Computational Methods for Linguists

Ling 471

Olga Zamaraeva (Instructor)
Yuanhe Tian (TA)

04/06/21

Readings

See syllabus!

- Some papers have **highly technical** parts
- It is normal to not understand all of the paper
  - Even **sholars don't understand everything** when they review
- Identify **specific goals** for yourself when reading
  - e.g.: "I want to understand whether the idea of "data statements" is applicable to the IMDB datasets
- Reading **documentation**
  - e.g. python
  - Boring but sometimes necessary
    - Many problems happen from people not understanding fully what a function is doing exactly
      - Which stems from not reading documentation

Assignment 1

Date	Topic	Reading	Due
March 30	Introduction, course structure, etc.		
April 1	Conceptual and technical overview	What is Data Science?	Online survey (on Canvas); "Assignment 0"
			Request an account on the patas cluster
April 6	Basic system/programming concepts	Basics of python programming	
April 8	Command line vs GUIs: What to use when and how	The IMDB reviews dataset paper	
	Version control (git, GitHub)	Data statements for NLP	Blogs 1
April 13	Basic programming	TBA	Assignment 1

https://olzama.github.io/Ling471/syllabus.html

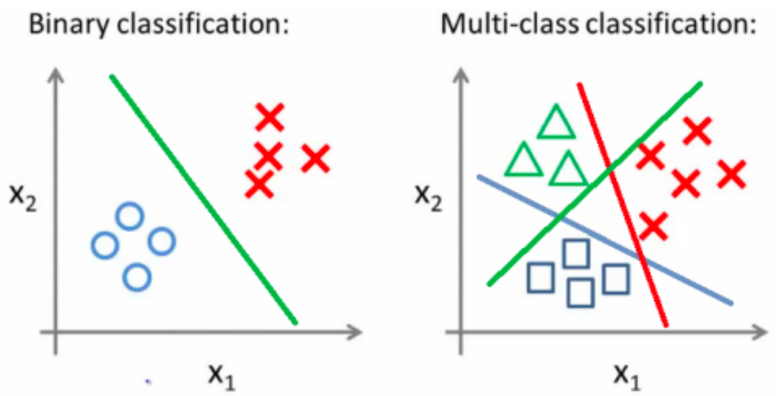
- Due April 13
- Please start!
  - You **can do** some parts of it **already**!
    - Skip the steps you don't know how to do.
  - Finished fast? Great!
  - But don't put yourself in a position when you need to do an impossible amount on the last day
    - also because you'll need to ask questions and wait for ans.
- **Several parts**
  - One **data-**focused, several **setup**-focused

Lecture surveys

Anonymous!

- After each lecture, let me know, via the anonymous surveys on Canvas:
  - Was it over your head?
  - Was it too easy?
    - In both cases, you **didn't** learn much
    - ...and I should adjust the level
  - If it feels like you **learned** things
    - Then the lecture is "just right" and I should keep them at the same level!

NLP
And machine learning



https://medium.com/@b.terryjack/tips-and-tricks-for-multi-class-classification-c184ae1c8ffc

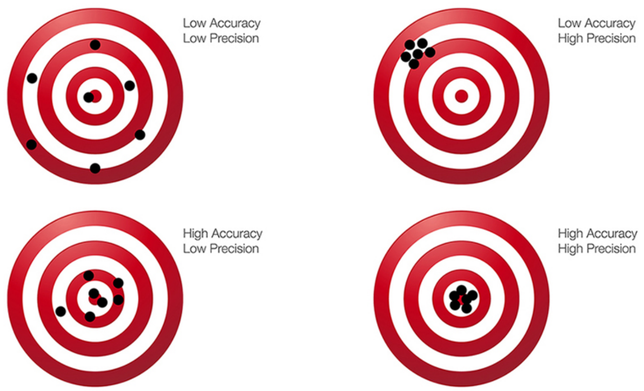
- ML is a study of computer algorithms that improve automatically by getting **feedback**
  - Correct predicton? Keep things as is! Incorrect? Adjust!
  - "Correct" vs. "incorrect" are **labels**
    - ML doesn't know the **meaning** of labels
    - "Correct" = 1; "incorrect" = 0
    - **Some** "labels" can be obtained automatically
      - Deep Learning
- Is NLP a ML/DL field?
  - Today - you could say so. Most work is DL.
  - 10 years ago: no. Other methodologies were also used.
  - 10 years from now?

	aardvark	...	computer	data	pinch	result	sugar	...
apricot	0	...	0	0	1	0	1	
pineapple	0	...	0	0	1	0	1	
digital	0	...	2	1	0	1	0	
information	0	...	1	6	0	4	0	

Figure 6.5 Co-occurrence vectors for four words, computed from the Brown corpus, showing only six of the dimensions (hand-picked for pedagogical purposes). The vector for the word *digital* is outlined in red. Note that a real vector would have vastly more dimensions and thus be much sparser.

Jurafsky & Martin *Speech and Language Processing 3rd edition*

NLP
and evaluation



- NLP (and some other fields) are **shaped** by evaluation
- Evaluation: computing **metrics** which indicate how well the system did on **held-out** data
  - Metrics: Methodically measured results
    - Accuracy
    - Precision/Recall
    - Coverage, etc.
  - Held-out: Not previously seen:
    - by the system
    - AND by the system developer
- Evaluation vs. **Training**
  - Algorithms **adjust** during **training**
  - **No adjustment** should be made during **evaluation**
    - Why?

https://www.rosette.com/blog/evaluating-nlp-assembling-a-test-dataset/

NLP
and evaluation

	aardvark	...	computer	data	pinch	result	sugar	...
apricot	0	...	0	0	1	0	1	
pineapple	0	...	0	0	1	0	1	
digital	0	...	2	1	0	1	0	
information	0	...	1	6	0	4	0	

Figure 6.5 Co-occurrence vectors for four words, computed from the Brown corpus, showing only six of the dimensions (hand-picked for pedagogical purposes). The vector for the word *digital* is outlined in red. Note that a real vector would have vastly more dimensions and thus be much sparser.

- To evaluate systems which make predictions:
  - **Test** data has to be **labeled**
    - **Training** data may or may not need to be labeled
    - Depending on the type of ML used for training
  - Labels are created by people **or** are inherent
    - What's the meaning of a non-human created label?
    - If a non-human-created label was **interpreted** by a human, what does that mean?
- Most "unsupervised" and "fully automatic" NLP systems **rely on humans**
  - Because systems need to be **evaluated**

- Example/exercise:
  - **Training:** Word cooccurrences
  - **System:** Autocorrect

How will you **evaluate** an Autocorrect system?

If you remember just one thing about NLP:

Automatic system evaluation is meaningless without humans

Questions?

Plan for today

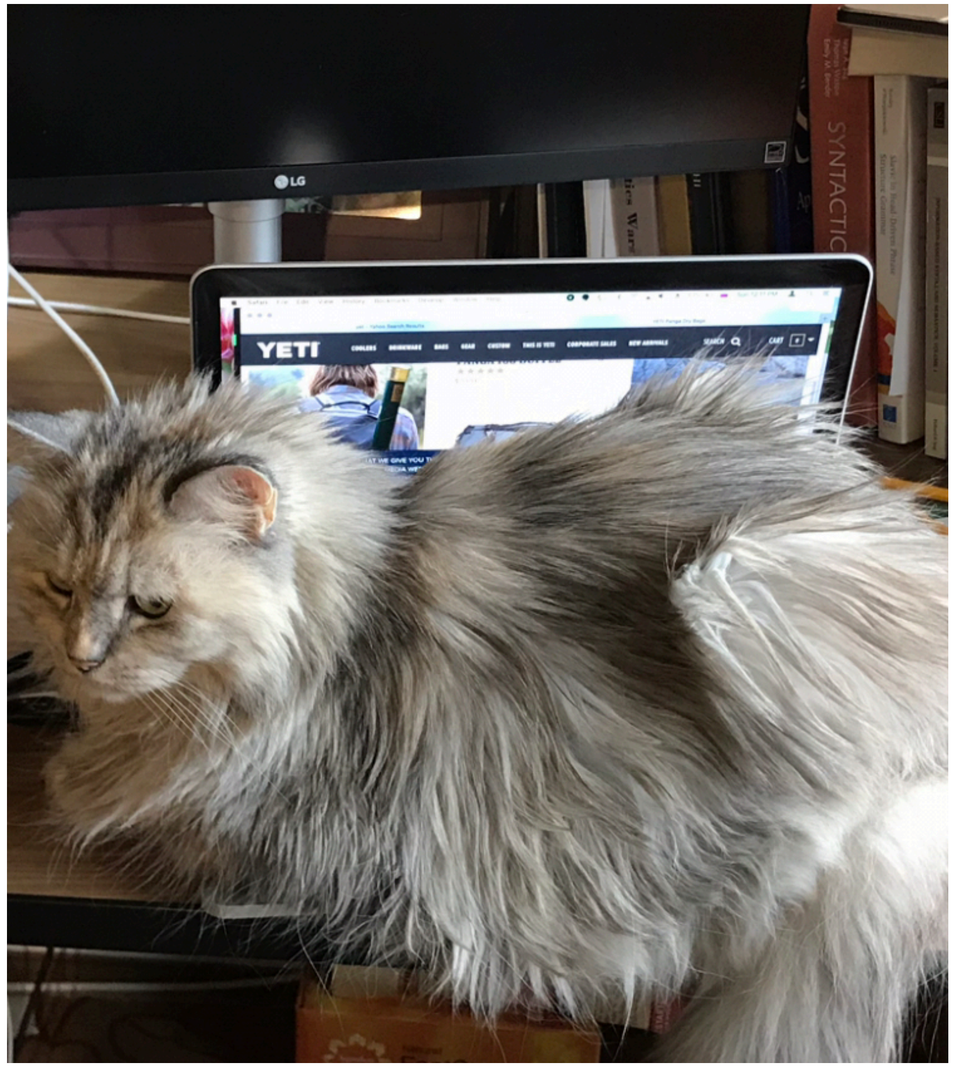
Systems and programming

- Learning about operating systems and software architecture: Why?
- How does programming relate to systems?
- Virtual machines
- Servers
- Remote “clusters”
- Programming languages

Programming

Why we need it?

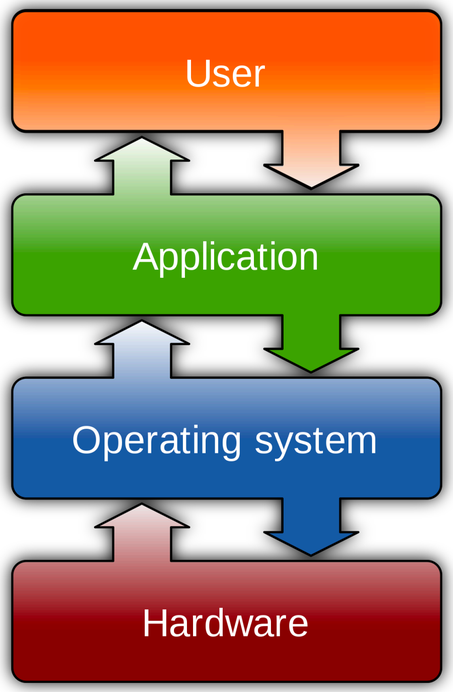
- We are **lazy**
  - Doing things by hand is tedious
  - Processing LOTS of data is impossible by hand
- We make **mistakes**
  - We tend to do things slightly differently every time we repeat tasks
  - Our attention is not predictable
- **Automation solves both problems**



Operating Systems

What are they?

- **Systems** are software which supports **basic functions**
  - E.g. how does a program get executed at the hardware level?
    - Memory allocation, file locations, input and output
  - No program can run without an operating system
  - There is one in any computer and behind any website
    - In case with the website, it is installed on the remote **server**
- Understaing systems is understanding **low-level** software
- We will be programming using **high-level** language
  - Why learn about systems?

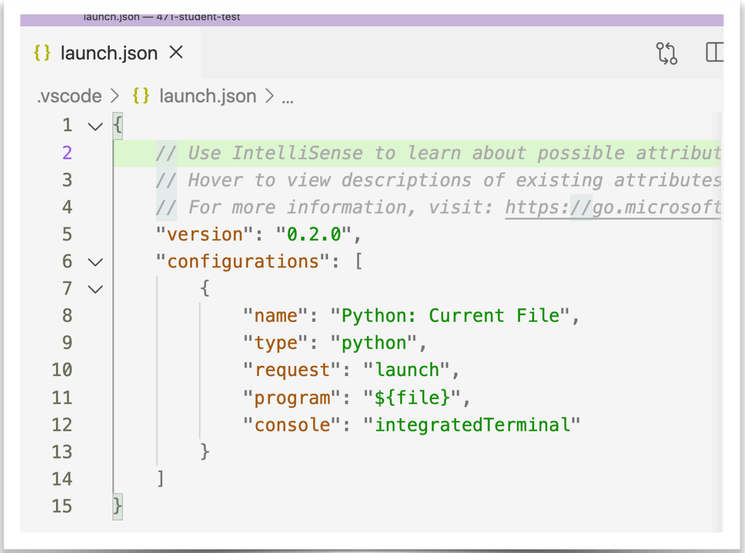


https://en.wikipedia.org/wiki/Operating_system

Systems and configurations

Why we need to learn about them?

- Programming is possible to do in the web browser
  - **High-level** programming
  - **Why** bother learning about **low-level** stuff?
- You can practice **some** programming in the browser
- But data science projects are **pipelines**
- Pipelines work as a **system**
- Putting modules together is one of the **hardest** things in programming
  - It requires instruction, tutoring, and practice



A scary “running configuration” in VS Code

Operating systems

The landscape

- Unix (AT&T/Bell labs)
- MS-DOS (Microsoft)
- Linux (open source, Linux **community**)
  - **unix**-based
- OSX (Apple)
  - **unix**-based
- Windows (Microsoft)
  - **DOS**-based
- Cloud
  - Next step; like remote **clusters** but more user-friendly :)

https://hackr.io/tutorials/learn-operating-systems

Operating systems

What does it mean to be unix/DOS-based?

- In practice:
  - Unix-based systems are common in research and engineering
    - Linux is **free**
    - Considered more **flexible, stable, and secure**
  - DOS-based Windows is common elsewhere
    - Established market
    - Considered more user-friendly

https://hackr.io/tutorials/learn-operating-systems

Operating systems

What does it mean to be unix/DOS-based?

- In practice:
  - Slightly different **command-line** language
    - “Command line” is a way to give your operating system tasks by typing specially formatted text in the terminal
    - ...vs. clicking on buttons and windows (“GUI”)
    - **Stay tuned** for next lecture
  - Different text file format
  - Different file path separator (in command line)
  - Different **applications** installled/written

Operating systems

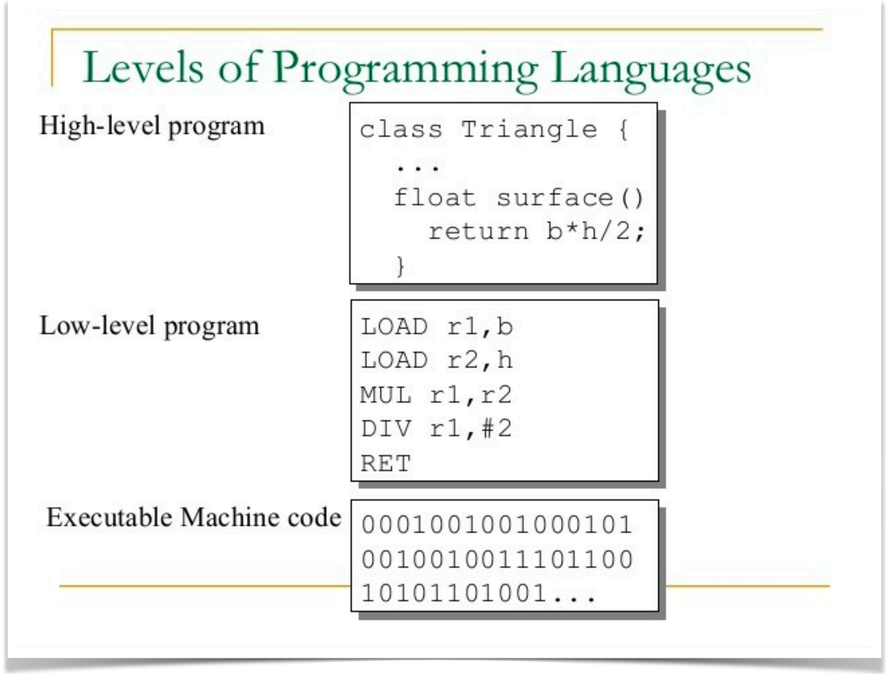
What does it mean to be unix/DOS-based?

- In practice, for this class:
  - Minimal differences
  - When I show you something, it's mostly on OSX
  - You should be able to do the same on your system
    - Occasionally, it means googling/asking on the Discussion Board, "how do I do X on Windows?"
  - You may need to do some things on the remote **Linux cluster**
    - Things will look **similar** to your Windows command prompt but you will need to think more **low-level**

High-level and low-level

How are they related?

- A high-level program is just **text**
  - Conforming to certain syntax and composed of valid keywords
- In order for the program to be executed:
  - There needs to be an **interpreter** or a **compiler**
  - ...installed on the operating system
  - It will ultimately translate the program into **binary code instructions** for the computer
    - ...which the computer can **execute**
    - ...by switching the state of transistors **on/off**

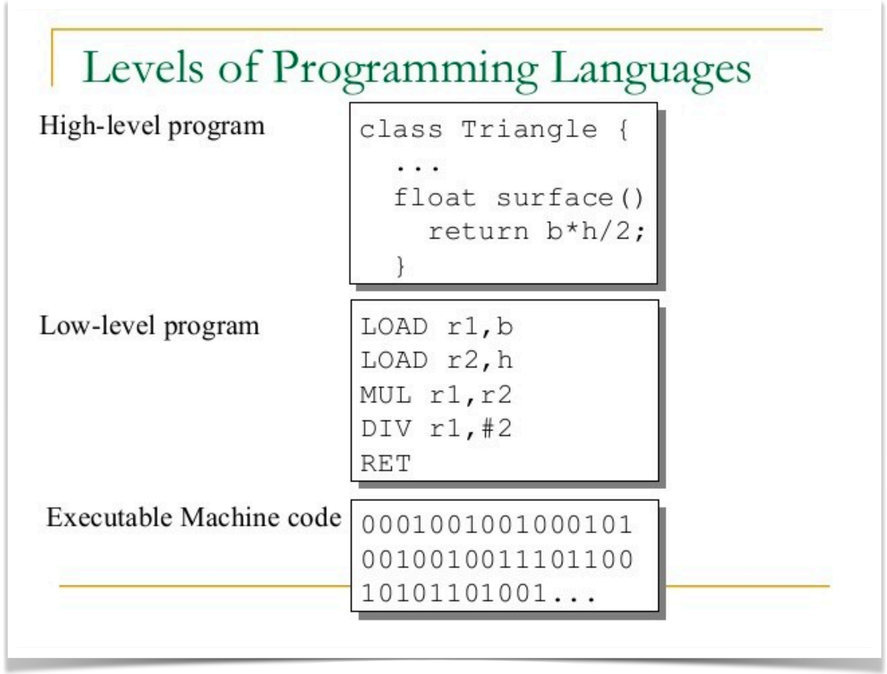


https://www.educba.com/high-level-languages-vs-low-level-languages/

High-level and low-level

How are they related?

- A high-level program consists of **parts**
  - **Pre**-built “Libraries”, “modules” **plus** your code
  - Without pre-built libraries, you’d need to write everything from scratch
    - **Including the compiler/interpreter**
    - (that would really suck!)
- The computer needs to know **exactly** where all the parts are
  - Setting this up is a **big** part of programming



https://www.educba.com/high-level-languages-vs-low-level-languages/

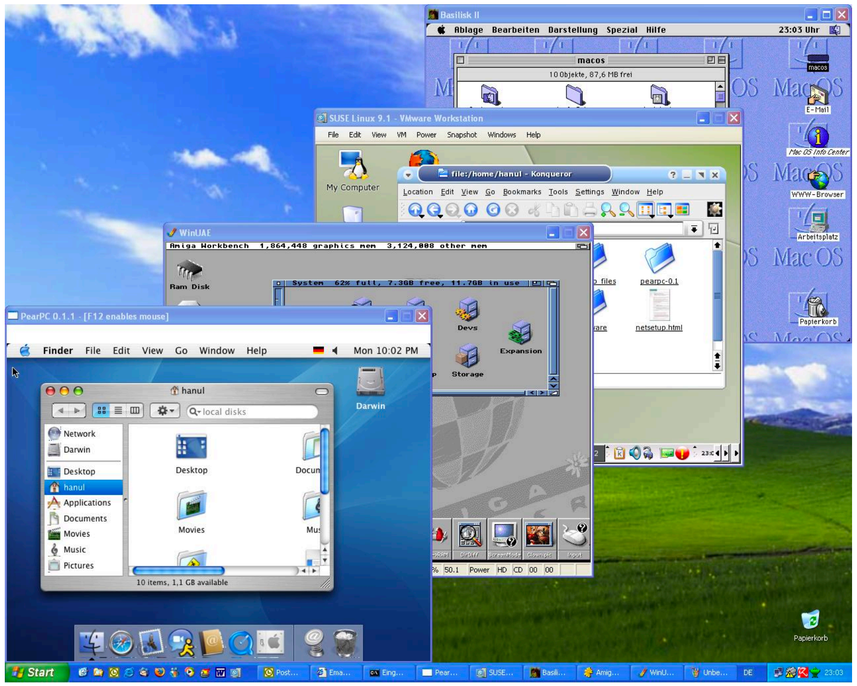
Juggling OSs

- Virtual machines
  - Good for small-to-medium tasks and casual gaming ;)
  - “Docker” containers
- Remote machines
  - May be necessary to access a particularly powerful machine you otherwise cannot afford
  - Our **patas** cluster
  - Amazon Web Services
    - Why isn’t everyone using AWS?

https://hackr.io/tutorials/learn-operating-systems

Virtual machines

- Your **host** OS dedicates some space on the disk
  - ...where it then installs **another, “guest”** OS
  - ...using special software
    - VirtualBox (free), VMWare (for $)
- Demo!
  - P.S.: You aren’t likely to need a VM for this class, but it’s important to not be afraid of them
  - You need **hard disk space** for VMs

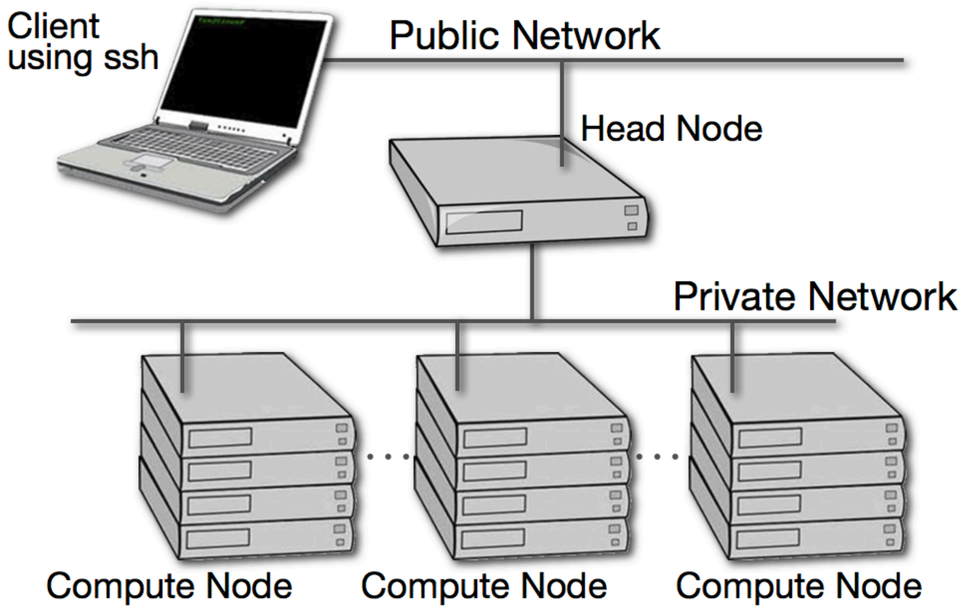


https://techgenix.com/virtual-machines/

Remote servers

- You can connect to a computer via net:
  - A computer with the “right” OS
  - ...with lots of memory/space
  - ...with powerful processors

- PS: We may or may not need remote computers in this class, but they are unavoidable in data science/NLP
  - Our **patas** is a remote cluster

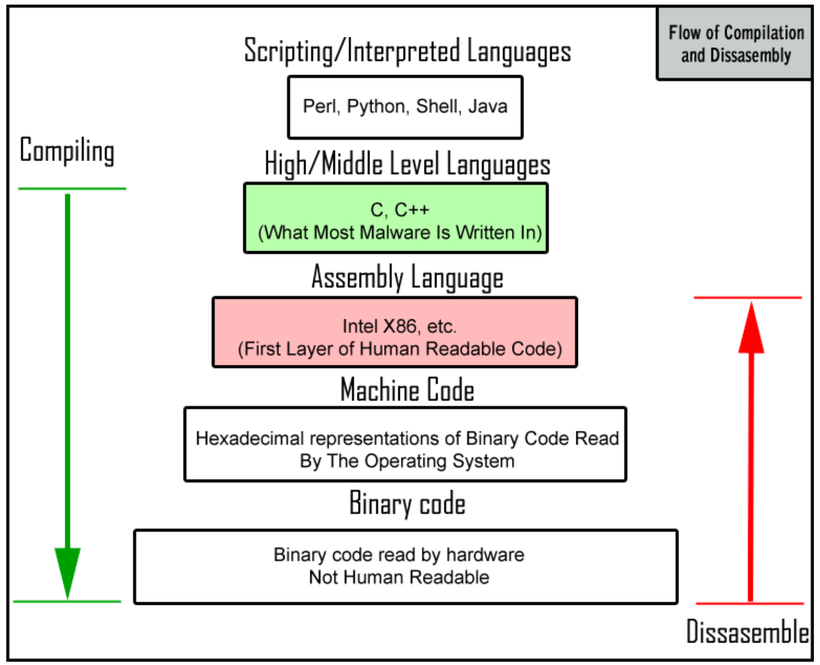


http://www1.udel.edu/it/research/training/config_laptop/linuxAgent.html

Questions?

Programming languages

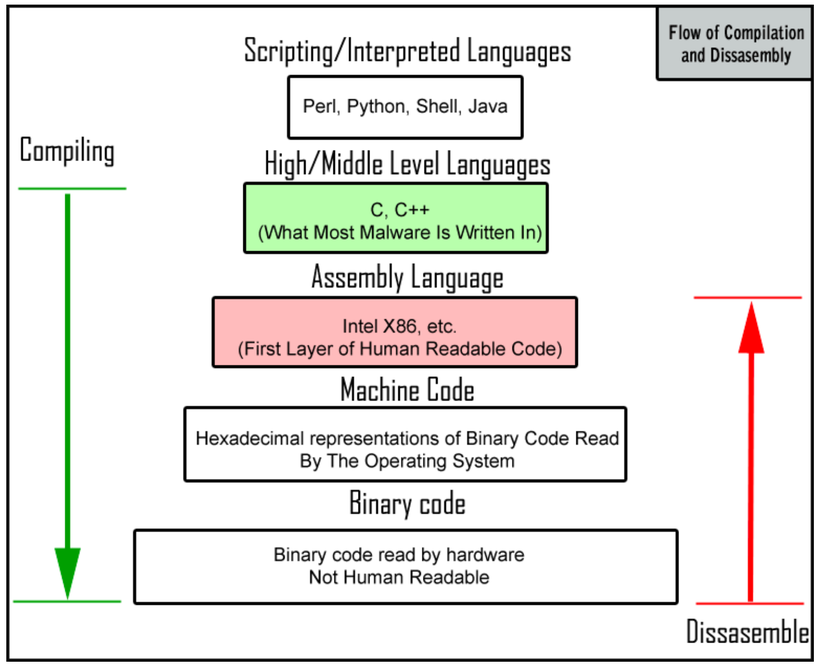
- We will be using **python**
- Other languages: C/C++, C#, Java, Ruby, Haskell, Javascript, GO, R, Mathlab…
- Languages can be more or less **suited** for a task
  - …and more or less **popular**
    - …**regardless** of how well suited they are for smth!



https://blog.malwarebytes.com/security-world/2012/09/so-you-want-to-be-a-malware-analyst/

Types of programming languages

- Compiled
  - Whole program is converted to binary
  - ...then executed
  - C/C++, C#, Java and other
- Interpreted
  - Each statement is executed in real time
  - **Python** and other
- Imperative/Procedural
  - Sequences of statements lead to results
  - Program may have different **state** at different point
- Declarative/Functional
  - A program is similar to a mathematical formula (that leads to a result)
  - Program state usually not stored



https://blog.malwarebytes.com/security-world/2012/09/so-you-want-to-be-a-malware-analyst/

Types of programming languages

Olga's opinion:

- Interpreted and procedural languages
  - Are **easier** to **learn** for more people
  - No need to deal with compiling errors
- Compiled and functional languages
  - Are **harder** to write **buggy** programs in
  - The compiler would've **caught many** of the bugs!
- Python: Interpreted and procedural
- Olga's "language of choice": C#
  - Compiled and procedural
- Functional languages have limited appeal/use
  - ...because people are usually **better at imagining steps than at imagining formulas**!

```java
public class HelloWorld {

    public static void main(String[] args) {
        // Prints "Hello, World" to the terminal window.
        System.out.println("Hello, World");
    }

}
```

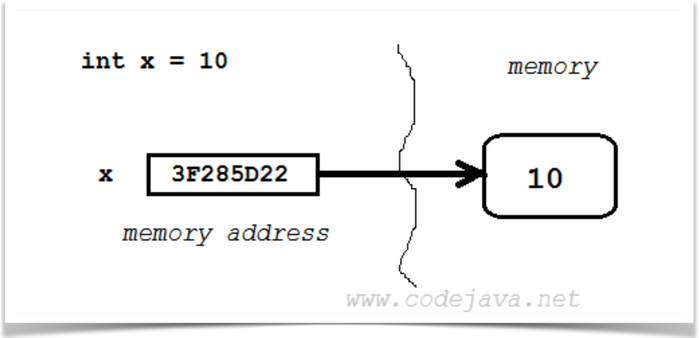
A "hello world" program in Java

```python
print ("Hello world!")
```

A "hello world" program in Python

Program state and variables

- Variables
  - Represent memory/storage locations
- State
  - The content of all variables at a given point
- Variables contain values of a certain **type**
  - e.g. integer, string…
- Variables may or may not **require** that the value be of a certain type
  - Python does not require it
    - Any variable can store any type of value at different points
  - Java does require it
    - A variable declared for integers will not be able to store strings, etc.



https://www.codejava.net/java-core/the-java-language/java-variables-passing-examples-pass-by-value-or-pass-by-reference

Statically, strongly-typed languages

Reduce bugs

- Declare variable as storing strings
- Try storing an integer there
  - Problem caught **before execution**
  - In weakly- and dynamically typed languages:
    - Problem caught **during execution**
    - ...which may be two years into production

```python
ints = {1,2,3}
s = "Hello world!"

def countChars(s):
    return len(s)

count = countChars(ints)
print(count)
```

Python is happy to execute this ^^

```java
public class hello {
    Run | Debug
    public static void main(String[] args) {
        // System.out.println("Hello cat");
        String s = "Hello world!";
        int[] ints = {1, 2, 3};
        int count = countChars(ints);
        System.out.println(count);
    }

    public static int countChars(String s) {
        return s.length();
    }
}
```

Java won't execute this (note the red squiggles)

Dynamically typed languages

Are better fore data science

- Data science:
  - Data can be seen as **generic**
  - Often, don't **want** to know what type you will get!
- So data science is an overwhelmingly **dynamically** typed field/practice

```python
ints = {1,2,3}
s = "Hello world!"

def countChars(s):
    return len(s)

count = countChars(ints)
print(count)
```

Python is happy to execute this ^^

```java
public class hello {
    Run | Debug
    public static void main(String[] args) {
        // System.out.println("Hello cat");
        String s = "Hello world!";
        int[] ints = {1, 2, 3};
        int count = countChars(ints);
        System.out.println(count);
    }

    public static int countChars(String s) {
        return s.length();
    }
}
```

Java won't execute this (note the red squiggles)

Python

What should we know about it?

- Interpreted, dynamically typed…
- Relies on **modules** aka **packages**
  - Python beginners suffer the most from not being able to **import packages correctly**
- Comes in different **versions**
  - 2.* (obsolete but still around)
  - 3.0—**3.9**
  - Which version is your computer running?
  - Which version is patas running?
  - Possible to specify which version you want
- **Next: Running programs: IDE and command line. Source/version control (git). Virtual environments**

```python
ints = {1,2,3}
s = "Hello world!"

def countChars(s):
    return len(s)

count = countChars(ints)
print(count)
```

Questions?