```
Examples of functions in Soy ...

--------------------------------------------------------------------------------

`*/================================================\
  | Any number divisible by 3? Fizz.                |
  | Any number divisible by 5? Buzz.                |
  | If a number is divisible by 3 and 5, FizzBuzz. |
  | Else? Return the number.                        |
  \================================================/*`
let FizzBuzz : `string <- (n : `int) =
    “Fizz” if n % 3 else “” =+
    “Buzz” if n % 5 else “”
        .. if not `empty else n;

--------------------------------------------------------------------------------

`` A function that finds the minimum value in a collection of numbers.
``
let Min : `number <- (c : `collection) =
{
    let val := c_0;
    for every e in c[1:] do
        val = e if < val;
    <<< val;
}

`` A function that finds the maximum value in a collection of numbers.
``
let Max : `number <- (c : `collection) =
{
    let val := c_0;
    for every e in c[1:] do
        val = e if > val;
    <<< val;
}

`#Min `` A function that finds the minimum value in a collection of numbers.
`#Max `` A function that finds the maximum value in a collection of numbers.
      ``
let Min & Max : `number <- (c : `collection) =
{
    let val := c_0;
    for every e in c[1:] do
        `#Min : val = e if < val;
        `#Max : val = e if > val;
    <<< val;
}

--------------------------------------------------------------------------------
```