Python for Language Processing

(2b) Algorithms

Dr. Jakob Prange

Fakultät für Angewandte Informatik - Universität Augsburg

CL Fall School 24

Credit: This course is based on material developed by
Annemarie Friedrich, Stefan Thater, Michaela Regneri, and Marc Schulder at Saarland University

What is **Programming**?

- Programmer wants to solve a problem in a systematic way

Example

Wash clothes using washing machine.

Example

Find the largest number in a list of numbers.

```
list = [1, 5, 3, 7, 2, 4]
```

What is **Programming**?

- Algorithm = abstract, detailled computing instruction that solves the problem (‘recipe’)

Example

WASH CLOTHES USING WASHING MACHINE

1. Load the laundry
2. Add detergent and additive
3. Switch on the machine
4. If clothes are wool: select wool program
   otherwise select normal program
5. Start the program
6. Wait until done
7. Remove clothes

What is **Programming**?

- Algorithm = abstract, detailled computing instruction that solves the problem (‘recipe’)

Example

ALGORITHM: Find maximum number of a list.

1. Remember first number in `list` as maximum[a]
2. Check each number from the 2nd to the last number in `list`:
   1. Compare number with current maximum
   2. If the number is greater, change maximum to be this value
3. Result (maximum of the list) is the recorded maximum after checking all numbers.

[a] Ignores the special case of an empty list.

What is **Programming**?

- An algorithm can be executed with different inputs

Example

Should wash different laundry correctly.
wool, normal, towels, curtains,...

Should find the maximum of any given list.

```
aList = [1, 2, 5, 7, 9]
anotherList = [5, 2, 29, 0]
```

- Algorithms can terminate with an output

Example

Return a message showing that the clothes are washed now - or failure.

Return the maximum number of the list.

What is **Programming**?

- An algorithm can be executed with different inputs

Example

Should wash different laundry correctly.
wool, normal, towels, curtains,...

Should find the maximum of any given list.

```
aList = [1, 2, 5, 7, 9]
anotherList = [5, 2, 29, 0]
```

- Algorithms can terminate with an output

Example

Return a message showing that the clothes are washed now - or failure.

Return the maximum number of the list.

What is **Programming**?

- Program = realization of the algorithm in a specific programming language

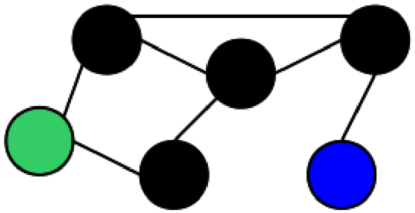
```python
maximum = aList[0]
for i in range(1,len(aList)):
    if maximum < aList[i]:
        maximum = aList[i]

print(maximum)
```

Problems

Some Problems:

- Computation of arithmetic functions
- Find the shortest path in a graph
- Computational Linguistics: morphological analysis, tagging, parsing,. . .



Problems & Subproblems

Problems can be decomposed into subproblems.

Example

Problem: Find greatest number in `list`.
Subproblem: Find maximum of two numbers.

Algorithms can provide ways of solving subproblems.

Can you think of examples for problems & subproblems?

Algorithms

- ‘Recipe’ for solving a problem.
- Should work for all inputs of the problem.
- Must terminate in a finite number of steps.
- Granularity of the steps: Recipe must be defined clearly.
  Depends on audience / programming language it is designed for / . . .
- There may be more than one algorithm per problem.
- Efficiency = measured in time and / or memory usage (often trade-off).

Programming Languages

- CPU only understands machine language instructions (0110101101. . .)
  $\Rightarrow$ inconvenient for humans
- Programming languages
  - hide complexity of machine language
  - provide more abstract constructs
  - make programmer-machine interaction efficient
- Different programming languages
  - are designed for different use-cases
  - support different programming styles (e.g. procedural / object-oriented)
  - have different advantages and disadvantages

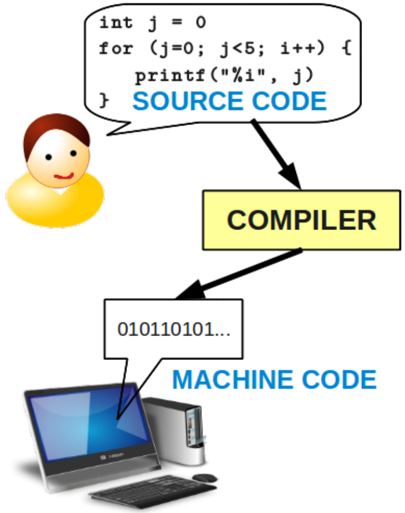
Python

- Object-oriented programming language.
- Can also be used as a procedural scripting language and (partially) for functional programming.
- More about Programming Paradigms later!

History

- Successor of ‘teaching language’ ABC
- Development started in late 80’s
- For ambitious users without programming skills
- Focus on easy file handling

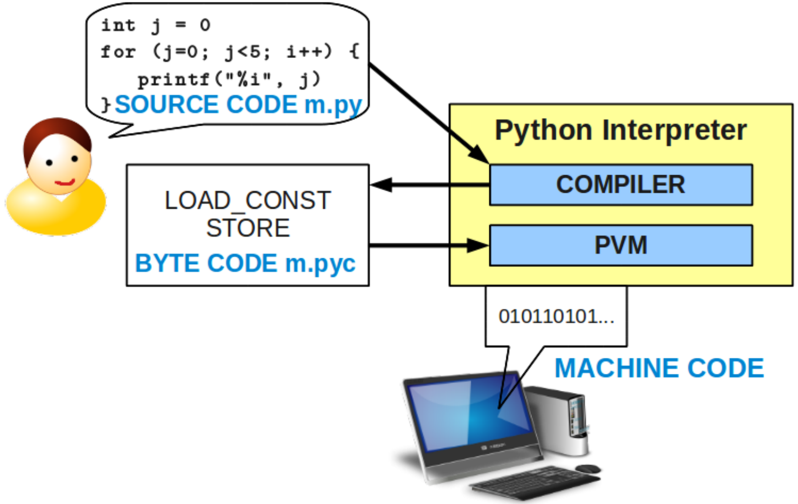
Compilation



```
int j = 0
for (j=0; j<5; i++) {
   printf("%i", j)
}
```

- Compiler = program that translates source code into *machine code* during *compile time*
- CPU executes this machine code during *runtime*
- e.g. C is a *compiled language*
- Advantage: once compiled, programs run very fast
- Machine code is specific for a platform (e.g. Linux/Windows)

Interpretation

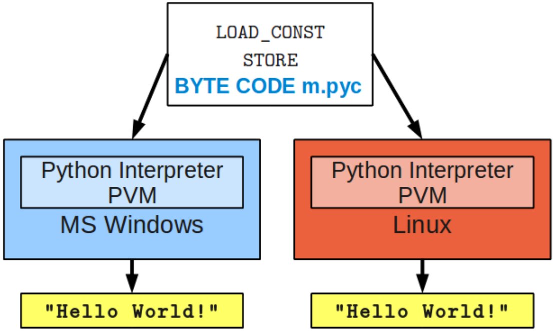


```
int j = 0
for (j=0; j<5; i++) {
   printf("%i", j)
}
```

Interpretation and Platform Independence

- Interpreter = program that executes the source code per command. Enables interaction between system and programmer.
- Python is compiled into byte code, which is interpreted by a Python Virtual Machine (PVM = runtime environment).

- The virtual machine implementation is specific for the platform.
- Your Python code can run on any platform.



Advantages of Python

- simple syntax
- very flexible - little is forbidden, much is convention.
- easy file handling
- full unicode support
- handles arbitrarily large integers
- convenient support for regular expressions
- handy toolkits for data science

Highly Recommended Reading

- **The Semicolon Wars** by Brian Hayes
  - *Every programmer knows there is one true programming language. A new one every week.*
  - *Jeder Programmierer weiß, dass es nur eine einzig wahre Computersprache gibt. Jede Woche eine neue.*
- `https://www.americanscientist.org/article/the-semicolon-wars`
- Deutsche Version: “Brian Hayes: Der Strichpunkt-Krieg (Spektrum Wissenschaft)”.

Literature

- Mark Lutz: **Learning Python (Animal Guide)**, 4th edition, 2009, O'Reilly
  Standard Python Reference Book, comprehensive
- Michael Dawson: **Python Programming for the absolute beginner**, 3rd edition, 2010, Course Technology / Cengage Learning
  Excellent for absolute beginners, good explanations, as fun to read as a programming book can be ☺
- Steven Bird, Ewan Klein, Edward Loper: **Natural Language Processing with Python**, O'Reilly Media 2009
  Explains the toolkit NLTK
  available online: `http://www.nltk.org/book`