

Programming languages

Silvio Peroni

✉ silvio.peroni@unibo.it  0000-0003-0530-4305  @essepuntato

Computational Thinking and Programming (A.Y. 2017/2018)

Second Cycle Degree in Digital Humanities and Digital Knowledge

Alma Mater Studiorum - Università di Bologna



Creative Commons Attribution 4.0 International License

Communication 1

Wednesday the 6th of December there will be the introduction to the project specifications and rules

In addition to that, the same day (the 6th of December) there will be also the official assessment of the course run by the University

Communication 2

As already announced in the mailing list, we will have a lecture on Thursday the 7th of December, from 15:30 to 17:30

Any question about the previous
lecture?

Historic hero: Grace Hopper

He was a computer scientist

The first programmer of the Harvard Mark I - another general purpose electromechanical computer used during the WWII

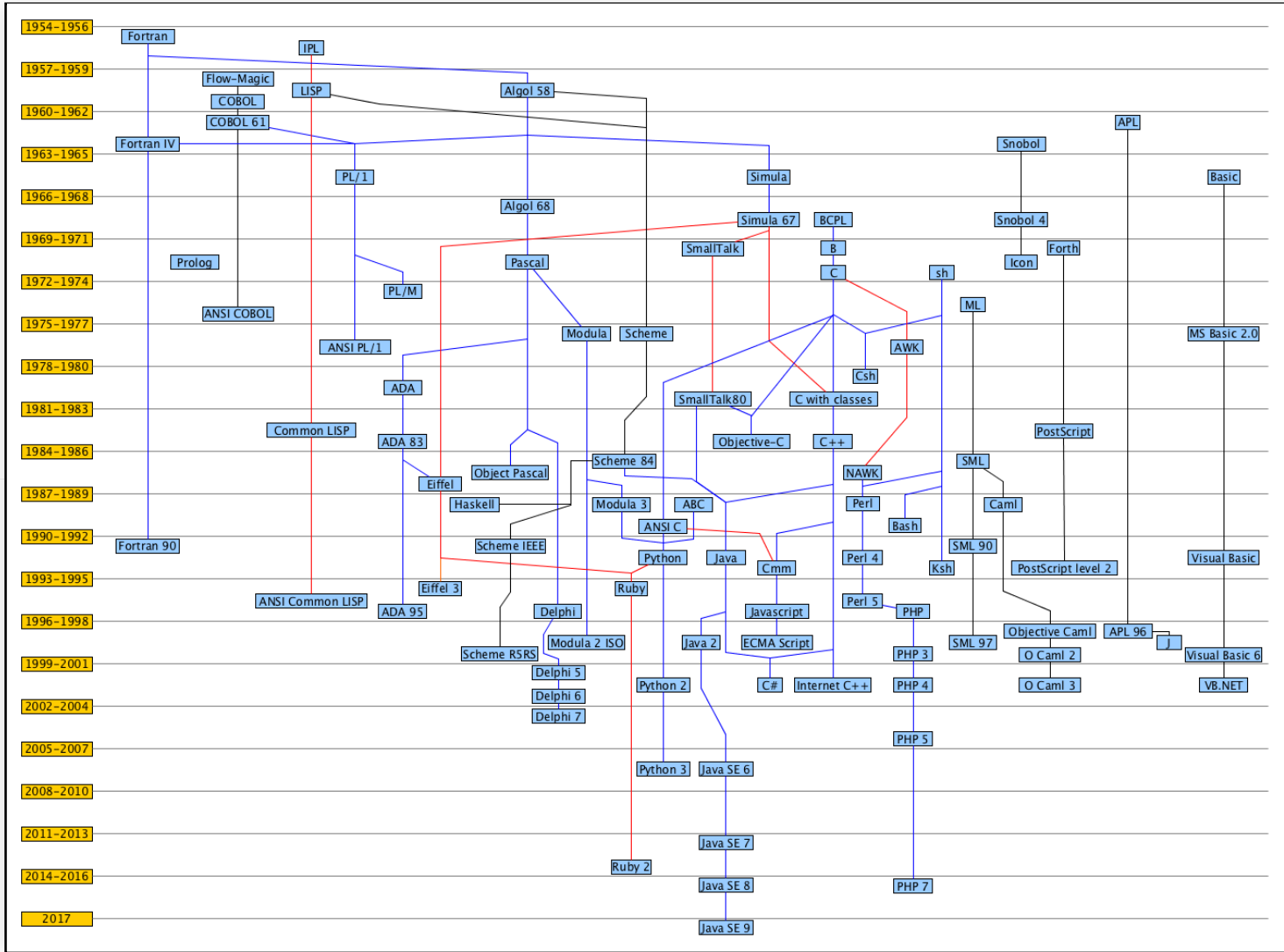
Main developer of COBOL, one of the first high-level programming languages

COBOL adopted English for commands:

`x IS GREATER THAN y`



A timeline



From ThyMopani to Python

t h y m o p a n i

5 6 4 2 7 3 1 8 0

i a m p y t h o n

Python

It is an high-level programming language for general-purpose programming

It is currently one of the most used languages for programming in the Web, for Data Science and Natural Language Processing tasks

It will be introduced with more details during the laboratory sessions

Play with Python: <https://repl.it/>

Functions

Definition of algorithms:

```
def <algorithm>(<param_1>, ...)
```

Actually it is the mechanism for implementing functions in Python, i.e. a mechanism for listing a sequence of instructions under a particular name

- *Built-in* functions: made available by the programming language itself, e.g. `len` or `list`
- *User-defined* functions: written by a user of the language for addressing some specific requirements or tasks that are not addressable by means of one built-in function directly, like the algorithm implemented

Executions

```
def add_one(n):  
    return n + 1  
  
result = add_one(41)  
print(result)
```

`def print(<object_1>, <object_2>, ...)` allows one to print to the screen a particular value (that can be referred by a variable)

Classes, objects, methods

Class: extensible template for creating objects having a certain type (e.g. the class for strings, for integers, for lists)

Object: a value of a certain kind (e.g. "a string", 42, `list([1, 2, 3])`)

Method: a function that can be run only if directly called via an object (e.g. `<list>.append(<item>)`)

Immutable and mutable values

type	immutable	mutable
string	x	
integer	x	
float	x	
boolean	x	
None	x	
set		x
dictionary		x
tuple	x	
list		x
deque		x

Immutable by value

```
def add_one(n):  
    n = n + 1  
    return n  
  
my_num = 41  
print(my_num)    # 41  
  
result = add_one(my_num)  
print(my_num)    # 41  
print(result)    # 42
```

Mutable by reference

```
def append_one(l):  
    l.append(1)  
    return l  
  
my_list = list()  
my_list.append(2)  
print(my_list)  # list([2])  
  
result = append_one(my_list)  
print(my_list)  # list([2, 1])  
print(result)   # list([2, 1])
```

Modules and packages

Module: a Python file (extension .py) that contains the definition of variables, functions, and even runnable code

Package: a mechanism to expose one or more Python modules

```
from collections import deque



my_stack = deque()
my_stack.append(1)
my_stack.append(2)
pop_el = my_stack.pop()

my_queue = deque()
my_queue.append(pop_el)
my_queue.append(3)
print(my_queue.popleft()) # 2
```

END

Programming languages

Silvio Peroni

✉ silvio.peroni@unibo.it  [0000-0003-0530-4305](https://orcid.org/0000-0003-0530-4305)  [@essepuntato](https://twitter.com/essepuntato)

Computational Thinking and Programming (A.Y. 2017/2018)

Second Cycle Degree in Digital Humanities and Digital Knowledge

Alma Mater Studiorum - Università di Bologna