

# Code sample

## 1. Numerical Calculation

### ➤ Null values judgment

	A	
1	=null	
2	=0	
3	=if(A1==null,"null","not null")	null
4	=if(A2!=null,"not null","null")	not null
5	=[,1,2,3].ifn()	1

### ➤ Random values

	A	
1	=rand()	Get a random value not less than <b>0</b> and less than <b>1</b>
2	=int(rand()*1000)	Get random integer values between <b>0</b> and <b>999</b>

### ➤ Sign of values

	A	
1	=sign(45)	Return 1 for positive values
2	=sign(-100.34)	Return -1 for negative values
3	=sign(0)	Return 0 for zero
4	=abs(-4.6)	Return 4.6 for absolute value

### ➤ Involution and evolution

	A	
1	=power(2,3)	Cube
2	=power(-2,3)	Cube
3	=power(4,0.5)	Square root
4	=power(27,1/3)	Cube root

### ➤ Decimal truncation and rounding



	A	
1	=round(3451251.274,1)	Round down to 1 decimal place
2	=round(3451251.274,2)	Round down to 2 decimal places
3	=ceil(3450001.003,-2)	Complete carry and accurate to 100
4	=ceil(3450001.003,2)	Complete carry and reserve 2 decimal
5	=floor(3451291.234,-2)	Round all and accurate to 100
6	=floor(3451281.238,2)	Round all and reserve 2 decimal places

➤ **Pi**

	A	
1	=pi()	pi
2	=pi(4)	4*pi

➤ **Trigonometric function**

	A	
1	=sin(pi(30/180))	Sine
2	=cos(pi()/2)	Cosine
3	=tan(pi()/4)	Tangent
4	=asin(0.5)	Arc sine
5	=acos(-0.5)	Arc cosine
6	=atan(1)	Arc tangent

➤ **Logarithm function**

	A	
1	=lg(10000)	Logarithm with 10 as its base
2	=ln(1000)	Natural logarithm
3	=exp(A2)	Powers of e

➤ **Compute different expressions on conditions**

	A	
1	3000	
2	=if(A1>10000, A1*0.45+450, A1>5000, A1*0.15+150, A1*0.05)	150.0
3	manager	
4	=case(A3,"president":500,"manager":300,"employee":150)	300

➤ **Use temp variables in the expressions**

	A	
1	=(a=1,b=a*3,b+4)	7
2	=a	1
3	=b	3

## 2. Strings

➤ **Generate strings of fixed length**

	A	
1	=fill(" ",10)	" "
2	=len(A1)	10
3	=fill("ab",10)	ababababababababab

➤ **Search and replace substrings**

	A	
1	=pos("abcdef","def")	Determine the position of "def" in "abcdef"
2	=pos("abcdefdef","def",5)	Search from the fifth character
3	=replace("abca","a","ABC")	Replace "a" with "ABC" in "abca"
4	=replace("abc'abc","a","ABC")	Substrings in the quotation marks will also be replaced
5	=replace@q("abc'abc","a","ABC")	Substrings in the quotation marks will not be replaced

➤ **Acquire part of a string**

	A	
1	=mid("abcde",2,1)	Acquire the second character
2	=mid("abcde",3,2)	Acquire 2 characters from the third position
3	=mid("abcde",2)	Acquire characters from the second position to the end
4	=left("abcdefg",3)	The left three characters
5	=right("abcde",2)	The right two characters

➤ **Joining of the strings**

	A	
1	="ab"+"cd"	
2	="3"+2	5, the sting is taken as number when computed with numbers.
3	="ab"+1	1, the character strings will not be converted into numbers and will be taken as 0.

➤ **Case identification and conversion**

A		
1	=upper("abcdef")	"ABCDEF"
2	=upper("ABCdef")	"ABCDEF"
3	=lower("abcDEF")	"abcdef"
4	=isupper("ABC")	true
5	=islower("ABC")	false
6	=islower("aBc")	false
7	=isupper("Bc")	false

➤ **Remove the blank spaces on both sides of a string**

A		
1	=trim(" abc ")	Remove the spaces on both sides
2	=trim@l(" abc ")	Remove the spaces on the left
3	=trim@r(" abc ")	Remove the spaces on the right

➤ **Match the pattern strings**

A		
1	=like("abc123", "abc1?3")	"?" is used to match a single character
2	=like("abcefg", "abc*")	"*" is used to match 0 or more characters

➤ **Match the pattern strings with '\\*'**

A		
1	=like("ab*123", "ab\*1?3")	The escape character "\" can be used to match its tailing character like "*" in this case.
2	=like("a*bcefg", "a\*bc*")	true

➤ **Acquire character codes and return characters by encoding**

A		
1	=asc("a")	Encode the character "a"
2	=char(68)	Get the character of code "68"
3	=asc("USA")	Get the code of character "U"



➤ **Split a string into sequences**

	A	
1	= <code>"12345678"</code>	
2	= <code>len(A1)</code>	
3	= <code>A2.(mid(A1,#,1))</code>	Split it into character sequences
4	= <code>"a,[b,c],d"</code>	
5	= <code>A4.array ()</code>	Return a sequence which consists of <b>a</b> , <b>[b,c]</b> , <b>d</b> three members where the member <b>[b,c]</b> is a sequence
6	= <code>A4.array@s()</code>	Return a sequence which consists of <b>a</b> , <b>[b,c]</b> , <b>d</b> three members where the member <b>[b,c]</b> is a string instead of a sequence
7	= <code>A4.array@b()</code>	Return a sequence which consists of <b>a</b> , <b>[b,c]</b> , <b>d</b> four members. The quotation marks and brackets will not be matched during processing.
8	= <code>"a;[b;c];d".array (";")</code>	Use ";" as the delimiter instead of the default","

➤ **Join the sequences into a string**

	A	
1	= <code>[1,"abc,def",[2,4],"{7,8}"]</code>	
2	= <code>A1.string()</code>	Use "," as the delimiter to join and quote the strings in the quotation marks
3	= <code>A1.string@d()</code>	String members don't need to use quotation marks when joining a string.
4	= <code>A1.string("&amp;")</code>	Use "&" as the delimiter
5	= <code>[1,"abc,def","123"].conj@s()</code>	Join strings directly without delimiter

➤ **Remove the not necessary characters from a string**

	A	
1	<code>abcda123efag</code>	
2	= <code>replace(A1,"123","")</code>	Remove "123"
3	= <code>replace(A1,"a","")</code>	Remove "a"
4	= <code>len(A1).(mid(A1,#,1)).select(pos("ace",~)==0).conj@s()</code>	Remove all of the "a,c,e"

➤ **Take out the letter or numerical part from a string**



	A	
1	2345\$#dfAgsdf23*	
2	=len(A1)	16
3	=A2.(mid(A1,#,1))	Split it into character sequences
4	=A3.select(isdigit(~)).conj@s()	Numerical part
5	=A3.select(isalpha(~)).conj@s()	Letter part

➤ **Check if a string is consisted of letters or numbers**

	A	
1	2345\$#dfAgsdf23*	
2	=len(A1)	16
3	=A2.(mid(A1,#,1))	Split it into character sequences
4	=A3.count(!isdigit(~) && !isalpha(~))==0	Check if it is a alphanumeric string

➤ **Compute the expressions stored in a string**

	A	
1	=eval("1+5")	Compute 1 + 5
2	=eval(\${A1+2})	Compute A1 + 2

➤ **Use arguments when computing the expressions in a string**

	A	
1	=eval("?+5",3)	Equivalent to "3 + 5"
2	=eval("(?1+1)/?2",3,4)	Equivalent to "(3 + 1)/4"

➤ **Expression strings can change with the edition process**

	A	
1	=\$[B1+4]	The strings will change with the edition process in case \$[] method is used in coding

### 3. Datetime

➤ **Acquire the current date and time**

	A
1	=now()



➤ **Split every part of the date time**

	A
1	=now()
2	=year(A1)
3	=month(A1)
4	=day(A1)
5	=time(A1)
6	=hour(A1)
7	=minute(A1)
8	=second(A1)

➤ **Join every part to make up the date time**

	A	B	C	D	E	F
1	1989	02	01	02	34	55
2	=string(A1)+"-"+string(B1)+"-"+string(C1)+"					
3	=datetime(A2)					
4	=datetime(A2,"yyyy-MM-dd HH:mm:ss")					
5	=date(A1,B1,C1)					
6	=time(D1,E1,F1)					
7	=datetime(A1,B1,C1,D1,E1,F1)					
8	=datetime(A5,A6)					

➤ **Datetime away from a point**

	A	
1	2006-07-05	
2	=after(A1,5)	5 days later
3	=after("1972-11-08 10:20:30",-10)	10 days before
4	=after@s(A1,5)	5 seconds later
5	=after@s("1972-11-08 10:20:30",-10)	10 seconds before
6	=after@m(A1,-1)	1 month before
7	=after@y(A1,-1)	1 year before

➤ **Interval between two datetime**





	A	B	
1	2010-5-01 23:20:15	2010-05-03 01:01:01	
2	=interval(A1,B1)		The number of days difference between two datetimes
3	=interval@s(A1,B1)		The number of seconds difference between two datetimes
4	=interval@y(A1,"2001-01-01")		The number of years difference between two dates
5	=interval@m(A1,"2001-01-01")		The number of months difference between two dates
6	=interval@ms(A1,now())		The number of milliseconds difference between two datetimes

➤ **Day of a week**

	A	
1	2005-01-08	
2	=day@w(A1)	The result is the day of the week and "7" stands for "Saturday"

➤ **The first and the last days of a week, a month and a quarter**

	A	
1	2006-03-06	
2	=pdate@w(A1)	The first day of the week
3	=pdate@we(A1)	The last day of the week
4	=pdate@q(A1)	The first day of the quarter
5	=pdate@qe(A1)	The last day of the quarter
6	=pdate@m(A1)	The first day of the month
7	=pdate@me(A1)	The last day of the month

➤ **The number of days of a month, a quarter and a year**

	A	
1	2007-08-08	
2	=days(A1)	The days of the month in A1
3	=days@y(2006)	The days of the year 2006
4	=days@y(A1)	The days of the current year
5	=days@q(A1)	The days of the quarter in A1

➤ **Set periodic interval to generate datetime sequences**

	A	
1	2000-08-10 12:00:00	
2	=periods@y(A1,now(),1)	Set 1 year as the interval unit
3	=periods@q(A1,now(),1)	Set 1 quarter as the interval unit
4	=periods@m(A1,now(),2)	Set 2 months as the interval unit
5	=periods@d(A1,now(),7)	Set 7 days as the interval unit

➤ **Get the second and last Fridays in a month/quarter/year and the total number of Fridays in this period**

	A	
1	=now()	
2	=pdate@m(A1)	The begin date of the month in A1
3	=pdate@me(A1)	The end date of the month in A1
4	=periods@d(A2, A3,1)	The day sequence of the month
5	=A4.select(day@w(~)==6)	The Friday sequence of A4
6		Another alternative
7	=after(A2,6-day@w(A2))	Get the first Friday
8	=periods@dx(A7,A3,7)	Get the Friday sequences
9	=A8(2)	Get the second Friday
10	=A8.m(-1)	Get the last Friday
11	=A8.len()	Get the number of Fridays

## 4. Sequence

➤ **Put the data in the cellset together to form a sequence**

	A	B	C	D
1	1	3	6	4
2	=[ A1:D1]			

[1,3,6,4]

➤ **Check if it is an sequence**

	A	
1	=ifa([1,2,3])	true
2	=ifa(123)	false



➤ **Get the sequence member and sub-sequence in reversal**

**direction**

	A	
1	= <code>[1,2,3,4,5,6].m(-3)</code>	4
2	= <code>[1,2,3,4,5,6].m([-2,-3])</code>	[5,4]

➤ **Get the sequence member and sub-sequence in cycles**

	A	
1	<code>[1,2,3,4,5,6]</code>	
2	= <code>A1.m@r(10)</code>	4
3	= <code>A1.m@r([1,5,10])</code>	[1,5,4]

➤ **Get the sub-sequence and raise no errors even if they are out of**

**range**

	A	
1	= <code>[1,2,3,4,5,6].m@0([10,1,4,5])</code>	The result is [1,4,5] and the members out of range do not appear

➤ **Get the first non-null value**

	A	
1	= <code>[null,1,5,7].ifn()</code>	1

➤ **Generate a fixed length sequence consisted of the same**

**members**

	A	
1	= <code>5*[1]</code>	[1,1,1,1,1]
2	= <code>3.("a")</code>	[a,a,a]

➤ **Duplicate an sequence (for many times) to generate a new**

**sequence**



	A	
1	[1,2,3]	
2	=A1.dup()	[1,2,3]
3	=3*A1	[1,2,3,1,2,3,1,2,3]

➤ **Generate continuous integer sequence intervals**

	A	
1	=to(8)	[1,2,3,4,5,6,7,8]
2	=to(3,5)	The result is [3,4,5] for counting from 3 to 5
3	=to@s(3,5)	The result is [3,4,5,6,7] for counting 5 numbers from 3
4	=to@s(7,-3)	The result is [7,6,5]

➤ **Exchange member groups of an sequence**

	A	
1	= [1,2,3,4,5,6,7,8].swap([2,3,4],[6,7])	[1,6,7,5,2,3,4,8]

➤ **Insert one or multiple members in to an sequence**

	A	
1	= [1,2,3,4].insert(0,5)	[1,2,3,4,5]. Inserted into the back
2	= [1,2,3,4].insert(1,5)	[5,1,2,3,4]. Inserted into the front
3	= [1,2,3,4].insert(3,[5,6])	[1,2,5,6,3,4]. Insert multiple members

➤ **Delete one or multiple members from an sequence**

	A	
1	= [11,12,13,14].delete(2)	The result is [11,13,14] since delete a member
2	= [11,12,13,14].delete([2,4])	The result is [11,13] since delete multiple members

➤ **Modify one or multiple members of an sequence**

	A	
1	[11,12,13,14]	
2	=A1(2)=6	The value in A1 is [11,6,13,14]
3	=A1([3,4])=[7,8]	The value in A1 is [11,6,7,8]



- **Modify the sequence member at the specified position and fill up the position in case out of range**

	A	
1	= <code>[11,12,13,14,15].modify(2,6)</code>	[11,6,13,14,15]
2	= <code>[11,12,13,14,15].modify(10,10)</code>	[11,12,13,14,15,null,null,null,null,10]
3	= <code>[11,12,13,14,15].modify(2,[7,8,9])</code>	[11,7,8,9,15]

- **Insert the whole sequence into another sequence as one of its members**

	A	
1	[1,2,3,4]	
2	[5,6,7,8]	
3	= <code>A1.insert(3,[A2])</code>	[1,2,[5,6,7,8],3,4]

- **Compare in dictionary mode**

	A	
1	= <code>cmp(["a","b","c"],["d","e","f"])</code>	-1
2	= <code>cmp(["d","b","c"],["a","e","f"])</code>	1

## 5. Aggregation

- **Count and Length**

	A	
1	[1,2,null,6,7,8]	
2	= <code>A1.count()</code>	5, calculate number of non-null members when counting
3	= <code>A1.len()</code>	6, calculate number of all sequence members

- **Sum and Average**

	A	
1	[1,2,null,6,7,8]	
2	= <code>A1.sum()</code>	24, sum
3	= <code>A1.avg()</code>	4.8, ignore null in calculating average



➤ **Maximum and Minimum**

	A	
1	[1,2,null,6,7,8]	
2	=A1.max()	8, maximum value
3	=A1.min()	1, minimum value

➤ **Calculate Quadratic Sum and Variance**

	A	
1	[1,2,3,4,5,6,7,8]	
2	=A1.sum2()	Quadratic sum
3	=A1.variance()	Variance

➤ **Calculate Proportion and Cumulation**

	A	
1	[1,2,3,4,5,6,7]	
2	=A1.proportion()	Proportion
3	=A1.cumulate()	Cumulation

➤ **Calculate Rankings**

	A	
1	[1,-2,null,6,7,8]	
2	=A1.rank()	Sequence of descending rankings of each member
3	=A1.rank@z()	Sequence of ascending rankings of each member
4	=A1.ranki(0)	The ranking of 0 in the sequence

## 6. Calculate Sequence Number and Quantity in Calculation

### Cellset

➤ **Row number**



0	1	2		A	B	C	
1-		1	Quarter	Month	=row()		1
	1-	1	2	1		=row()	2
		1	3		Jan	=row()	3
		1	4		Feb		4, Auto-calculate using the expression in homocell C3
		1	5		Mar		5, Auto-calculate using the expression in homocell C3
	1-	1	6	2			6, Auto-calculate using the expression in homocell C2
		1	7		Apr		7, Auto-calculate using the expression in homocell C3
		1	8		May		8, Auto-calculate using the expression in homocell C3

➤ **Sequence numbers of homocells of current cell**

0	1	2		A	B	C	
1-		1	Quarter	Month			
	1-	1	2	1		=#	1, 1 <sup>st</sup> quarter
		1	3		Jan	=#	1, 1 <sup>st</sup> month of this quarter
		1	4		Feb		2, 2 <sup>nd</sup> month of this quarter
		1	5		Mar		3, 3 <sup>rd</sup> month of this quarter
	1-	1	6	2			2, 2 <sup>nd</sup> quarter
		1	7		Apr		1, 1 <sup>st</sup> month of this quarter
		1	8		May		2, 2 <sup>nd</sup> month of this quarter

➤ **Number of homobands of current cell**

0	1	2		A	B	C	
1-		1	Quarter	Month			
	1-	1	2	1		=##	2, 2 quarter-records in total
		1	3		Jan	=##	3, total 3 month-records in this quarter
		1	4		Feb		3, total 3 month-records in this quarter
		1	5		Mar		3, total 3 month-records in this quarter
	1-	1	6	2			2, 2 quarter-records in total
		1	7		Apr		2, total 2 month-records in this quarter
		1	8		May		2, total 2 month-records in this quarter

➤ **Sequence number of homocells of current cell at specified level**



0	1	2		A	B	C	
1-		1	Quarter	Month			
	1-	2	1				
		1	3		Jan	=A1#	1, 1 <sup>st</sup> month at the level where A1 is located
		1	4		Feb		2, 2 <sup>nd</sup> month at the level where A1 is located
	1	5		Mar		3, 3 <sup>rd</sup> month at the level where A1 is located	
	1-	6	2				
		1	7		Apr		4, 4 <sup>th</sup> month at the level where A1 is located
		1	8		May		5, 5 <sup>th</sup> month at the level where A1 is located

➤ **Number of homobands of current cell at specified level**

0	1	2		A	B	C	
1-		1	Quarter	Month			
	1-	2	1				
		1	3		Jan	=A1##	5, total 5 month-records at the level where A1 is located
		1	4		Feb		5, total 5 month-records at the level where A1 is located
	1	5		Mar		5, total 5 month-records at the level where A1 is located	
	1-	6	2				
		1	7		Apr		5, total 5 month-records at the level where A1 is located
		1	8		May		5, total 5 month-records at the level where A1 is located

➤ **Sequence number of homocells of specified cell at specified level**

0	1	2		A	B	C	
1-		1	Quarter	Month			
	1-	2	1			=ord(C3,A1)	1, C3 is located in the 1 <sup>st</sup> month at the level where A1 is located
		1	3		Jan		
		1	4		Feb		
	1	5		Mar			
	1-	6	2				4, auto-migrate and calculate ord(C7,A1)
		1	7		Apr		
		1	8		May		

➤ **Number of homobands of specified cell at specified level**





0	1	2		A	B	C
1-		1	Quarter	Month		
	1-	2	1			=num(C3,A1)
		1	3		Jan	
		1	4		Feb	
		1	5		Mar	
	1-	6	2			
		1	7		Apr	
		1	8		May	

5, total 5 Homocells of C3 at the level where A1 is located

5, Auto-migrate and calculate num(C7,A1)

➤ **Page Number and Number of Pages**

0	1	2		A	B	C
1-		1	Quarter	Month		=pgall()
	1-	2	1			=pgno()
		1	3		Jan	
		1	4		Feb	
		1	5		Mar	
	1-	6	2			
		1	7		Apr	
		1	8		May	

Number of Pages

Page number, Use related expression, and calculate when paging

Homocell of C2, Calculate page number when paging

**7. Cells and sets in the calculation cellset**

➤ **Cell sequence**

0	1	2		A	B	C
1-		1	Quarter	Month		=[A2:B4]
	1-	2	1			
		1	3		Jan	
		1	4		Feb	
		1	5		Mar	
	1-	6	2			
		1	7		Apr	
		1	8		May	

[1,null,null,Jan,null,Feb], not related to level

➤ **Homocell sequence**



0	1	2		A	B	C	
1-		1	Quarter	Month	={B5:C7}		[Mar,Apr], Homocells of B5 in [B5:C7]
	1-	2	1				
		1	3		Jan		
		1	4		Feb		
		1	5		Mar		
	1-	6	2				
		1	7		Apr		
		1	8		May		

➤ **Cell restricted by level**

0	1	2		A	B	C	
1-		1	Quarter	Month			
	1-	2	1				
		1	3		Jan		null, No record of previous month
		1	4		Feb		Jan, Auto-migrate and calculate previous month of this quarter
		1	5		Mar		Feb, Auto-migrate and calculate previous month of this quarter
	1-	6	2				
		1	7		Apr		null, No record of previous month of this quarter
		1	8		May	=B7[A6]	Apr, Previous month of this quarter

➤ **Set of homocells**

0	1	2		A	B	C	
1-		1	Quarter	Month	=A2{B3}		[Jan,Feb,Mar]
	1-	2	1			=A2{B3}	[Jan,Feb,Mar]
		1	3		Jan	=A2{B3}	[Jan,Feb,Mar]
		1	4		Feb		[Jan,Feb,Mar], Auto-migrate and calculate A2{B4}
		1	5		Mar		[Jan,Feb,Mar], Auto-migrate and calculate A2{B5}
	1-	6	2				[Apr,May], Auto-migrate and calculate A6{B7}
		1	7		Apr		[Apr,May], Auto-migrate and calculate A6{B7}
		1	8		May		[Apr,May], Auto-migrate and calculate A6{B8}

➤ **Set of homocells with no level specified**



0	1	2		A	B	C	
1-		1	Quarter	Month	=B3		[Jan, Feb, Mar, Apr, May], Get the current cell for the level
	1-	2	1			=B3	[Jan, Feb, Mar], Get the current cell for the level
		1	3		Jan	=B3	[Jan, Feb, Mar], Get the parent row of B3 for the level
		1	4		Feb		[Jan, Feb, Mar]
		1	5		Mar		[Jan, Feb, Mar]
	1-	6	2				[Apr, May], Auto-migrate and calculate (B7)
		1	7		Apr		[Apr, May], Auto-migrate and calculate (B7)
		1	8		May		[Apr, May]

➤ **Get homocell according to offset**

0	1	2		A	B	C	
1-		1	Quarter	Month			
	1-	2	1				
		1	3		Jan	=A2[B3:-1]	null, No records of previous month in this quarter
		1	4		Feb		Jan, Previous month of this quarter
		1	5		Mar		Feb, Previous month of this quarter
	1-	6	2				
		1	7		Apr		null, No records of previous month in this quarter
		1	8		May		Apr, Previous month of this quarter

➤ **Get homocell by specifying multi-level offset**

0	1	2		A	B	C	
1-		1	Quarter	Month			
	1-	2	1				
		1	3		Jan		null, No records of previous quarter
		1	4		Feb		null, No records of previous quarter
		1	5		Mar		null, No records of previous quarter
	1-	6	2				
		1	7		Apr	=A1[A6:-1][B7:1]	Feb, What is the month at the next position of previous quarter?
		1	8		May		Mar, What is the month at the next position of previous quarter?

➤ **Get set of homocells according to the starting and ending offsets**



0	1	2		A	B	C	
1-		1	Quarter	Month			
	1-		2	1			
		1	3		Jan	=A2{B3;1:2}	[Feb,Mar], Last 2 months in this quarter
		1	4		Feb		[Mar], Last 2 months in this quarter
	1	5		Mar		[], Last 2 months in this quarter	
	1-		6	2			
		1	7		Apr		[May], Last 2 months in this quarter
		1	8		May		[], Last 2 months in this quarter

➤ **Set of homocells within the page**

0	1	2		A	B	C	
1-		1	Quarter	Month			
	1-		2	1		==pgcell(B3)	All months in this page with related expression
		1	3		Jan		
		1	4		Feb		
	1	5		Mar			
	1-		6	2			Homocells of C2. Calculate all months in this page
		1	7		Apr		
		1	8		May		

## 8. Compare with Excel

➤ **String function**

**1) String concatenation**

In Excel, you can use the CONCATENATE function to concatenate several text character strings into one string. In esCalc, you can simply use “+” to concatenate several character strings. One thing to note is that the componential parts to be concatenated must be character string.

	A	
1	= "ab" + "cd"	
2	= "3" + 2	5, The sting is taken as number when computed with numbers.
3	= "ab" + 1	1, The character strings will not be converted into numbers and will be taken as 0.
4	= "ab" + string(1)	ab1

**2) DOLLAR function**

In Excel, use the DOLLAR function to round the decimal up or down to specified digits in the format of currency, and convert it to text. In esCalc, you can use string() function to handle it.

0	1	2	A	B	C	D
1-		1	Department	Name	Salary	
	1-	2	R&D			
		1		Alexis	5000	=string(C3,"\$0.00")
		1		Megan	10000	\$5000.00
		1		Victoria	7000	\$10000.00
		1				\$7000.00
	1-	6	Sale			
		1		Samantha	6500	\$6500.00
		1		Jonathan	8000	\$8000.00

The string() function has a similar functionality to that of TEXT() function in Excel.

For more string function usages in esCalc, please refer to the above mentioned **2.Strings**.

### ➤ Logic function

In Excel, for the AND, OR, and NOT logic operators, use AND(), OR(), and NOT() functions respectively. In esCalc, the syntax rule of common senior programming languages is adopted to use the "&&", "||", and "!" to represent. In Excel, value1=value2 is used to represent the corresponding judgment, while it is value1==value2 in esCalc.

0	1		A	B	C	D	E
1-		1	Food	Type	Origin	Price	
	1	2	Apple	F	MI	1.68	=if(B2=="V"    C2=="FL",D2*0.5,D2)
	1	3	Banana	F	FL	0.68	1.68
	1	4	Cucumber	V	CA	0.78	0.34
	1	5	Onion	V	FL	0.94	0.39
	1	6	Orange	F	FL	4.98	0.47
	1	7	Red Grape	F	OR	0.96	2.49
	1	8	Peach	F	CA	0.88	0.96
							0.88

In the column E, calculate the price of vegetables and all products of Florida after promoting them at half price.

0	1		A	B	C	D	E
1-		1	Food	Type	Origin	Price	
	1	2	Apple	F	MI	1.68	=B2!="V"&&C2!="FL"
	1	3	Banana	F	FL	0.68	true
	1	4	Cucumber	V	CA	0.78	false
	1	5	Onion	V	FL	0.94	false
	1	6	Orange	F	FL	4.98	false
	1	7	Red Grape	F	OR	0.96	true
	1	8	Peach	F	CA	0.88	true

In the column E, calculate if the price will keep intact after the sales promotion of vegetables

and all products in Florida at half price.

## ➤ Statistical functions

### 1) Sum

In Excel, the common sum functions are SUM() and SUMIF(). In esCalc, you can use  $A.sum()$  to calculate the sum of every member in the set  $A$ , or use  $A.sumif(...)$  to sum up the members meeting the “...” conditions. The  $A$  could be the integer sequence or set of cells.

0	1	2	A	B	C	D
1-		1	Department	Name	Salary	$=\{C3\}.sumif(;\sim>7000)$ 18000, Specify conditions for sum up
	1-	2	R&D			$=\{C3\}.sum()$ 22000, Total departmental salaries
		1		Alexis	5000	
		1		Megan	10000	
		1		Victoria	7000	
	1-	6	Sale			14500, homocell of D2
		1		Samantha	6500	
		1		Jonathan	8000	

### 2) Count

In Excel, the most common count function is COUNT(). In esCalc, you can use  $A.count()$  to calculate the number of all members in the set  $A$ , or use  $A.countif(...)$  to calculate the number of employees in set  $A$  meeting the conditions “...”. The  $A$  could be integer sequence or set of cells.

0	1	2	A	B	C	D
1-		1	Department	Name	Salary	$=\{C3\}.countif(;\sim>7000)$ 2, specify conditions for counting
	1-	2	R&D			$=\{B3\}.count()$ 3, number of departmental employee
		1		Alexis	5000	
		1		Megan	10000	
		1		Victoria	7000	
	1-	6	Sale			2, homocell of D2
		1		Samantha	6500	
		1		Jonathan	8000	

### 3) Calculate average

In Excel, the most common function for calculating the average is the AVERAGE(). In esCalc, you can use  $A.avg()$  to calculate the average of all members in the set  $A$  or use  $A.avgif(...)$  to calculate the average on members meeting the conditions “...”. The  $A$  could be integer sequence or set of cells.



0	1	2	A	B	C	D	
1-		1	Department	Name	Salary	= <b>{C3}.avgif(;&gt;7000)</b>	9000, calculate average on conditions
	1-	2	R&D			= <b>round({C3}.avg(),0)</b>	7333, average of departmental salary
		1		Alexis	5000		
		1		Megan	10000		
		1		Victoria	7000		
	1-	6	Sale				7250, homocells of D2
		1		Samantha	6500		
		1		Jonathan	8000		

For more statistics function usages in esCalc, please refer to the above mentioned

### 5. Aggregation.

#### ➤ Date Time Function

	A	
1	= <b>now()</b>	Get the current date and time
2	= <b>date(2012,2,26)</b>	Make up the date time
3	= <b>day(A1)</b>	Get the day of a date

For more date time function usages in esCalc, please refer to the above mentioned

### 3. Datetime.

#### ➤ Mathematic function

	A	
1	= <b>abs(-4.6)</b>	Return 4.6 for absolute value
2	= <b>cos(pi()/2)</b>	Cosine
3	= <b>exp(A2)</b>	Powers of e

For more mathematic function usages in esCalc, please refer to the above-mentioned

### 1. Numerical Calculation.