# What is Programming?

Telling a story?

Instructions?

Recipe?

To whom? Audience

Computer

Instructor, Boss

Coworkers, classmates

Self

Why?

To get the story done

To impress the boss, coworkers, …

The best computer, language, style, …

Conflict:

religion, politics, sports

BUT NOT spoken language

Absolute freedom of choice – unless you want to be understood

Similar in computer language

# 1      Programming – A Task with Tools

Computer programming can be viewed as telling a story.  A programming language, like python, provides a number of tools/parts to aid us in telling the story.  While helpful, we need not use all these tools in the telling of every story.

In most of the tasks we do in life, we make use of tools to help us complete these tasks.   These tools have come into being over a long time.  Computer programming can be one of these tasks.  A computer programming language like python contains many tools.  A good thing to remember is these tools are present because somebody thought they were useful.  The more we recognize these tools are there because they have been helpful, the better we can cope with the parts of programming which may be new to us.

# **1.1**     **Rules – How to use the tools**

Life’s activities are full of rules.  Many are the result of our environment and not of our choosing.  Things in the air fall down.  Things on a table stay put – usually.  We have come to accept and even make use of these rules.  In computer programming we often create the rules we want.   In textual things like arithmetic, story writing, and computer programming these rules have been often labeled as syntax.

We’re not going to include any where near all the tools and rules – too long.  Just to say that is often useful to recognize they are there and it is often useful to know some of therm.  The following are some examples of tools and rules we use.

## 1.2     Building Blocks

I’ve always liked building blocks, especially those made of hard smooth wood.

**1.2.1**     **Story / Goal**

 Construction

**1.2.2**     **Tools:**

Blocks

·       Rectangular solid

·       Ramp

·       Arch

·       Small block

·       Double length block

·       Special e.g., base with map

**1.2.3**     **Rules**

Can place on top of each other

Can’t cut

No glue

1. **1.3**     **Arithmetic**

**1.3.1**     **Story / Goal**

calculations

**1.3.2**     **Tools**

·       Characters

o   Digits

o   Decimal point

o   Operators: +, -, \*,/

·       Numbers

o   From Digits, Decimal point

·       Expressions

o   From Numbers, Operators

o   E.g., 1 + 2

**1.3.3**     **Rules**

Expression à Number or Expression Operator Expression

1. **1.4**     **Algebra**

**1.4.1**     **Story / Goal**

Show relationships, General numeric “truths”

**1.4.2**     **Tools**

·       From Arithmetic

·       Letters: a-z

·       Variables: letters

·       Assignments

o   Using =

·       Comparisons

o   Using characters: ==, !=, >, <, <=

1. **1.5**     **Written Communication – Story writing / reading**

**1.5.1**     **Story / Goal**

 Providing / receiving a story

**1.5.2**     **Tools**

Parts to aid in providing / receiving the story

·       Characters

o   Letters

o   Digits

·       Words

o   From characters

·       Definitions – rules for what words mean and how they are constructed(spelled)

·       Sentence

o   From words

o   From syntax – sentence rules

·       From Arithmetic

·       From Algebra

**1.5.3**     **Rules / Syntax**

·       Word syntax

·       Sentence syntax

·       Paragraph syntax

1. **1.6**     **Vehicles**

Wheels

Carriage

Motors

1. **1.7**     **Transportation**

Roads

Rail

Air

1. **1.8**     **Buildings**

Floors

Doors

Windows

Heating

1. **1.9**     **Food Preparation**

Measurement

Containers

Heating / Cooking

Timing

Cutting

**Programming Language, e.g., python**

**1.10.1**   **Story/Goal**

Perform a process, Provide answer

**1.10.2**   **Tools**

·       From parts of Written Communication

·       From parts of Algebra

o   With longer names

o   Changes to account for character usage e.g., use “\*” for multiplication

·       Control flow – alternate paths for operation/computation

o   Branches

o   Loops (repeating)

o   Exception control (special form of branching) – program flow modified by exceptional circumstances

§  Python: **try, except**- capturing unexpected occurrences

o   Named collection of operation/computation - function

·       Definition of Process – **function -** name which does a possibly, complex set of operations

o   Predefined - Built-in – previously defined to do well know things

o   User defined – facilitating use collections of often done things

·       Data Definition

o   predefined

§  Character strings

§  List

§  Dictionary

o   User defined

·       Definition of Objects (combination of process and data)

o   Available in most new programming languages, including python

o   Often useful in   providing “lifelike” variables – those that behave like “real-world” objects