Section 1.1

Every programming language must provide following functions.

1. **Primitive expressions**
2. **Means of combinations**
3. **Means of abstractions**

A program consists of the data and procedures.
Data is the thing that we are manipulating and procedures are the instructions to manipulate them.

Expressions

As mentioned above, each programming language must provide primitive data types. They are called primitive expressions.
Expressions can be combined with other expressions to create another expression.
For numerical data, one of the procedures to combine expressions is '+'.
Expression '+' Expression '=' Compound expression
Ex. (+ 23 42)

Combination

Expressions formed by the list of expressions, delimited by parentheses, to denote procedure application, are called

Prefix Notation

(+ 32 45)
(+ 23 45 35 67) => apply + operation on a list of arguments.

Combinations can be nested

(* (+ 45 782) 78)

In Lisp, every expression has a value.

Section 1.2

In scheme dialect of Lisp, name of the variable is defined with `define`.
(define a 10)

Complex programs are built step by step making complex abstractions.

In order to use this abstraction of naming `objects`, the interpreter must maintain a memory of this mapping. This memory is called 'Environment'.