# Lecture 1 Computation in Python

Benay Tomas

April 9, 2024

## What does a computer do?

#### Fundamentally:

Performs calculations, a billion per second!

#### What kinds of calculations?

- Built-in of most languages like subtraction, addition, multiplication and division.
- Ones you define as the programmer.
- Others you might import made by others.

#### Types of Knowledge

- Declarative Knowledge: statements of facts.
  Ex: Simon did that.
  In programming this would be the variables.
- Imperative Knowledge: command, instructions.
  Ex: Simon, do that.
  In programming this would be the expressions and calculations that generate variables and results.

#### What is a recipe?

A sequence of steps with a flow control process that specifies when each step is executed and has a mean of determining when to stop.

Recipes are also called algorithms.

#### Stored Program Computer

Sequence of instructions stored inside computers. Built from predefined set of primitive instructions.

1. Arithmetic and logic. 2. Simples tests. 3. Moving data.

Special programs executes each instruction in the order you planned, if you declare a variable after using it, the computer won't understand, so you need to do everything you need on a specific order.

#### Aspects of Languages

- Primitive Constructs: The most basic aspect of languages, in English it's the words. In programming it's numbers, strings and operators.
- Syntax: Another aspect of languages is the Syntax, this will define the difference every language. If you use the incorrect syntax the interpreter won't be able to understand what you're trying to do.
- Semantics: Is which syntatically expressions also have meanings. For instance you can do the expression:
  3 + 'a' and this is syntatically correct because there's 2 objects and 1 operator between them, but it doesn't make sense.
  3 + 5 is syntatically correct and have semantics sense.

#### Python programs

A program is a sequence of definitions and commands. Commands are executed by the Python interpreter. To make programs you can use IDE's there you may use a Shell or the Editor. Shell is used for more basics and short programs, the Editor is used for the opposite.

#### Objects

Python is a programming language object oriented. This means everything is an object from a class, and every class has a series of methods and atributes.

- Programs can manipulate data objects.
- Objects have a type that defines the kinds of things programs can do to them.
  Integers can make mathematical operations.
  Lists can have multiple single values.
- Objects can be:
  Scalar and non-Scalar. Scalar objects are objects that are associated with just one value. While non-scalar objects can be associated with multiple values, like Lists.

**Scalar Objects**

- int: represent integers, ex: 5.
- float: represent real numbers, ex: 3.42.
- bool: represent boolean values True and False.
- NoneType: represent no type, None.
- *type*(): shows the type of an object.

Its possible to convert objects of one type to another.
$float(3)$ = converts int 3 to float 3.0.
$int(3.9)$ = truncates float 3.9 to int 3.

to show an output from code to user, there's the *print*() command:
$print(3 + 5)$
output: 8

**Expressions**

- Combine objects and operators to form it.
- An expression has a resultant-value, which has a type that depends on its objects and in case of the division it will always turn into float.
- Syntax for a simple expression: $< object >< operator >< object >$

### 0.1 Operators

- i + j = sums i and j.
- i − j = subtracts j from i.
- i * j = multiplies i with j.
- i / j = divides i with j.
- i // j = rounds the division with the nearest whole number.
- i % j = returns the remainder of i/j.
- i ** j = i to the power of j.
- There's also the logic operators but we'll see them later.

**Binding variables and values**

We use the equal sign '=' to represent assignment of a value to a variable. In this format: $< variablename > = < value >$

For example: $pi = 3.14159$

The value 3.14159 will be associated with the variable $pi$ and it will be store in computer memory. By typing $pi$ we will be able to use the value whenever needed in the program. We can change the value if needed too. But if used in an expression before changed, the expression will have to be rewrite after the changes, if not the old value will still be in use.