 'default' #Better Styling
new_style = {'grid': False} #Remove grid
matplotlib.rc('axes', **new_style)
from matplotlib import rcParams
rcParams['figure.figsize'] = (17.5, 17) #Size of figure
rcParams['figure.dpi'] = 250

P.set_axis_bgcolor('black') #Background Color

P=df.plot(kind='scatter', x='long', y='lat',color='white',xlim=(-74.06,-73.77),ylim=(40.61,
40.91),s=.02,alpha=.6)
```

And the Result:



WHERE TO LEARN Editors

pyzo - <http://www.pyzo.org/>

komodo IDE - <http://komodoide.com/>

sublime text - <http://www.sublimetext.com/>

pywin - <http://sourceforge.net/projects/pywin32/files/pywin32/>

pycharm - <https://www.jetbrains.com/pycharm/>

atom - <https://atom.io/>

learnpython - <http://www.learnpython.org/>

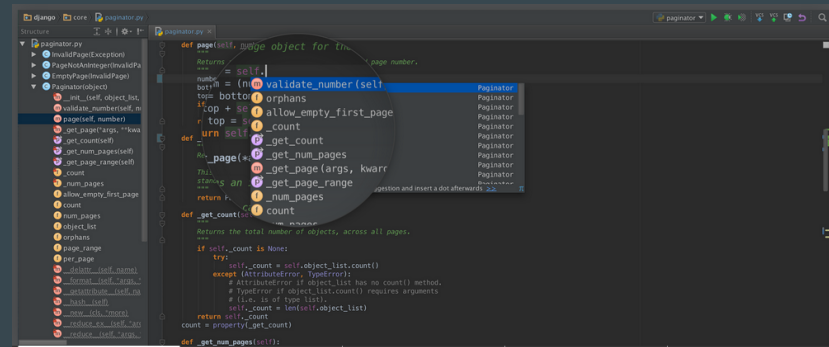
getdatajoy - www.getdatajoy.com

pyfiddle - <http://pythonfiddle.com/>



The banner features the Atom logo (a stylized atom symbol) and the text "ATOM A hackable text editor for the 21st Century". On the right, there is a circular video player showing a group of people. Below the text is a red button that says "Download Windows Installer" with a download icon. Underneath the button, it says "For Windows 7 & 8 - Other platforms". The bottom half of the banner shows a screenshot of the Atom editor interface with a file named "atom.coffee" open. The code in the editor is as follows:

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



The screenshot shows a Python IDE with a file named "paginator.py" open. The code defines a "Paginator" class with several methods. A callout box highlights the "validate_number" method. The code is as follows:

```
def page(self, page: object for the
Returns
number = (self.validate_number(self
bottom orphans
top + self.allow_empty_first_page
top = self._count
return self._get_count
def _page(self, _get_num_pages
num, _get_page(args, kwargs
return self._get_page_range
return self._num_pages
def _get_count(self)
Returns the total number of objects, across all pages.
if self._count is None:
try:
self._count = self.object_list.count()
except (AttributeError, TypeError):
# AttributeError if object_list has no count() method.
# TypeError if object_list.count() requires arguments
# (like, as of type list).
self._count = len(self.object_list)
return self._count
return properly_get_count()
def _get_num_pages(self):
```