Automating GIS-processes
FEC 2017

Lecturers: Vuokko Heikinheimo & Henrikki Tenkanen
vuokko.heikinheimo@helsinki.fi
henrikki.tenkanen@helsinki.fi

6.3.2017

Materials: Henrikki Tenkanen, David Whipp, Vuokko Heikinheimo

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Goals of the course

There are basically four goals in this intensive course

1. Introduce the Python programming language
2. Develop basic programming skills
3. Discuss essential (good) programming practices needed by young scientists
4. Introduce automatization of different GIS tasks in Python

Goals of this lecture

- Provide an overview of basic computing practices, and why you should learn them

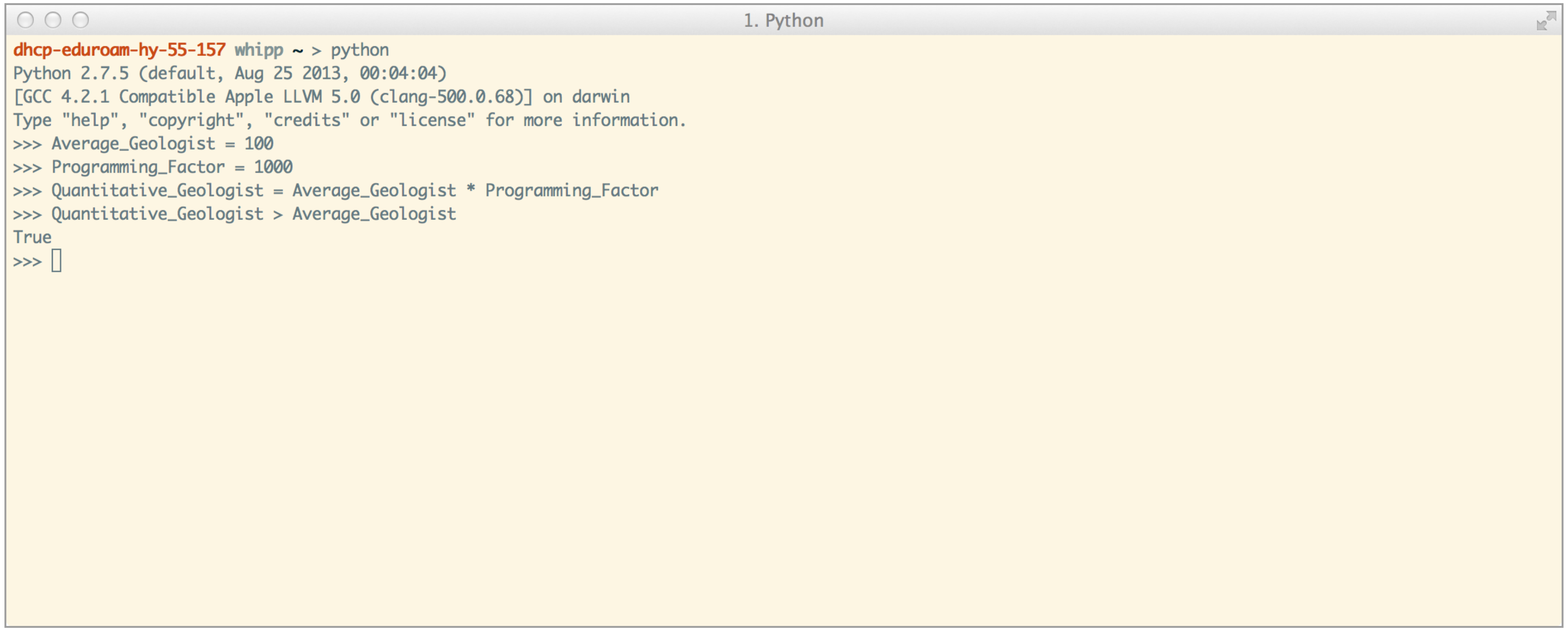
- Define computers and programming languages, and how they operate

- Look at the components of a computer program and a <u>strategy for writing your own code</u>

Learning to program

- A significant part of this course will be development of basic programming skills that will help you write and use simple numerical models
  - I know you're not computer scientists - I'm not either
    - Our goal is take small steps to learn together
  - Do you really need to know how to program? Yes.
    - You might not be a superstar, but learning to write simple codes can be very useful

Why learn to program?

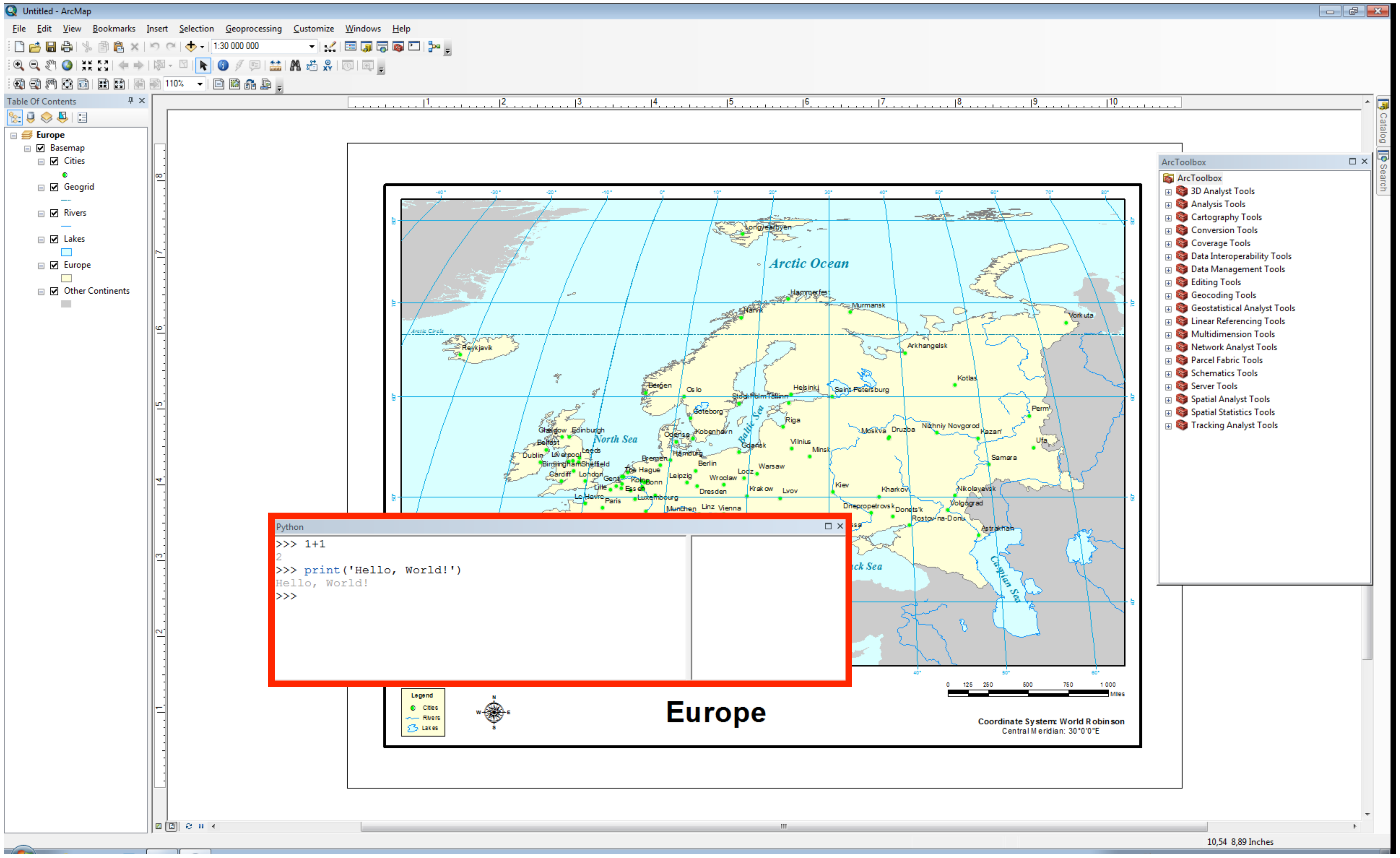


```
dhcp-eduroam-hy-55-157 whipp ~ > python
Python 2.7.5 (default, Aug 25 2013, 00:04:04)
[GCC 4.2.1 Compatible Apple LLVM 5.0 (clang-500.0.68)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> Average_Geologist = 100
>>> Programming_Factor = 1000
>>> Quantitative_Geologist = Average_Geologist * Programming_Factor
>>> Quantitative_Geologist > Average_Geologist
True
>>> 
```

- Rather than being restricted to using existing software, you will have the ability to <u>develop your own solutions</u> when solutions do not exist or are inefficient
    - Many software packages offer the ability to extend their capabilities by <u>adding your own short programs</u> (e.g., ArcGIS, ParaView, Google Earth, etc.)

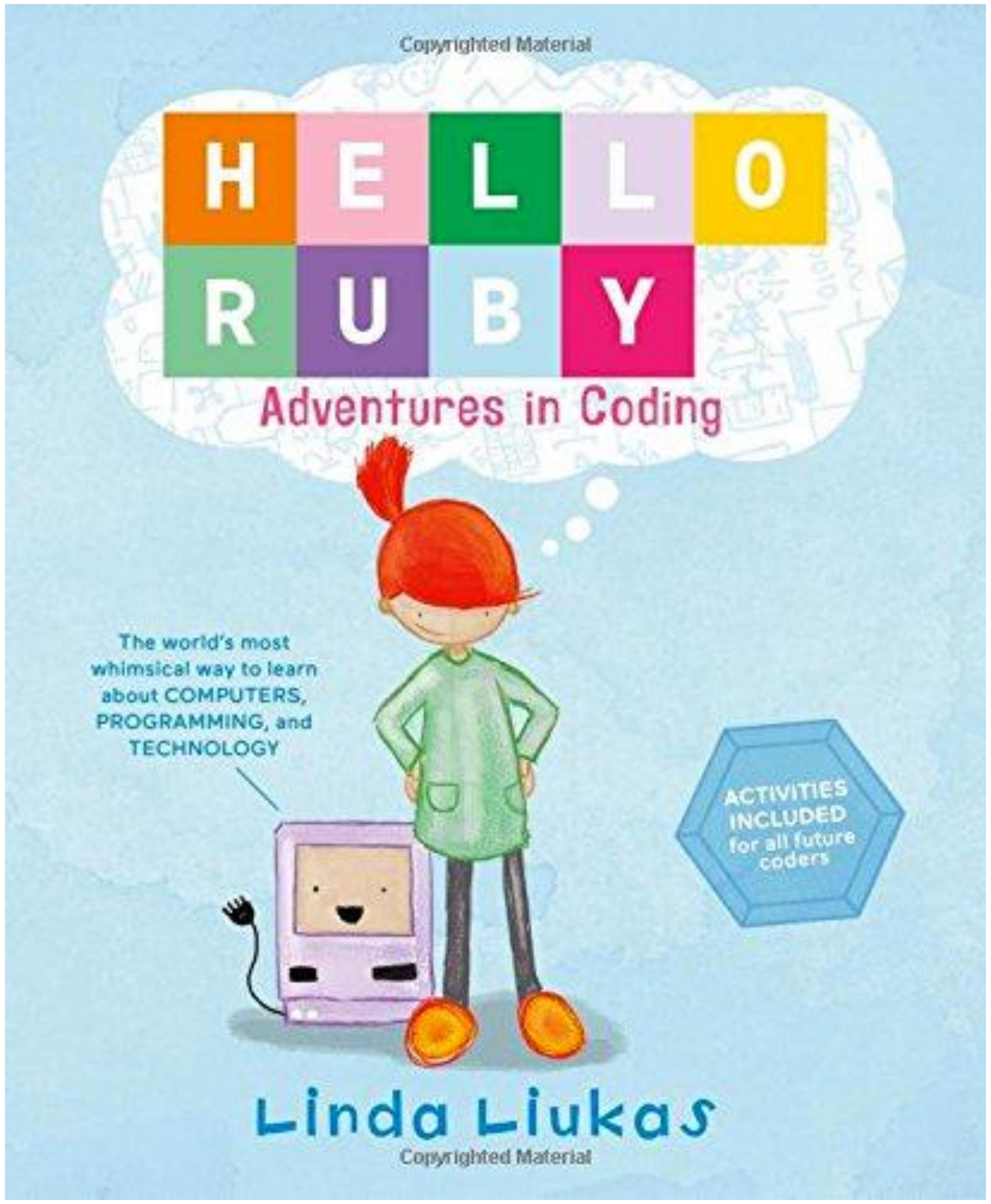
HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Python can be called directly from ArcGIS



```
>>> 1+1
2
>>> print('Hello, World!')
Hello, World!
>>>
```

Why learn to program?


Copyrighted Material
HELLO RUBY
Adventures in Coding
The world's most whimsical way to learn about COMPUTERS, PROGRAMMING, and TECHNOLOGY
ACTIVITIES INCLUDED for all future coders
Linda Liukas
Copyrighted Material

- Believe it or not, programming is fun! It involves
  - Breaking complex problems down into simpler pieces
  - Developing a strategy for solving the problem
  - Testing your solution
- All of this can be exciting and rewarding (when the code works…)

The scientific method…

…and how programming can make you a better scientist

1. Define a question
2. Gather information and resources (observe)
3. Form an explanatory hypothesis
4. Test the hypothesis by performing an experiment and collecting data in a reproducible manner
5. Analyze the data
6. Interpret the data and draw conclusions that serve as a starting point for new hypothesis
7. Publish results
8. Retest (frequently done by other scientists)

Learning to program can help us…

1. Define a question
2. Gather information and resources (observe)
3. Form an explanatory hypothesis
4. Test the hypothesis by performing an experiment and collecting data in a reproducible manner
5. Analyze the data
6. Interpret the data and draw conclusions that serve as a starting point for new hypothesis
7. Publish results
8. Retest (frequently done by other scientists)

Good programming practices can help us…

1. Define a question
2. Gather information and resources (observe)
3. Form an explanatory hypothesis
4. Test the hypothesis by performing an experiment and collecting data in a reproducible manner
5. Analyze the data
6. Interpret the data and draw conclusions that serve as a starting point for new hypothesis
7. Publish results
8. Retest (frequently done by other scientists)

Programming

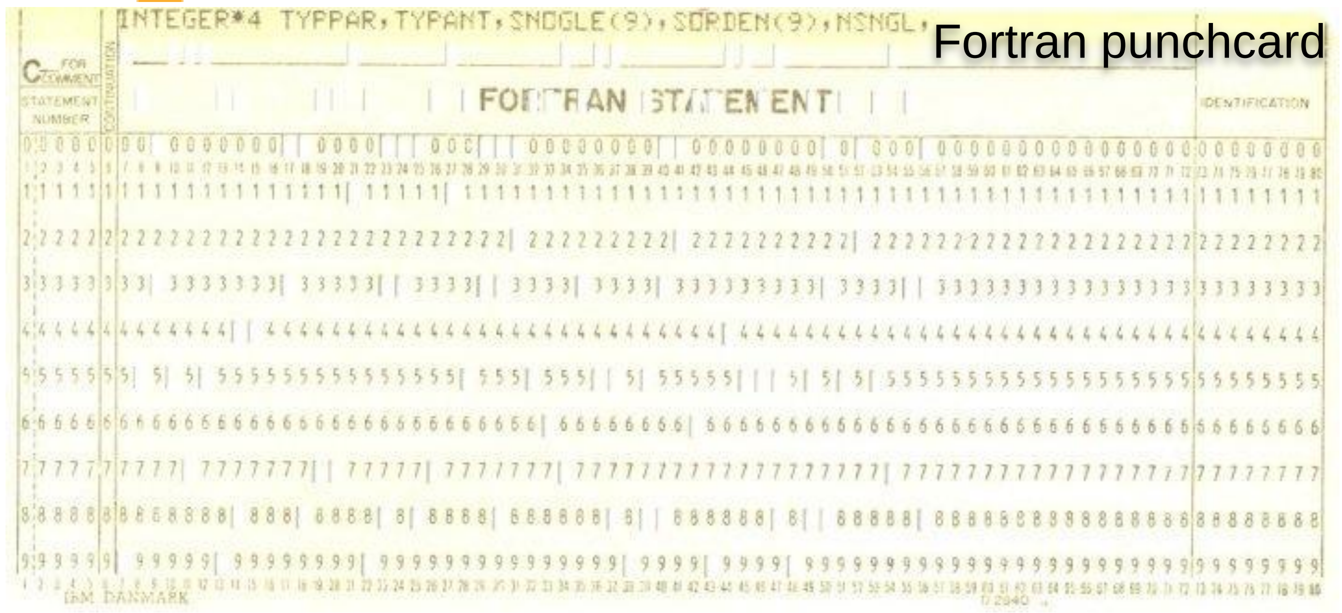
"Computer programming (often shortened to programming) is **a process** that leads from an original formulation of a computing problem to executable computer programs. Programming involves activities such as analysis, developing understanding, generating algorithms, verification of requirements of algorithms including their correctness and resources consumption, and implementation (commonly referred to as coding) of algorithms in a target programming language."

-Wikipedia (2015)

"A program is like a recipe. It contains a list of ingredients (called variables) and a list of directions (called statements) that tell the computer what to do with the variables."

- Webopedia (2015)

What is a program?



```
INTEGER*4 TYPPAR,TYPANT,SNDGLE(9),SORDEN(9),NSNGL,
```

Fortran punchcard

```
# Define plot variables
misfit = NA_data[:,0]
var1 = NA_data[:,1]
var2 = NA_data[:,2]
var3 = NA_data[:,3]
clrmin = round(min(misfit),3)
clrmax = round(min(misfit),2)
trans = 0.75
ptsize = 40
```

Python source code

- A program is a detailed list of step-by-step instructions telling the computer exactly what to do
- The program can be changed to alter what the computer will do when the code is executed
- Software is another name for a program

What is a programming language?

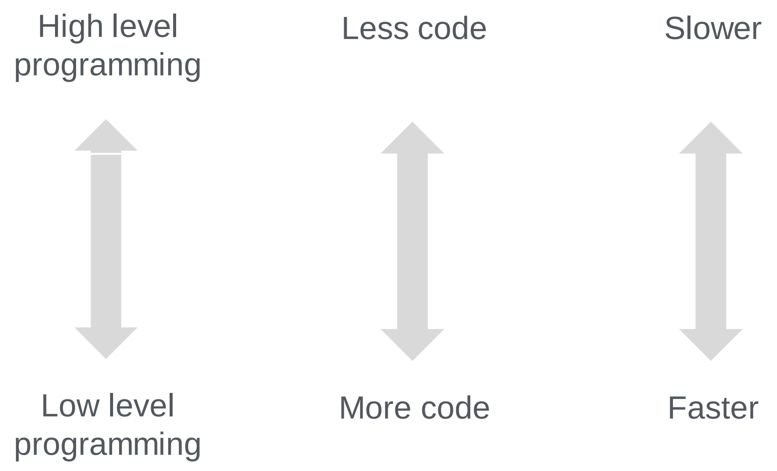
- A computer language is what we use to ‘talk’ to a computer
    - Unfortunately, computers don’t *yet* understand our native languages
- A programming language is like a code of instructions for the computer to follow
    - It is exact and unambiguous
    - Every structure has a precise form (syntax) and a precise meaning (semantics)
- Python is just one of many programming languages

Programming

Various different programming languages exist

- Python
- Perl
- Ruby
- JavaScript
- Java
- C++
- C
- Fortran

+ Many others



Programming languages remind our spoken languages!

- There are different ways to express the same meaning
- Some languages are more similar to each other than others
- After learning one language it is easier to learn other ones!

Moikka, Hello, Hej, Hallo, Hola

Spoken languages

```
echo ”Moikka”, print(”Hello”), console.log(”Hej”),
puts(”Hallo”), System.out.println(”Hola”)
```

Programming languages

Programming

Python is:

- General purpose
- High level
- Cross-platform
- Open source
- Multi-paradigm: Object oriented / imperative / functional / procedural / reflective
- Uses dynamic type-checking

Why to learn / use Python?

- Easy to learn (a good programming language for the beginners)
- Easy to read and understand – elegant code
- Powerful enough – used in scientific programming
- Countless ready-made modules / libraries to use (also GIS stuff)
- Supportive, large and helpful user community
- Widely supported in different GIS-softwares
  - ArcGIS, QGIS, Erdas Imagine, IDRISI, uDig, ILWIS, PostGIS …
- **Extremely useful for automating GIS processes**

Developing a program

- Coming up with a specific list of instructions for the computer to follow in order to accomplish a desired task is <u>not easy</u>
- The following list will serve us as a general software development strategy

1. Analyze the problem
2. Determine specifications
3. Create a design
4. Implement the design
5. Test/debug the program
6. Maintain the program (if necessary)

Let’s consider an example

- As an American, David was raised in a country that uses Fahrenheit for temperatures
  - 70°F is lovely
  - 90°F is hot
  - Water freezes at 32°F

- The problem here in Finland is that he doesn’t always know what he should wear to work when he finds weather reports with temperatures in degrees Celsius
  - A simple program could help

Developing a program

1. Analyze the problem

   - Before you can solve a problem, you must figure out exactly what should be solved

Developing a program

1. Analyze the problem
    - Before you can solve a problem, you must figure out exactly what should be solved
2. Determine specifications
    - Describe exactly what the program will do
        - Don't worry about how it will work. Determine the input and output values and how they should interact in the program

Developing a program

3. Create a design
   - What is the overall structure of the program? How will it work?
   - It is often helpful to write out the code operation in pseudocode, precise English (or Finnish) describing the program. Be specific!

Developing a program

3. Create a design
    - What is the overall structure of the program? How will it work?
    - It is often helpful to write out the code operation in pseudocode, precise English (or Finnish) describing the program. Be specific!
4. Implement the design
    - If you've done a good job with the previous steps, this should be fairly straightforward. Take your pseudocode and 'translate' it into Python

Developing a program

5. Test/debug the program

- Now you can put your new Python code to the test (literally) by running it to see whether it reproduces the expected values
  - For any test, you should know the correct values in advance of running your code. How else can you confirm it works???

Developing a program

5. Test/debug the program
    - Now you can put your new Python code to the test (literally) by running it to see whether it reproduces the expected values
        - For any test, you should know the correct values in advance of running your code. How else can you confirm it works???
6. Maintain the program
    - If you've written something that will be shared by other users, a helpful programmer will continue to add features that are requested by the users

References

Zelle, J. M. (2010). *Python programming: an introduction to computer science* (2nd ed.). Franklin, Beedle & Associates, Inc.