

# Science fonctionnelle

Gianni Mocellin

<b>Introduction</b>	<b>3</b>
<b>Fonctions</b>	<b>3</b>
<b>Array</b>	<b>3</b>
Array[f,n]	3
Array[f,n,r]	3
<b>Inner</b>	<b>3</b>
Inner[f,list1,list2,g]	3
<b>List</b>	<b>3</b>
<b>Listable</b>	<b>3</b>
<b>Map</b>	<b>4</b>
Map[f,expr]	4
f/@expr	4
<b>Orderless</b>	<b>4</b>
<b>Outer</b>	<b>4</b>
Outer[f,list1,list2,...]	4
<b>SparseArray</b>	<b>5</b>
SparseArray[{pos1->val1,pos2->val2,...}]	5
SparseArray[{pos1,pos2,...}->{val1,val2,...}]	5
SparseArray[list]	5
SparseArray[data,{d1,d2,...}]	5
SparseArray[data,dims,val]	5
SparseArray[{{i_i}->1},{d,d}]	5

# Introduction

## Fonctions

### Array

#### Array[f,n]

Generates a list of length n, with elements f[i].

#### Array[f,n,r]

generates a list using the index origin r.

### Inner

#### Inner[f,list<sub>1</sub>,list<sub>2</sub>,g]

In Dot f plays the role of multiplication and g of addition.

### List

Lists are very general objects that represent collections of expressions.

Listable lists do arithmetic on the whole list at once.

### Listable

A listable function applies separately on each element in a list or to the corresponding elements of each list if there is more than one list.

Functions with attribute Listable are automatically "threaded" over lists, so that they act separately on each list element.

All list arguments must have the same length.

Arguments that are not lists are as many times as there are elements in the lists, replicated as needed.

## Map

**Map[f,expr]**

or

**f/@expr**

applies f to each element on the first level in expr.

applies f to parts of expr specified by levelspec.

The default value for levelspec in Map is {1}.

{n}: level n only

{n<sub>1</sub>,n<sub>2</sub>}: levels n<sub>1</sub> through n<sub>2</sub>

Infinity: levels 1 through Infinity

Map[f]

represents an operator form of Map that can be applied to an expression.

## Orderless

Is an attribute that can be assigned to a symbol f to indicate that the elements e<sub>i</sub> in expressions of the form f[e<sub>1</sub>,e<sub>2</sub>,...] should automatically be sorted into canonical order.

This property is accounted for in pattern matching.

## Outer

**Outer[f,list<sub>1</sub>,list<sub>2</sub>,...]**

Forms all possible combinations of the lowest-level elements in each of the lists, and feeds them as arguments to f.

## SparseArray

**SparseArray[{pos1->val1,pos2->val2,...}]**

Generates a sparse array in which values  $val_i$  appear at positions  $pos_i$ .

By default, SparseArray takes unspecified elements to be 0.

**SparseArray[{pos<sub>1</sub>,pos<sub>2</sub>,...}->{val<sub>1</sub>,val<sub>2</sub>,...}]**

Yields the same sparse array.

**SparseArray[list]**

Yields a sparse array version of list.

**SparseArray[data,{d<sub>1</sub>,d<sub>2</sub>,...}]**

yields a sparse array representing a  $d_1 \times d_2 \times \dots$  array.

**SparseArray[data,dims,val]**

yields a sparse array in which unspecified elements are taken to have value val.

**SparseArray[{{i<sub>1</sub>,i<sub>2</sub>}->1},{d,d}]**

Gives a  $d \times d$  identity matrix.