Procedure Application

The main form of a Scheme expression is the procedure application. (Terminology: in Scheme, the official name for what you would think of as a function is *procedure*.)

```
(procedure arg1 arg2 ... argn)
```

Evaluation

- Each argument is evaluated.
- The procedure is applied to the results.

Exception: **syntactic forms**.

Syntactic forms violate the rule—they are built in to the language to handle cases the rule above can't handle. Examples: `define, if, cond, lambda---more on this later.`

Examples

- `(- 1) => -1`
- `(* 5 7) => 35`
- `(+ 1 2 (* 2 3)) => 9`
- `(+ (- 6 3) (/ 10 2) 2 (* 2 3)) => 16`
- `(cos 0) => 1`

Exercice: run Scheme and try the arithmetic operators with 0, 1, 2 and 3 arguments, and figure out how the results make sense.