

Note on Mathematica Programming

Hu Xiping

July 14, 2019

Contents

I	Basics of Wolfram Language	5
1	Features of Mathematica	7
1.1	Evaluating Commands	7
1.2	Auto Complete	7
1.3	Studying Resources	7
1.4	Elementary Arithmetic	7
2	First glance at the Functions	9
2.1	Usage	9
2.1.1	Some basic functions	9
3	Introduction to Lists	11
3.1	List is a way to store numbers	11
3.2	Create a List	11
3.2.1	Range[] is a basic function to create lists	11
3.2.1.0.1	Note	11
3.2.2	IntegerDigits[] convert number to list	11
3.2.3	Use Table to create List iteratively	11
3.2.3.1	Repeated	11
3.2.3.2	Iterating	11
3.3	Visualizing Lists	12
3.3.1	ListPlot	12
3.3.2	ListLinePlot	12
3.3.3	BarChart	13
3.3.4	PieChart	13
3.3.5	NumberLinePlot	14
3.4	Advanced Operation on List	14
3.4.1	Operate with Basic arithmetic operators	14
3.4.1.1	Operate with numbers	14
3.4.1.2	Operate with other lists	14
3.4.2	Operate with Functions	14
3.4.2.1	Whole list operation	14
3.4.2.2	Elements operation	14

Part I

Basics of Wolfram Language

Chapter 1

Features of Mathematica

1.1 Evaluating Commands

On desktop and web, you may press **Shift+Enter**. On mobile, press the Wolfram icon vbutton

1.2 Auto Complete

Within the Mathematica notebook, you'll see a variety of aids to help you enter the Wolfram Language.

1.3 Studying Resources

You may RTFM(Read The F***ing Manual) or visit Wolfram website to equip essential skills on Wolfram Language. Or you can JFGI(Just F***ing Google It) if your problem can't be solved. Note that you need to use Google instead of Baidu due to study efficiency and you'd better use English to search for help.

1.4 Elementary Arithmetic

Command	Expression	Example
Add	+	2+2
Subtract	-	2-2
Multiply	*	2*2
fDivision	/	2/2
Power	^	2^2
Brackets	(and)	(2+3)/5

Chapter 2

First glance at the Functions

2.1 Usage

Functions names are all started with capital letters. To use a function, attach a "[]" behind the name of function and input parameters separated with "," into the brackets. Tip: insert a single space after the comma to make your code more visualized.

Example `Plus[3,4,5]`

Output: 12

You may use the output of function as a parameter of other functions.

Example `Times[2,Plus[2,3]]`

Output: 7

2.1.1 Some basic functions

Plus[2, 3]

Subtract[2, 3]

Times[2, 3]

Divide[2, 3]

Power[2, 3]

Max[2, 3]

Min[2, 3]

RandomInteger[100]

Chapter 3

Introduction to Lists

3.1 List is a way to store numbers

Example `{1,2,3,4,5}` is a list

3.2 Create a List

3.2.1 `Range[]` is a basic function to create lists

Example `Range[10]`

Output: `{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}`

3.2.1.0.1 Note `Range[m,n,p]` means a list start with m, end with n, in step of p

Example `Range[3,7]`

Output: `{3, 4, 5, 6, 7}`

Example `Range[2,10,3]`

Output: `{2, 5, 8}`

3.2.2 `IntegerDigits[]` convert number to list

Use `IntegerDigits` to create lists out of integer number

Example `IntegerDigits[1988]`

Output: `{1,9,8,8}`

3.2.3 Use `Table` to create List iteratively

3.2.3.1 Repeated

Usage: `Table[content,times]`

Example `Table[2,5]`

Output: `{2,2,2,2,2}`

Example `Table[x,4]`

Output: `{x,x,x,x}`

Example `Table[{1,2},3]`

Output: `{{1,2},{1,2},{1,2}}`

3.2.3.2 Iterating

Iterate from 1 to n

Usage: `Table[expression, {varibale, n}]`

Example `Table[x^2, {x, 4}]`

Output: `{1,4,9,16}`

Example `Table[Range[expt], {expt, 3}]`

Output: `{{1}, {1,2}, {1,2,3}}`

Iterate from m to n

Usage: `Table[expression, {variable, m, n}]`

Example `Table[f[n], {n, 4, 7}]`

Output: `{f [4], f [5], f [6], f [7]}`

Iterate from m to n in steps of p

Usage: **Table**[expression, {m,n,p}]

Example **Table**[g[a],{4,10,2}]

Output: {g[4], g[6], g[8], g[10]}

Use 2 variables to iterate

Example **Table**[i, {i, 4}, {j, 5}]

Output: {{1, 1, 1, 1, 1}, {2, 2, 2, 2, 2}, {3, 3, 3, 3, 3}, {4, 4, 4, 4, 4}}

3.3 Visualizing Lists

3.3.1 ListPlot

Example **ListPlot** [1,2,3,4,3,4]

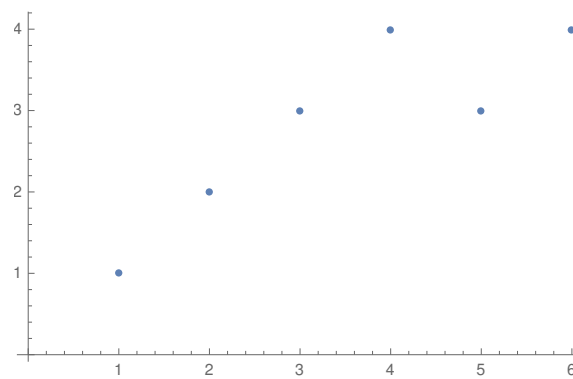


Figure 3.1: ListPlot

Example **ListPlot**[Join[Range[20]], Reverse[Range[20]], Range[30]]

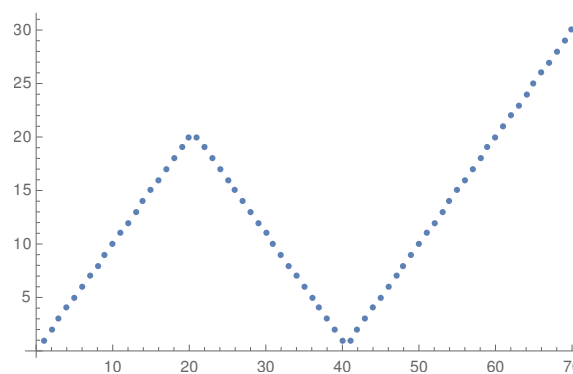


Figure 3.2: ListPlot

3.3.2 ListLinePlot

Example **ListLinePlot** [{1.5, 2, 4, 2.3, -9}]

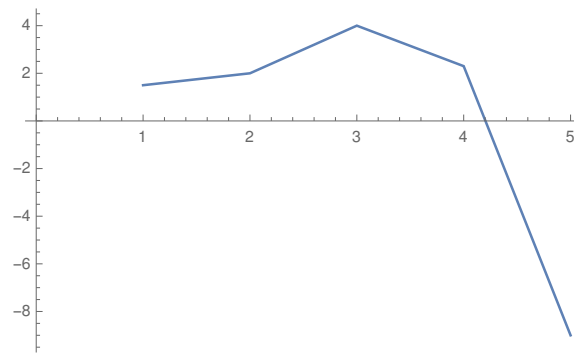


Figure 3.3: ListLinePlot

3.3.3 BarChart

Example `BarChart[{1.5, 2, 4, 2.3, -9}]`

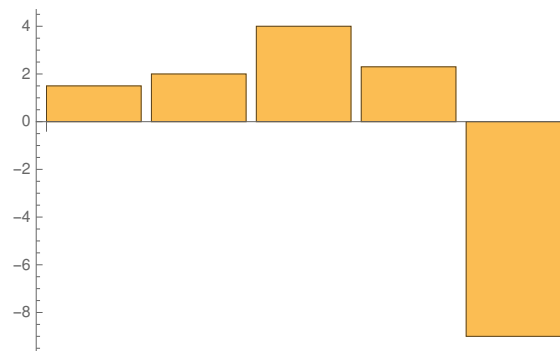


Figure 3.4: BarChart

3.3.4 PieChart

Example `BarChart[{1, 3, 5, 4}]`

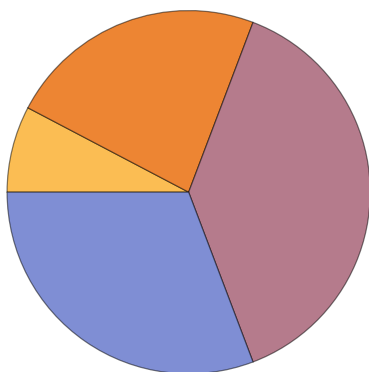


Figure 3.5: PieChart



Figure 3.6: Interact with PieChart segments

Note that You can also left click on the chart to interact with the segments

3.3.5 NumberLinePlot

Example `NumberLinePlot[{2, 3, 1, 5, -6.2}]`



Figure 3.7: NumberLinePlot

3.4 Advanced Operation on List

3.4.1 Operate with Basic arithmetic operators

3.4.1.1 Operate with numbers

Example `{1, 2, 3} + 10`

Output: `{11,12,13}`

Example `{1,2,3}-10`

Output: `{-9,-8,-7}`

Example `{1,2,3}*10`

Output: `{10,20,30}`

Example `{1,2,3}/10`

Output: `{1/10, 1/5, 3/10}`

3.4.1.2 Operate with other lists

Example `{1,2,3}+{1,2,3}`

Output: `{2,3,4}`

Example `{1,2,3}-{2,3,4}`

Output: `{-1,-1,-1}`

Example `{1,2,3}*{1,2,3}`

Output: `{1,4,9}`

Example `{1,2,3}/{2,3,4}`

Output: `{1/2, 2/3, 3/4}`

3.4.2 Operate with Functions

3.4.2.1 Whole list operation

Join lists together

Example `Join[Range[3],Range[5],Range[3]]`

Output: `{1, 2, 3, 1, 2, 3, 4, 5, 1, 2, 3}`

Reverse elements

Example `Reverse[{1,2,3}]`

Output: `{3, 2, 1}`

Length of the list

Example `Length[{5,3,4,2,3,4}]`

Output: `6`

Sum up

Example `Total[{1,2,3}]`

Output: `6`

Sort the elements

Example `Sort[{6,7,1}]`

Output: `{1,6,7}`

3.4.2.2 Elements operation

See how many times an element appears

Example `Count[{a,b,a,a,c,b,a},a]`

Output: `4`

Extract elements

Usage: `Part[list, position]`

Example `Part[{7,6,5},2]`

Output: `6`

Extract the first Element

Example `First[{7,6,5}]` (*The same as `Part[list,1]`*)

Output: `7`

Extract the last element

Example `Last[{7,6,5}]`

Output: 5

Extract the Max and the Min

Example `Min[{6,7,1}]`

Output: 1

Example `Max[{6,7,1}]`

Output: 7

Slice the list

Usage: `Take/Drop[list, position]`

Example `Take[{101,203,401,602,332,412},3]`

Output: {101,203,401}

Example `Drop[{102,203,401,602,332,412},3]`

Output: {602,332,412}