

## Lecture 7 - Introduction to Python

\* prayer spiritual thought.

\* unit of the day:  $1 \text{ BTU} = 1055.06 \text{ J}$

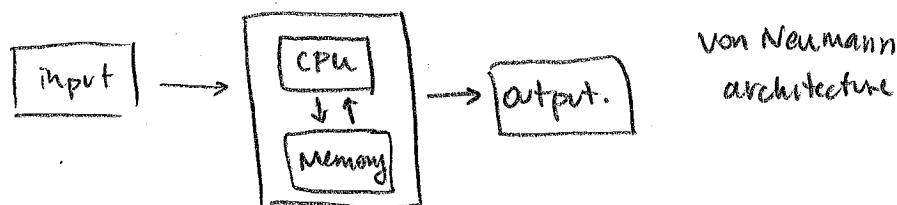
$$\begin{array}{ccc} \uparrow & & \uparrow \\ \text{British thermal} & & \text{Joule} \\ \text{unit} & & \end{array}$$

\* energy unit

\* BTU = raise some mass of water (anirduois)  
 $1^\circ\text{F}$  at  $P=1 \text{ atm.}$

## I. Introduction to programming

\* what is a computer?



input: mouse, keyboard, etc

output: screen, printer, etc.

CPU: control unit - instructions. e.g. get

arithmetic unit - add, subtract, multiply, divide  
 and, or, not.

Memory: information storage.

volatile memory: (RAM) goes away when off.

non-volatile: stays when power is off (hard disk)

- All computers do the same things! Some have more memory or faster CPUs.

\* what is a programming language?

- A program is a series of instructions that ultimately is sent to the CPU to deal with input/output, read/write to memory or do arithmetic.
- A programming language is a codified way of writing & interpreting such instructions.
- All languages have four elements
  - input & output capabilities
  - arithmetic & logic capabilities
  - control structures, e.g. loops & conditionals
  - abstractions, e.g. functions & classes

Analogy: A program is like a river.

- the source is input
- the mouth is output
- Arithmetic / logic are like mills. They are what we do at the river
- Control structures are like dams & canals. used to control the flow
- Abstractions are like the plans for the mills & dams. they help us organize arithmetic & control structures.

Activity: Install python, Hello world

\* Spend 5-10 minutes trying to start Spyder.

## II. Python syntax

\* A programming language is defined by its syntax.

Syntax are the words & symbols we use to express the language.

\* Syntax consists of keywords & operators

- keywords express actions like print (output), or for (control)

- operators manipulate the values of operands  
e.g. + (add), or (logical)

- we will see more examples in a minute.

\* In addition to language defined keywords & operators, we can define identifiers.

For example we can define a variable,  $x$ .

$x = 5.$

↑  
identifier

or a function

$f(x):$   
identifier return  $x+2$

\* Variables can hold different types of data :

- integers - floats

- boolean values (true/false)

- strings (characters)



## Basic keywords and operators

- cover basic i/o : input, print
- arithmetic : +, -, \*, /, %, \*\*
- comparison: ==, >, <, !=, <=, >=
- assignment: =, +=, -=
- logical : and, or, not
- data types: integer, bool, float, string
- comment character, semicolon ? backslash
- Indentation