ETH zürich

Introduction to Scientific Computation
Lecture 0
Fall 2023

The rules of the game
Intro

Mikhail Usvyatsov September 20, 2023

General course info

Lectures can be downloaded and viewed on Github:
https://github.com/Aelphy/ISC
There you can find information on how to prepare the environment, launch the code, submit assignments, communicate with staff etc.

Team & TAs:

Mikhail Usvyatsov

Communication system

Communication with staff (questions, suggestions, etc) could be done via emails m.usvyatsov+ethz@gmail.com. Please make sure that the subject in your email is [ISC2023]
Telegram chatroom for students is here: https://t.me/iscfall23

Plagiarism

Students are encouraged to discuss the material, but all the assignments should be done individually. Assignments will be checked for plagiarism. If noticed, can be punished in various ways from zeroing out some parts of a specific assignment to unsatisfactory grade for the course.

Assignments

This course uses Github for submitting assignments. You should create a dedicated **private!** repository with name **ISC2023-solutions**. E.g. github.com/YOUR_NAME/ISC2023-solutions. And add me (@Aelphy) to collaborators.
Assignment becomes submitted after sending email to m.usvyatsov+ethz@gmail.com with the subject [ISC2023: assignment_N], where N is the number of assignment. In the body of this email there should be a link to a dedicated folder for this particular assignment. For the first assignment it would be: github.com/YOUR_NAME/ISC2023-solutions/assignment_0

LLMs

Various AI systems became way too advanced as of the Fall 2023.
When extensive similarity with AI generated content is suspected, the student pretending for the authorships will face a defence procedure: detailed questions to test how well did the student understand the underlying matters.

Deadlines & Attendance

Unless stated otherwise, submission deadline is 00:01 of 10's day after hanging out.
You loose 10% points per day until submission (from the full amount of points on the previous day).
By the time of grading the assignment will be checked in the state of the last commit before the deadline. All later commits are subject to late submission policy.
Assignment can't yield negative amount of points, even if submitted 2 years past deadline.
Attendance is not strict, but do not disappoint us.

Grading

The course grade depends entirely on points obtained mostly by completing homework assignments. There will be no course exam at the end.

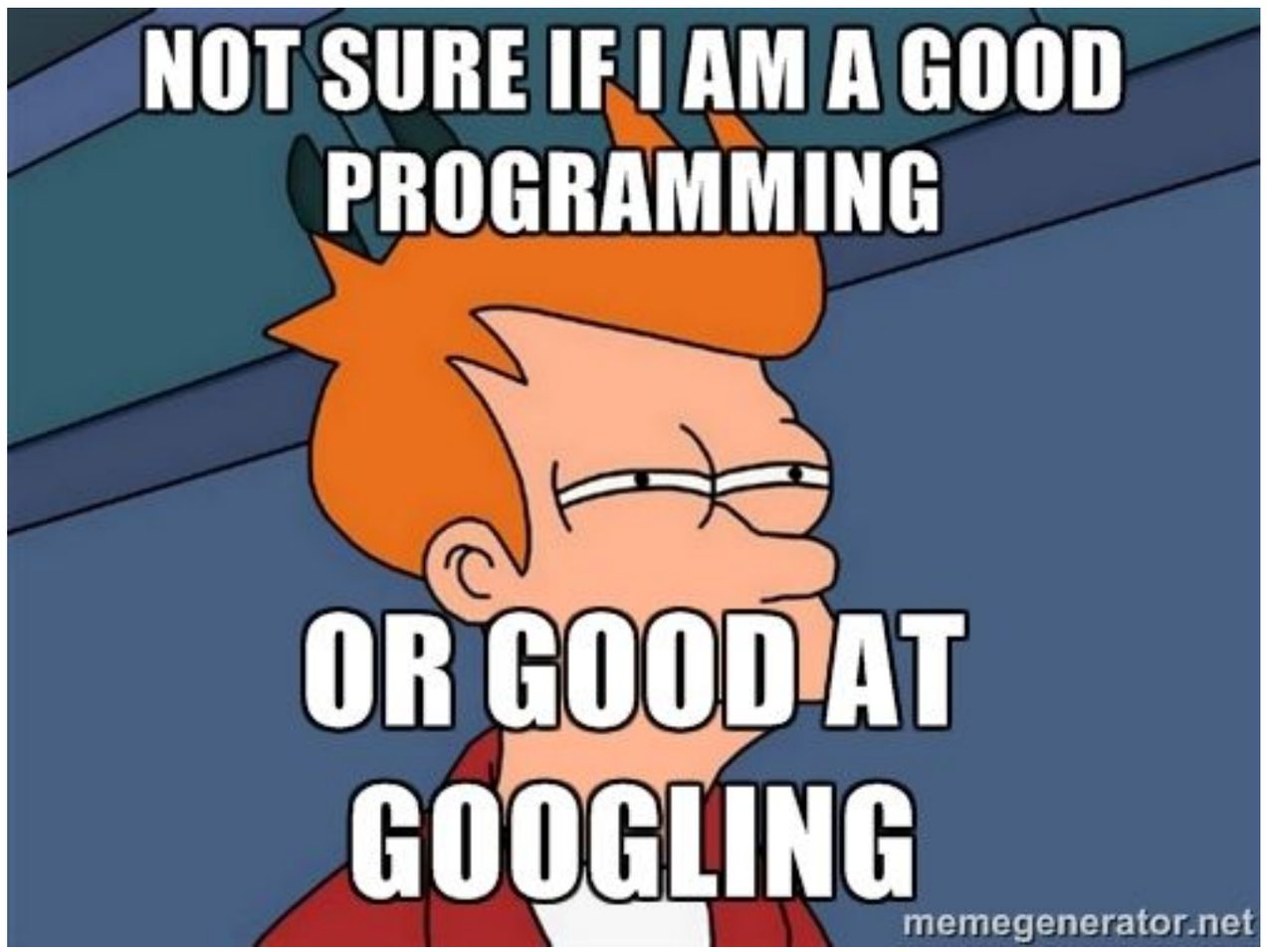
Some assignment examples are "Implement and optimise factorial computation algorithm", "Implement gradient descent algorithm for logistic regression", "Compare Ridge and Lasso regression for house prices problem", "Implement Perceptron Algorithm"

Materials

The Hitchhiker's Guide to Python
The Primer on Scientific Programming With Python. H. Langtangen

Always use

Google
StackOverflow
As of Fall 2023,
chatgpt, Bard, etc.,
might or might not be a good resource for consulting. Mind The references.



Introduction

How is your code running ?

Semantic gaps

Domain applications

- Tickets search platform
  - Airlines
  - Routes
  - Prices
- Modelling celestial motion
  - Speed
  - Space
  - Mass
  - Coordinates
  - Time

Programming primitives

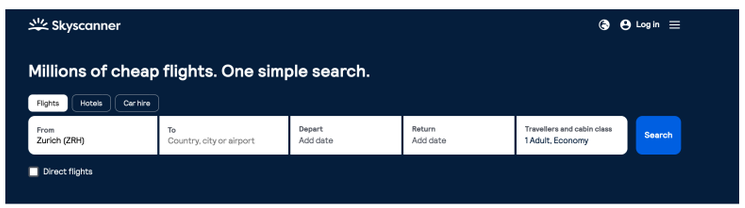
- Class
- Function
- Variable
- Data structures
  - Array
  - List

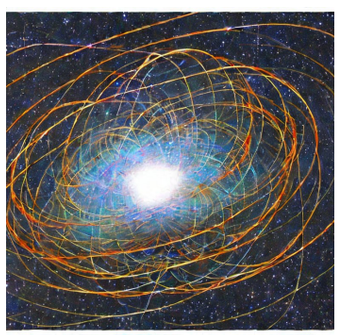
Hardware

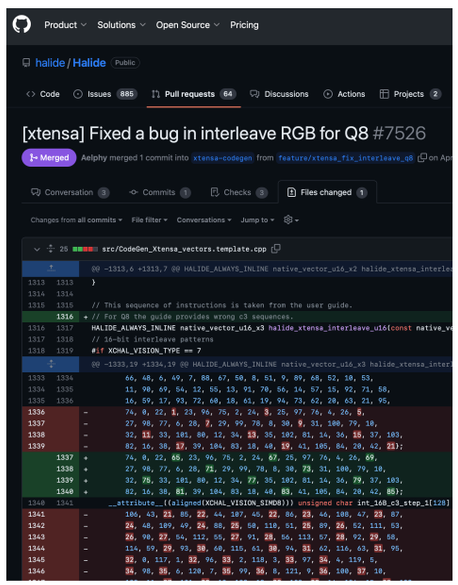
- CPU
  - ARM
  - X86
  - RISC-V
  - Instruction set
  - ASIC
- GPU
- Bytes
- Memory
- Registers
- Cache
- RAM
- Bus

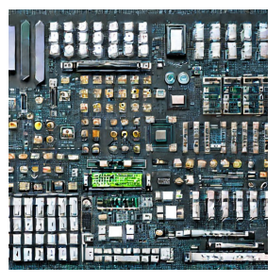
Physics

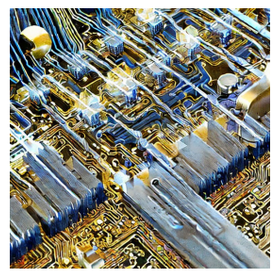
- Transistors
  - Atoms
  - Electrons
- Semiconductors
  - Lithography
  - Nano meters
- Heat
- Radio waves
- Super conductors

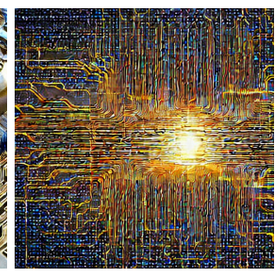


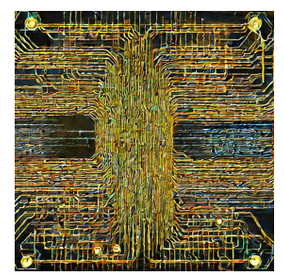












Images were generated with Stable Diffusion from HuggingFace, unless the source is present on an image

Semantic gaps

Programming primitives

- Class
- Function
- Variable
- Data structures
  - Array
  - List

Hardware

- CPU
  - ARM
  - X86
  - RISC-V
  - Instruction set
- GPU
- Bytes
- Memory
- Registers
- Cache
- RAM
- Bus

So, the main compiler task is to map human-readable code to hardware to solve the task efficiently

(c) The Demon Seated. Mikhail Vrubel

Machine Code & Assembly Language

Off-topic

Mnemonic notation:
ADD 5 7

Binary code of the instruction:
1101010010100111

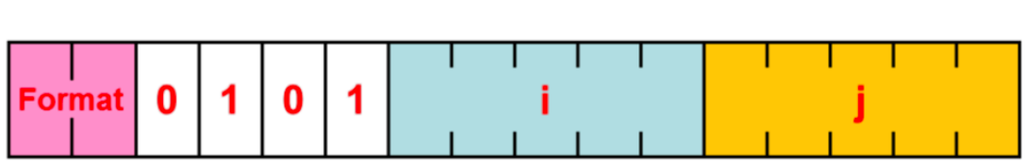
Hexadecimal code of the instruction:
D4A7

Assembler code:

```
.format 32
R5 += R7
```

ADD i j

- **The ADD instruction denotes the two's complement arithmetic addition. The contents of registers Ri and Rj are arithmetically added, and the result is put into the register Rj.**
- The instruction format is as follows:



- Memory state is not considered in the instruction, and the memory state does not change.
- If the addition gives a result which cannot be put into the format specified in the instruction, then **overflow** happens

Suggested assembly statement for the ADD instruction:

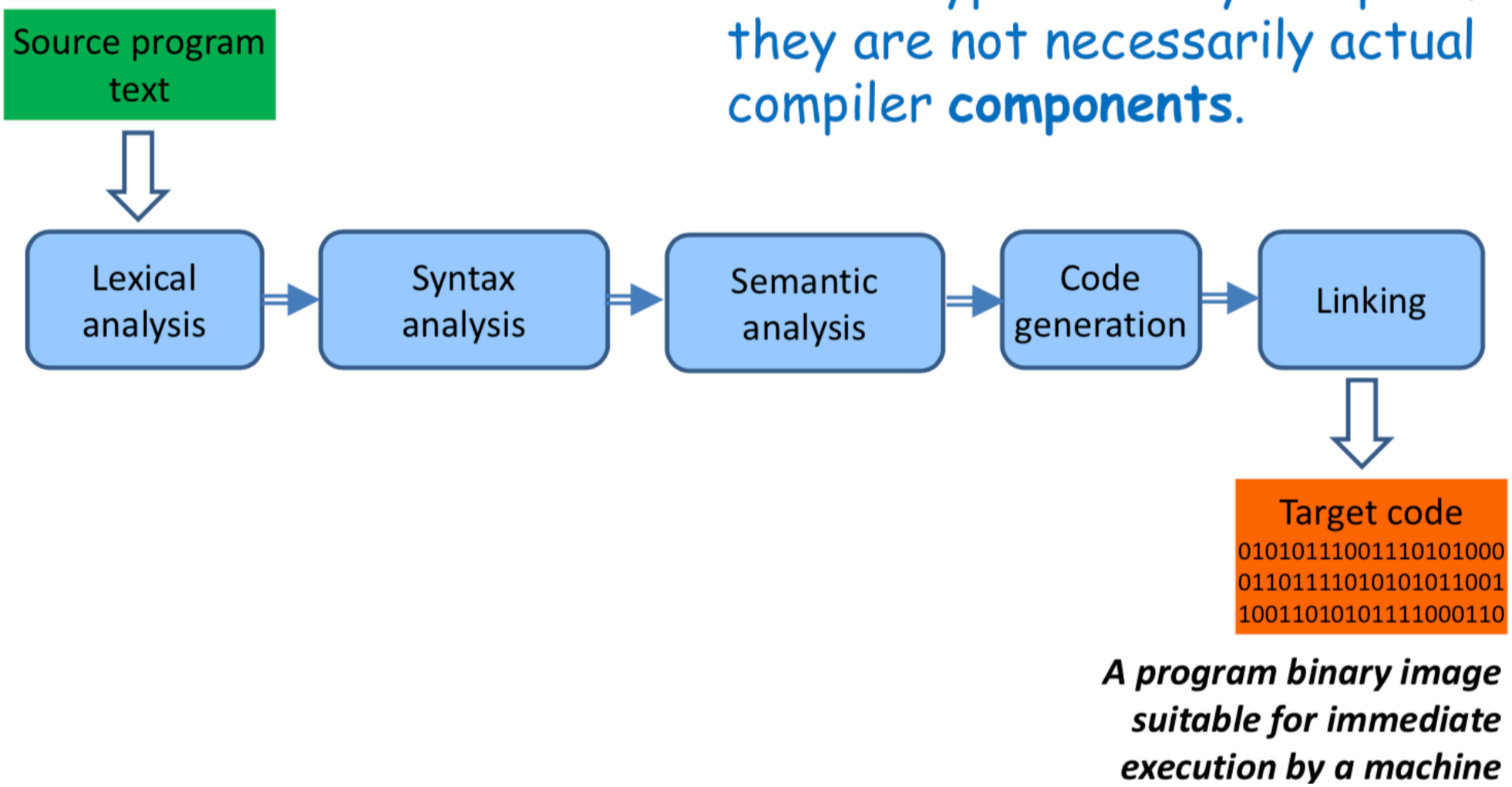
```
Rj += Ri;
```

Additional assembly directives specifying the current instruction format:

```
.format 8; or .format 16; or .format 32;
```

Compilation: An Ideal Picture

A program written by a human (or by another program)

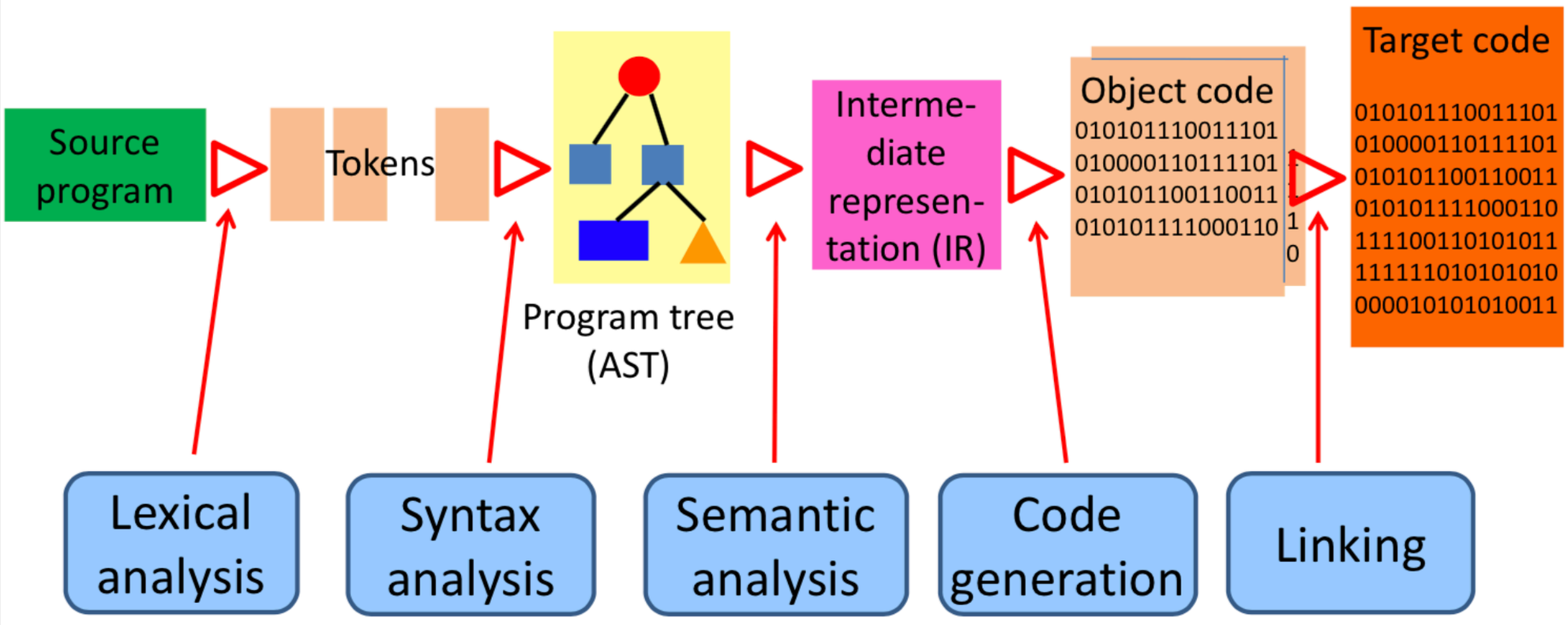


A program binary image suitable for immediate execution by a machine

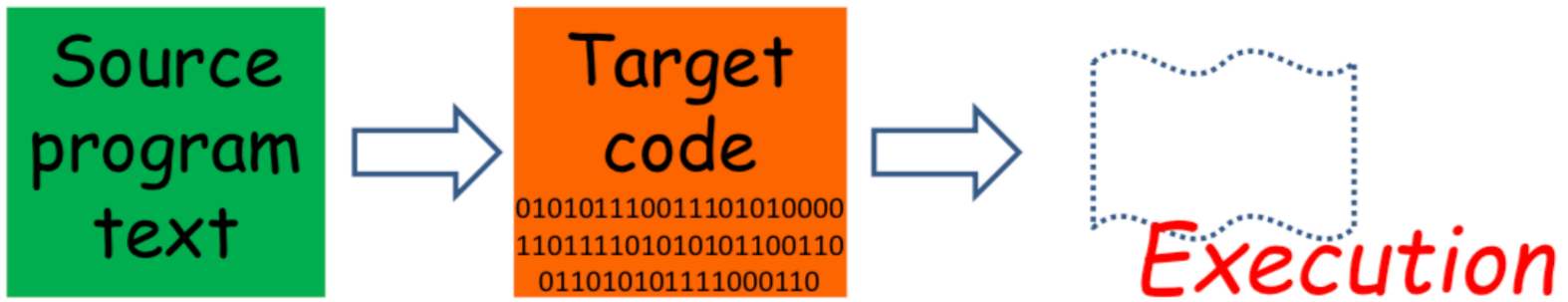
Blue squares just denote some **actions** typical to any compiler; they are not necessarily actual compiler **components**.

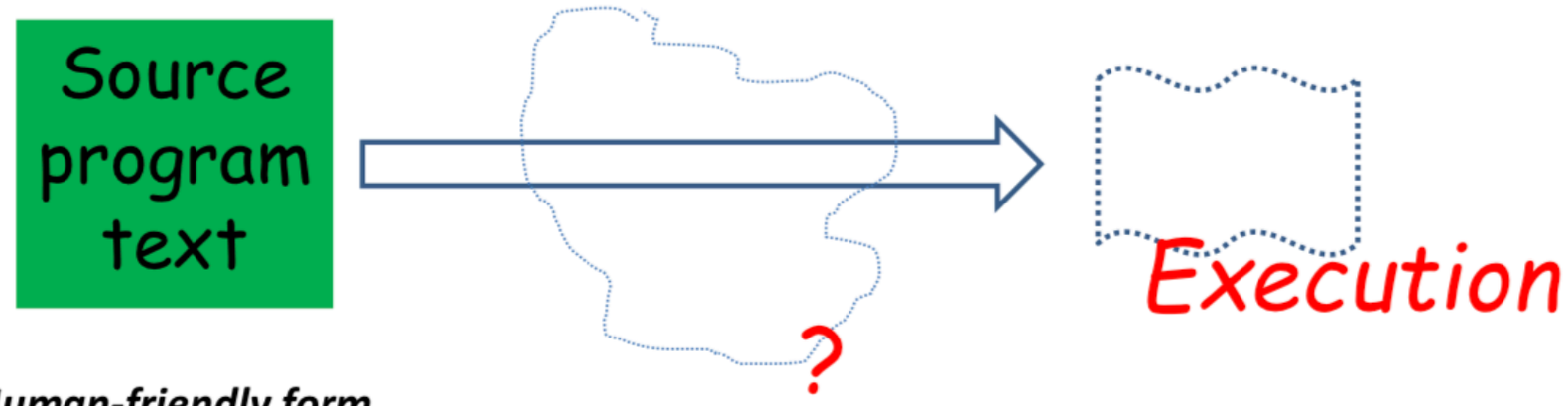
Compilation Data Structures

This is a different, **data-centric** view at the compilation process. Compilation is **a sequence of transformations** of a source program.



Compilation vs Interpretation



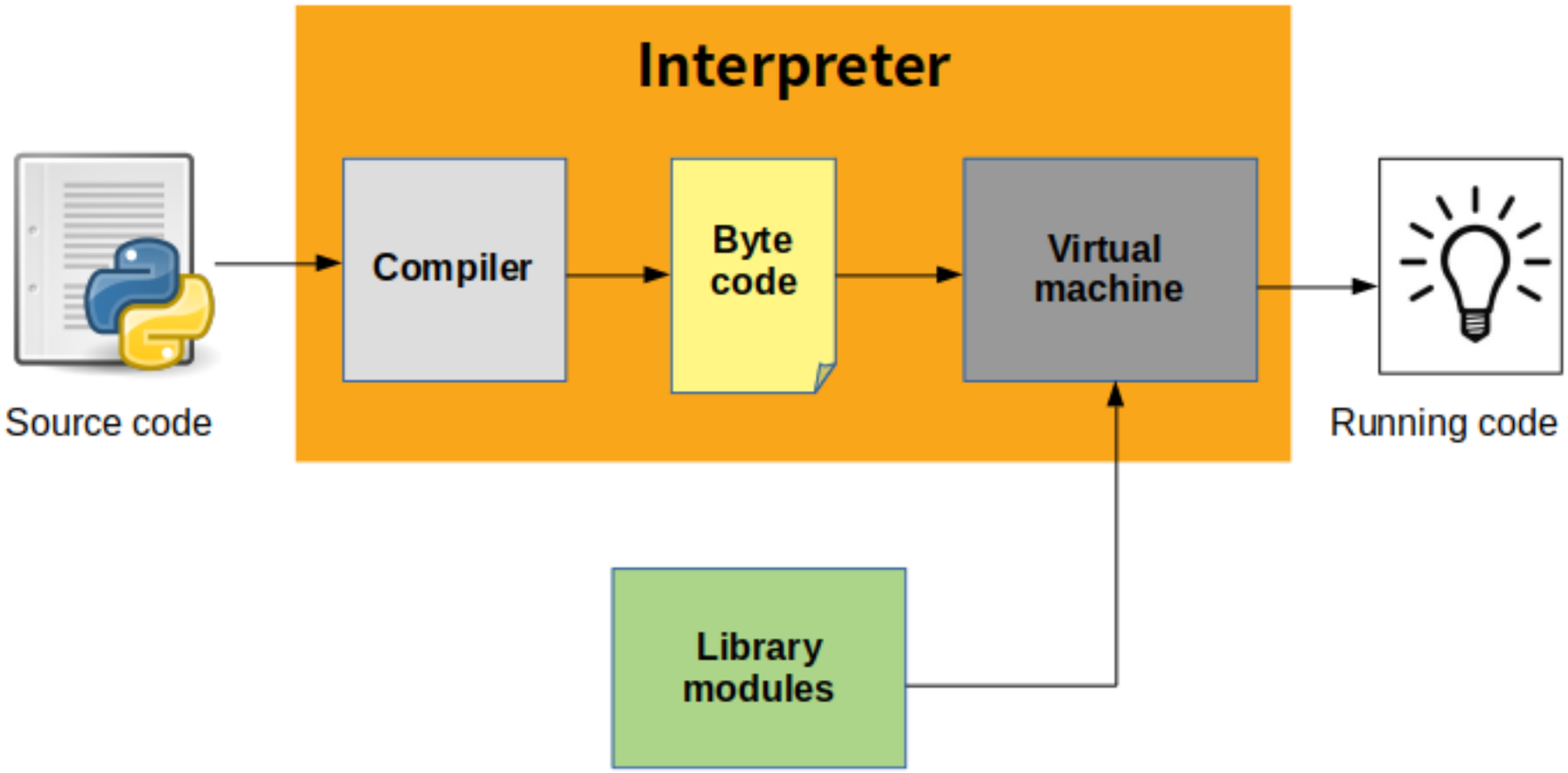


So, what about Python ?

From a programmer point of view

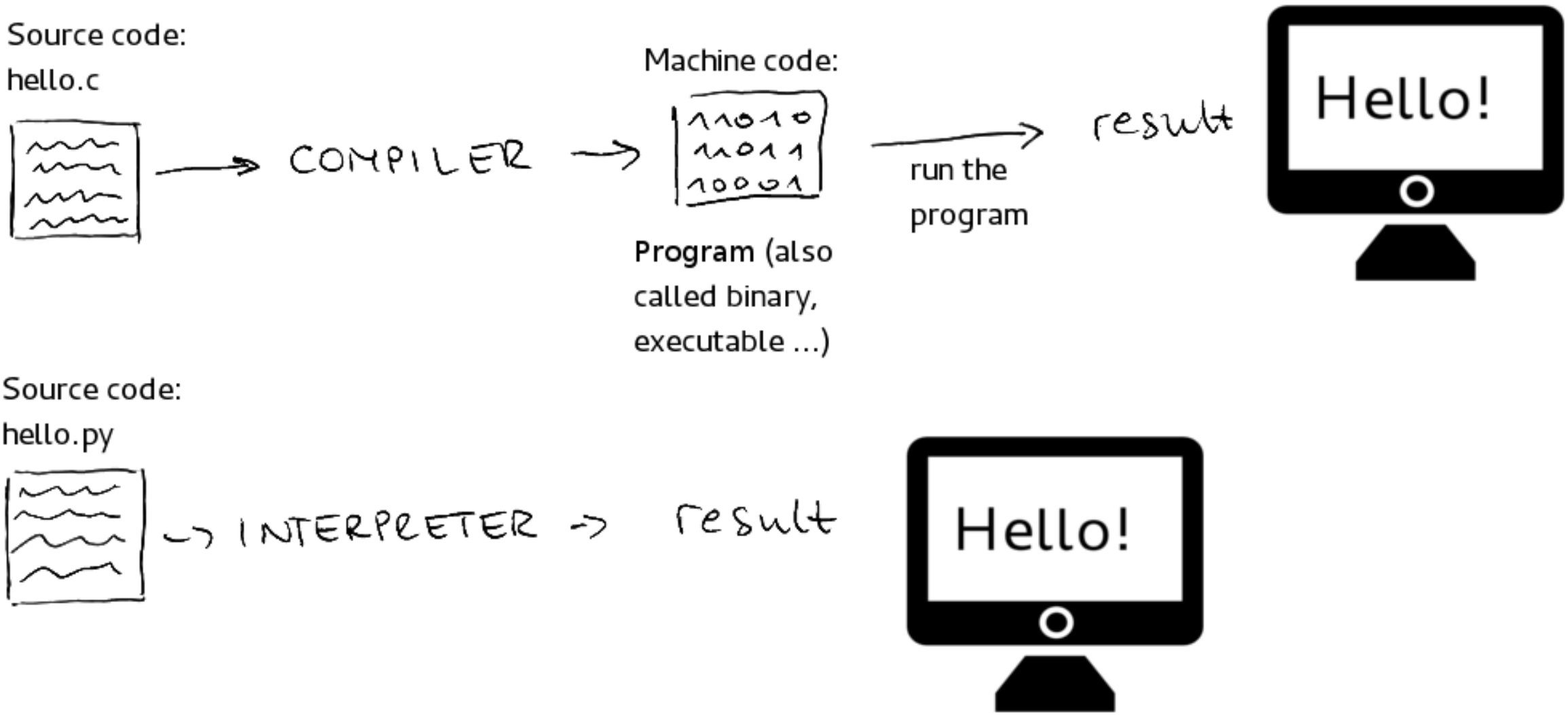
(c) https://indianpythonista.wordpress.com/2018/01/04/how-python-runs/

From Python point of view



(c) https://indianpythonista.wordpress.com/2018/01/04/how-python-runs/

So, what about Python ?



(c) medium.com

How does Python interpret this:

$$X + Y * Z + U$$

How does Python interpret this:

$X + Y * Z + U$

Stack

```
push x
push y
push z
multiply
add
push u
add
```

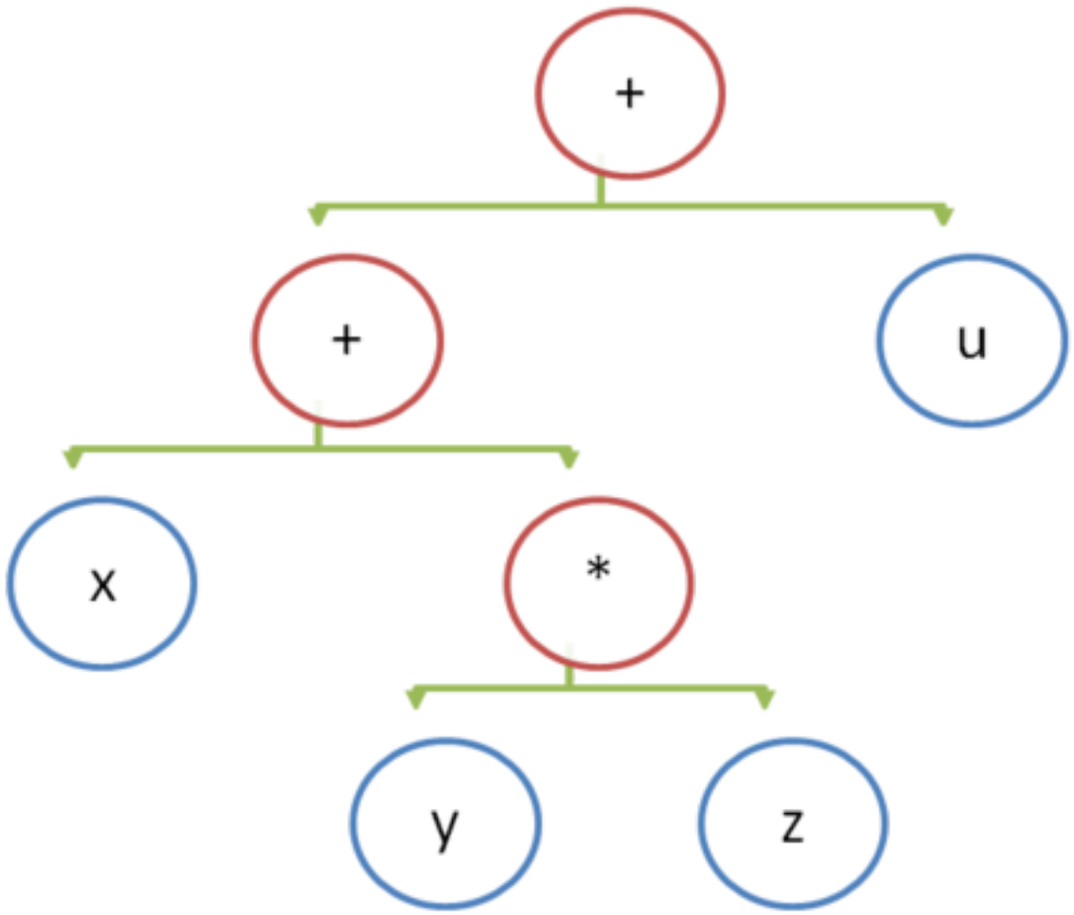
How does Python interpret this:

Stack

```
push x
push y
push z
multiply
add
push u
add
```

~

AST



Python

Python

Pros:

- **Extensive Support Libraries**
- **Can use C/C++ libs via pybind11**
- **Cross platform**
- **Relatively low entry level**
- **Free & open-sourced**
- **Huge community**

Cons:

- **Slower than C/C++, assembly**
- **Runtime errors**
- **Parallel(GIL)**

Python environments

pip - Pip Installs Packages - package manager for python. Can be used together with conda but very carefully

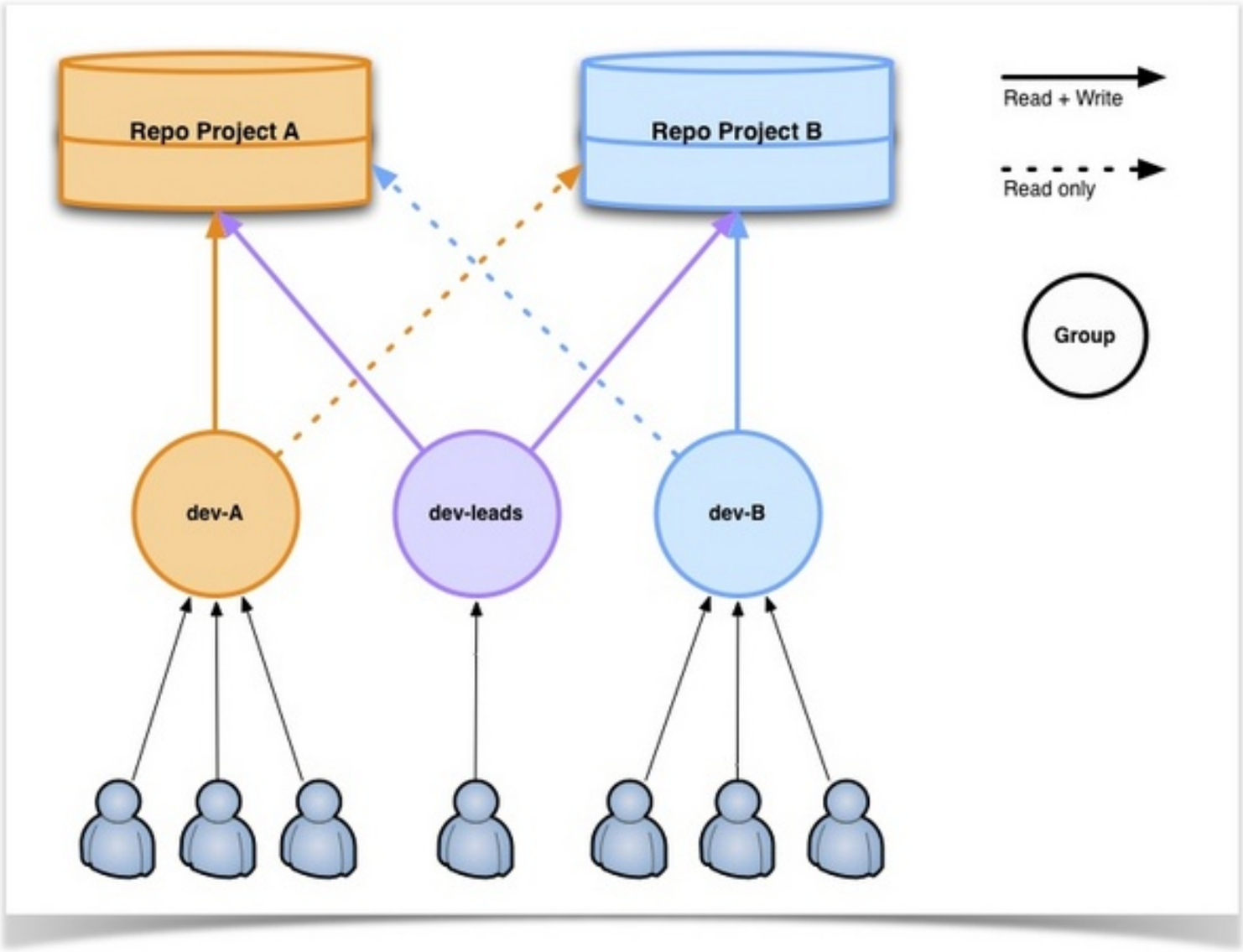
If you use python without anaconda:

- virtualenv, source activate, pip install

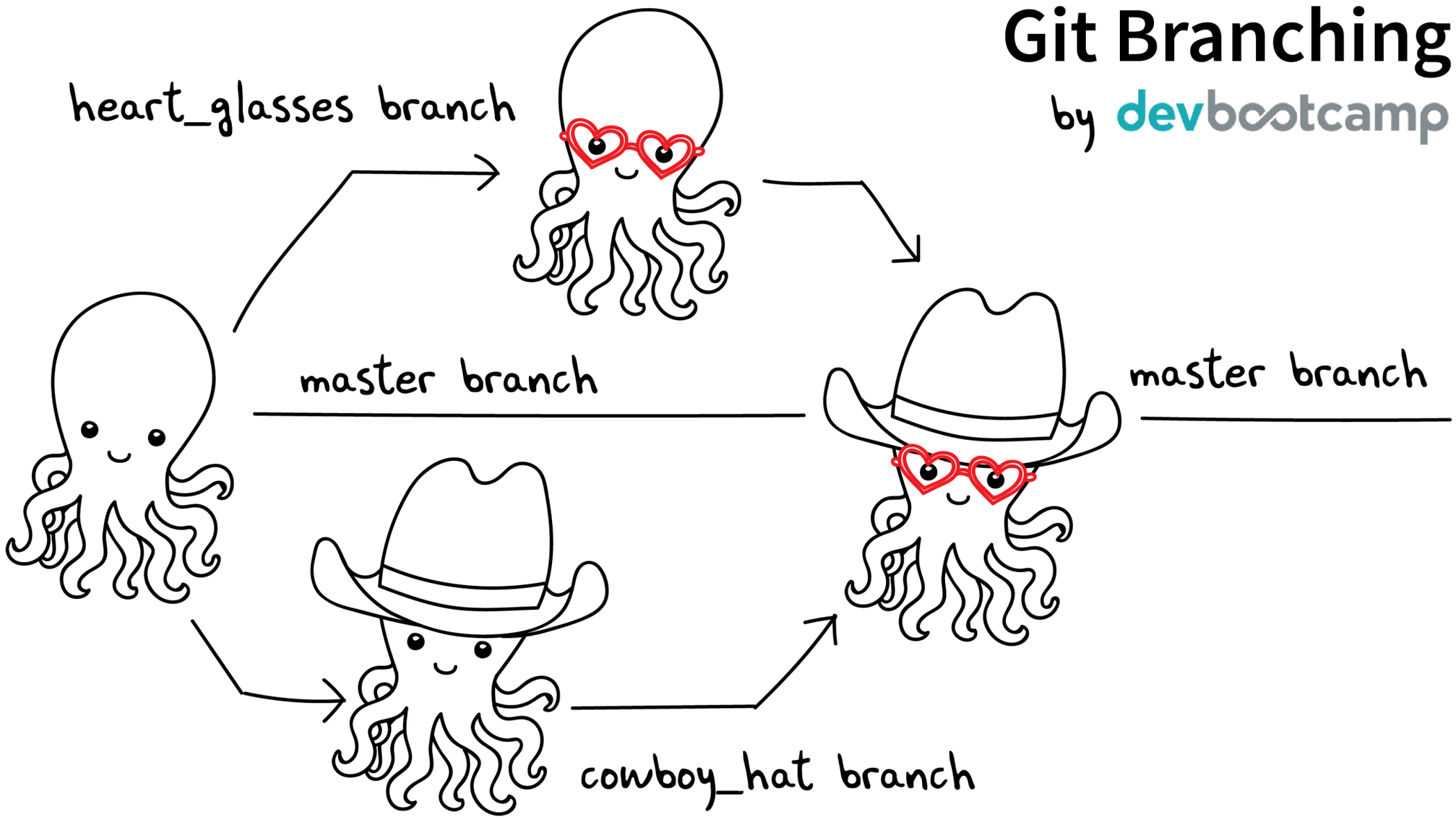
If you use anaconda:

- conda activate, conda install + pip install

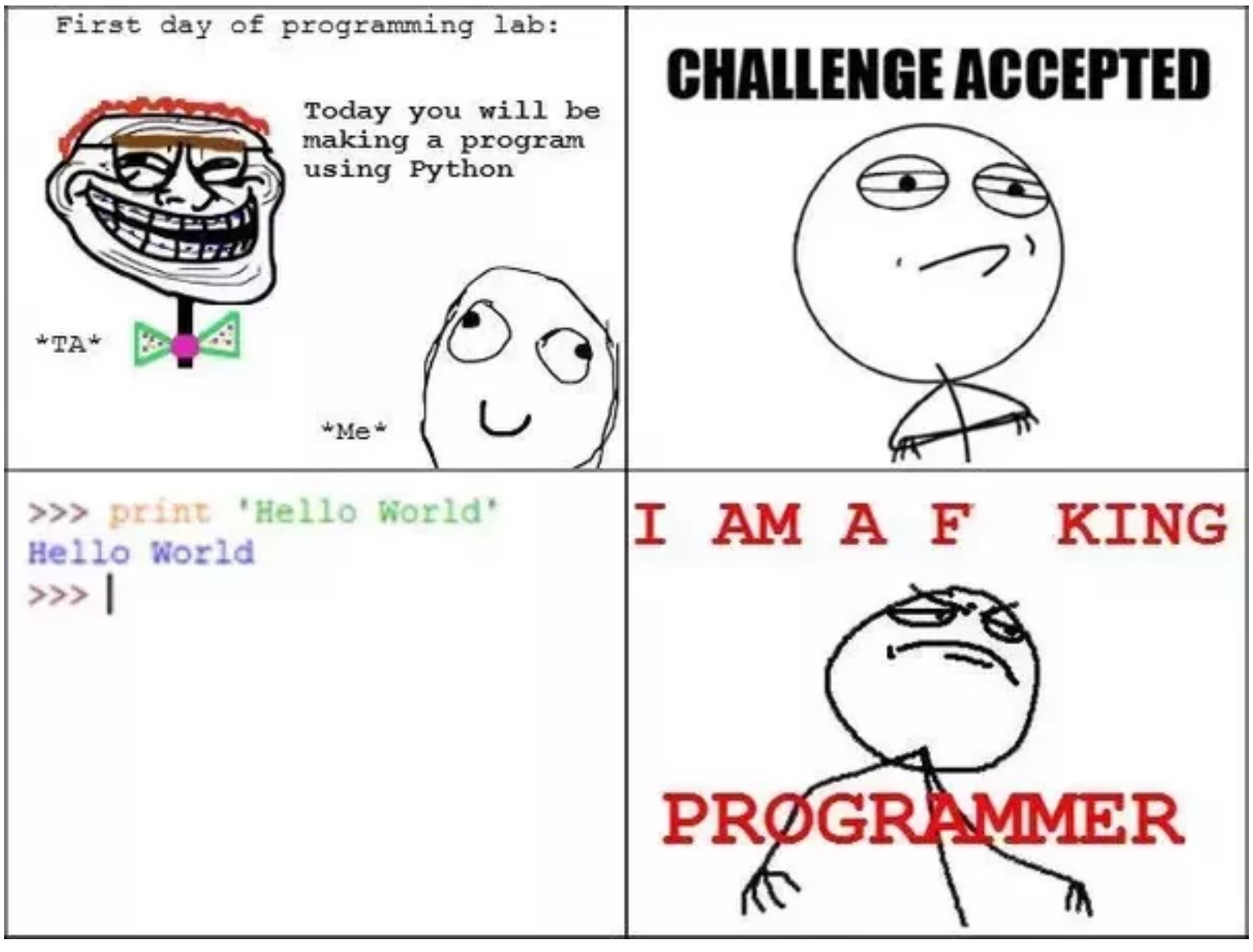
Git: repositories



Git: branches



Let's code


First day of programming lab:
Today you will be making a program using Python
TA
Me
CHALLENGE ACCEPTED
>>> print 'Hello World'
Hello World
>>>
I AM A F KING
PROGRAMMER