Functions

- We would like to execute the same code many times, each time with different values for some of the identifiers
  - To avoid repeating the same code, we can define a function
- Functions are shortcuts for program code to execute, just as variables are shortcuts for values
  - To be precise, functions are just another kind of value in memory, like numbers (as we will see later)
- Function Sqr returns the square of its input:

```
declare
fun {Sqr X} X*X end
```

- The **fun** keyword identifies the function. The identifier Sqr refers to a variable that is bound to the function.

Numbers

- There are two kinds of numbers in Oz
  - Exact numbers: integers
  - Approximate numbers: floating point
- Integers are exact (arbitrary precision)
- Floats are approximations of real numbers (up to 15 digits precision – 64-bit internally)
- There is never any automatic conversion from exact to approximate and vice versa
  - To convert, we use functions IntToFloat or FloatToInt
  - Design principle: don't mix incompatible concepts

Sum of digits function

- Function SumDigits calculates the sum of digits of a three-digit positive integer:

  ```
  declare
  fun {SumDigits N}
     (N mod 10) + ((N div 10) mod 10) +
     ((N div 100) mod 10)
  end
  ```

- **mod** and **div** are integer functions
- / (division) is a float function
- * (multiplication) is a function on both floats and integers