

数尊信息科技（上海）有限公司

函数说明文档

数尊塔系统支持的函数使用详解

2018/6/11

本文档只针对数尊塔 2.5 版本支持的函数进行详解，其他版本需要找对应的函数说明文档参考，暂不适用本文档。

目录

1.目的	3
2.注意事项.....	3
3.函数的适用范围.....	3
4.字符型函数.....	4
4.1 charAt(source,index)	4
4.2 compareTo(arg1,arg2)	4
4.3 compareToIgnoreCase(arg1,arg2)	4
4.4 concat(arg1,arg2)	5
4.5 contains(arg1,arg2)	5
4.6 notContains(arg1,arg2)	5
4.7 endsWith(arg1,arg2)	6
4.8 startsWith(arg1,arg2,arg3)	6
4.9 equals(arg1,arg2)	6
4.10 notEquals(arg1,arg2)	7
4.11 equalsIgnoreCase(arg1,arg2)	7
4.12 age(arg1)	7
4.13 left(arg1,arg2)	7
4.14 mid(arg1,arg2,arg3).....	8
4.15 right(arg1,arg2)	8
4.16 toNumber(arg1)	8
4.17 trim(arg1)	9
4.18 toUpperCase(arg1)	9
4.19 toLowerCase(arg1)	9
4.20 substring(arg1,arg2,arg3)	9
4.21 replace(arg1,arg2,arg3)	10
4.22 length(arg1)	10
4.23 lastIndexOf(arg1,arg2).....	10
4.24 indexOf(arg1,arg2,arg3)	11
5.数值型函数.....	12
5.1 abs(arg1)	12
5.2 acos(arg1).....	12
5.3 cos(arg1).....	12
5.4 asin(arg1).....	12
5.5 sin(arg1)	13
5.6 atan(arg1).....	13
5.7 tan(arg1).....	13
5.8 exp(arg1)	14
5.9 avg(arg1,arg2,.....)	14
5.10 ceil(arg1).....	14
5.11 floor(arg1)	14
5.12 round(arg1)	15
5.13 ln(arg1)	15

5.14 log(arg1)	15
5.15 max(arg1,arg2,.....)	15
5.16 min(arg1,arg2,.....)	16
5.17 pow(arg1,arg2)	16
5.18 random()	16
5.19 rint(arg1)	16
5.20 sqrt(arg1)	17
5.21 sum(arg1,arg2,.....)	17
5.22 toDegrees(arg1)	17
5.23 toRadians(arg1)	18
6. 日期函数	19
6.1 dateif(arg1,arg2,'Y')	19
6.2 dateif(arg1,arg2,'M')	19
6.3 dateif(arg1,arg2,'D')	19
6.4 dateif(arg1,arg2,'YM')	20
6.5 dateif(arg1,arg2,'YD')	20
6.6 dateif(arg1,arg2,'MD')	20
6.7 dateif(arg1,arg2,'DD')	20
6.7 day(arg1)	21
6.8 month(arg1)	21
6.9 now()	21
6.10 today()	21
6.11 year(arg1)	22
7. 运算符	23
7.1 数学运算符	23
7.2 关系运算符	23
7.3 逻辑运算符	23
8. 数组函数	24
8.1 gavg(arg1)	24
8.2 gmax(arg1)	24
8.3 gmin(arg1)	24
8.4 gsum(arg1)	25
8.5 count(arg1)	25
8.6 distinct (arg1)	25
8.7 median (arg1)	25

1.目的

本文档旨在说明数尊塔系统支持的所有函数、运算符的功能，参数详解，以及使用举例。

2.注意事项

- 在系统中使用函数需要注意函数的返回值类型与接收的字段类型匹配，否则，可能会引起执行失败。
- 本文档中前后带@的字符串，表示该函数使用了系统中定义的字段。
- 函数传参注意事项，已在函数参数详解中说明，使用时请关注。
- 函数中的所有符号必须为英文，暂不支持中文符号。
- 返回类型为布尔型的函数，如果把计算结果赋值给字符型或者数值型字段，则返回为 true 时取值为 1，返回为 false 时取值为 0。
- 数组函数仅用于二维数组参数的加工计算，不适用于普通字段的加工计算。
- 除数组函数外其他函数均不适用于数组类型的数据加工计算。

3.函数的适用范围

- 衍生字段使用公式衍生的时候，可能会用到函数。
- 评分卡的维度计算得分的时候，可能会用到函数。
- 决策选项节点使用决策公式的时候可能会用到函数。

4.字符型函数

4.1 charAt(source,index)

功能：返回 source 字符串的 index 索引处的字母

参数详解：

source：需要操作的字符串

index：需要截取的字符位置，从 0 开始计算

返回值类型：字符型

使用举例：

eg1: charAt('hello',0) 返回结果为'h';

eg2: charAt(@用户姓名@,0) 运行输入用户姓名为'张俊杰'，运算结果为获取到用户的姓氏'张';

4.2 compareTo(arg1,arg2)

功能：字符串比较函数，大小写敏感。返回 0 则 arg1 与 arg2 完全一样，其余值表示不同。

参数详解：

arg1：被比较的字符串

arg2：用来比较的字符串

返回值类型：数值型，取整数

使用举例：

eg1: compareTo('hello','hello')返回结果为 0;

eg2: compareTo(@身份证归属地@,@手机号归属地@) 运行输入身份证归属地为'北京市'，手机号归属地为'北京市'，函数返回结果为 0;

4.3 compareToIgnoreCase(arg1,arg2)

功能：字符串比较函数，忽略大小写。返回 0 则 arg1 与 arg2 忽略大小写后完全一样，其余值表示不同。

参数详解：

arg1：被比较的字符串

arg2：用来比较的字符串

返回值类型：数值型，取整数

使用举例：

eg1: compareToIgnoreCase('Hello','hello')返回结果为 0;

eg2: compareToIgnoreCase(@用户姓名@,@曾用名@) 运行输入身份证归属地为'sunshine'，手机号归属地为'SunShine'，函数返回结果为 0;

4.4 concat(arg1,arg2)

功能：拼接两个字符串

参数详解：

arg1: 被拼接的字符串

arg2: 用来拼接的字符串

返回值类型：字符型

使用举例：

eg1: concat('Hello','World')返回结果为'HelloWorld'；

eg2: concat (@用户姓氏@,@用户名@) 运行输入用户姓氏为'张'，用户名为'强强'，函数返回结果为'张强强'；

4.5 contains(arg1,arg2)

功能：判断字符串 arg1 是否包含字符串 arg2，大小写敏感。返回 true，arg1 包含 arg2；返回 false，表示不包含。

参数详解：

arg1: 被比较字符串

arg2: 用来比较的字符串

返回值类型：布尔型

使用举例：

eg1: contains('Hello world','world')返回结果为 true；

eg2: contains('world','Hello world')返回结果为 false；

eg3: contains(@频繁出现的 top2 城市@,@身份证归属城市@)运行输入频繁出现的 top2 城市为 '北京、上海'，身份证归属城市 '上海'，函数返回结果为 true；

4.6 notContains(arg1,arg2)

功能：判断字符串 arg1 是否不包含字符串 arg2，大小写敏感。返回 false，arg1 不包含 arg2；返回 true，表示包含。

参数详解：

arg1: 被比较字符串

arg2: 用来比较的字符串

返回值类型：布尔型

使用举例：

eg1: notContains ('Hello world', 'world')返回结果为 false；

eg2: notContains ('world','Hello world')返回结果为 true；

eg3: notContains (@频繁出现的 top2 城市@,@身份证归属城市@)运行输入频繁出现的 top2 城市为 '北京、上海'，身份证归属城市 '上海'，函数返回结果为 false；

4.7 endsWith(arg1,arg2)

功能：判断字符串 **arg1** 是否以字符串 **arg2** 结尾，大小写敏感。返回 **true**，表示 **arg1** 以 **arg2** 结尾；返回 **false** 表示否。

参数详解：

arg1：被比较字符串

arg2：用来比较的字符串

返回值类型：布尔型

使用举例：

eg1: `endsWith('Hello world','world')`返回结果为 **true**;

eg2: `endsWith (@频繁出现的 top2 城市@,@身份证归属城市@)`运行输入频繁出现的 top2 城市为 '北京、上海'，身份证归属城市 '上海'，函数返回结果为 **true**;

4.8 startsWith(arg1,arg2,arg3)

功能：判断字符串 **arg1** 是否从索引位置 **arg3** 处开始以字符串 **arg2** 开始，大小写敏感。返回 **true**，表示 **arg1** 以 **arg2** 开头；返回 **false** 表示否。

参数详解：

arg1：被比较字符串

arg2：用来比较的字符串

arg3：索引位置，从 0 开始计算

返回值类型：布尔型

使用举例：

eg1: `startsWith('Hello world','Hello',0)`返回结果为 **true**;

eg2: `startsWith (@频繁出现的 top2 城市@,@身份证归属城市@,0)`运行输入频繁出现的 top2 城市为 '北京、上海'，身份证归属城市 '上海'，函数返回结果为 **false**;

4.9 equals(arg1,arg2)

功能：判断字符串 **arg1** 是否与字符串 **arg2** 内容相等，大小写敏感。返回结果为 **true** 表示两个字符串相等，返回结果为 **false**，表示两个字符串不相等。

参数详解：

arg1：被比较字符串

arg2：用来比较的字符串

返回值类型：布尔型

使用举例：

eg1: `equals('Hello world','Hello')`返回结果为 **false**;

eg2: `equals (@频繁出现的城市@,@身份证归属城市@)`运行输入频繁出现的 top2 城市为 '上海'，身份证归属城市 '上海'，函数返回结果为 **true**;

4.10 notEquals(arg1,arg2)

功能：判断字符串 **arg1** 是否与字符串 **arg2** 内容不相等，大小写敏感。返回结果为 **true** 表示两个字符串不相等，返回结果为 **false**，表示两个字符串相等。

参数详解：

arg1：被比较字符串

arg2：用来比较的字符串

返回值类型：布尔型

使用举例：

eg1: `notEquals('Hello world','Hello')`返回结果为 **true**;

eg2: `notEquals(@频繁出现的城市@,@身份证归属城市@)`运行输入频繁出现的 top2 城市为 '上海'，身份证归属城市 '上海'，函数返回结果为 **false**;

4.11 equalsIgnoreCase(arg1,arg2)

功能：判断字符串 **arg1** 是否与字符串 **arg2** 内容相等，忽略大小写。返回结果为 **true** 表示两个字符串相等，返回结果为 **false**，表示两个字符串不相等。

参数详解：

arg1：被比较字符串

arg2：用来比较的字符串

返回值类型：布尔型

使用举例：

eg1: `equalsIgnoreCase('Hello','hello')`返回结果为 **true**;

eg2: `equalsIgnoreCase (@频繁出现的城市@,@身份证归属城市@)`运行输入频繁出现的 top2 城市为 'London'，身份证归属城市 'London'，函数返回结果为 **true**;

4.12 age(arg1)

功能：结合当前时间，根据身份证号获取年龄，**arg1** 固定为 18 位身份证号。

参数详解：

arg1：身份证号字符串，例如 612523199006204050

返回值类型：数值型，取整数

使用举例：

eg1: `age('612523199006204050')`若当前时间为 2018-03-16，返回结果为：27，此结果只对当前执行有效；

eg2: `age(@身份证号@)`，若当前时间为 2018-03-16，试运行输入身份证号为 '110324198803094556'，返回结果为 30，此结果只对当前执行有效；

4.13 left(arg1,agr2)

功能：从前往后截取 **arg1** 字符串的前 **arg2** 位，返回结果为截取的字符串。

参数详解:

arg1: 被截取字符串

arg2: 需要截取的位数 (必须 \leq arg1 参数的长度)

返回值类型: 字符型

使用举例:

eg1: `left('abcd',3)`返回结果为: `abc`;

eg2: `left(@家庭住址@,3)`,试运行输入家庭住址为'陕西省西安市莲湖区', 返回结果为'陕西省';

4.14 mid(arg1,arg2,arg3)

功能: 截取字符串 arg1 的第 arg2 位到第 arg3 位。

参数详解:

arg1: 被截取的字符串

arg2: 截取的起始位置 (取值必须 ≥ 0 且 \leq arg1 参数的长度)

arg3: 截取的结束位置, 不包含结束位 (取值必须 \leq arg1 参数的长度, 且 \geq arg2)

返回值类型: 字符型

使用举例:

eg1: `mid('abcd',0,3)`返回结果为: `abc`;

eg2: `mid(@家庭住址@,3,6)`,试运行输入家庭住址为'陕西省西安市莲湖区', 返回结果为'西安市';

4.15 right(arg1,agr2)

功能: 从后往前截取字符串 arg1 的后 arg2 位。

参数详解:

arg1: 被截取的字符串

arg2: 需要截取的位数 (必须 \leq arg1 参数的长度)

返回值类型: 字符型

使用举例:

eg1: `right('abcd',3)`返回结果为: `bcd`;

eg2: `right(@家庭住址@,3)`, 试运行输入家庭住址为'陕西省西安市莲湖区', 返回结果为'莲湖区';

4.16 toNumber(arg1)

功能: 数字字符串转化成数值类型

参数详解:

arg1: 数字字符串

返回值类型: 数值型

使用举例:

eg1: `toNumber('100')+1` 返回结果为: `101`;

eg2: toNumber(substring(@身份证号@,6,10)),试运行输入身份证号为'110324198803094556', 返回结果为数值型的出生年份 1988;

4.17 trim(arg1)

功能: 去除字符串 **arg1** 开头和结尾的空格

参数详解:

arg1: 需要处理的字符串

返回值类型: 字符型

使用举例:

eg1: trim('abcd')返回结果为: abcd;

eg2: trim(@家庭住址@),试运行输入家庭住址为' 陕西省西安市莲湖区', 返回结果为'陕西省西安市莲湖区';

4.18 toUpperCase(arg1)

功能: 字符串 **arg1** 中所有字母转化为大写。

参数详解:

arg1: 需要处理的字符串

返回值类型: 字符型

使用举例:

eg1: toUpperCase('abCD')返回结果为: ABCD;

eg2: toUpperCase(@用户评级@), 试运行输入用户评级'a', 返回结果为'A';

4.19 toLowerCase(arg1)

功能: 字符串 **arg1** 的所有字母转化为小写。

参数详解:

arg1: 需要处理的字符串字符串

返回值类型: 字符型

使用举例:

eg1: toLowerCase('abCD')返回结果为: abcd;

eg2: toUpperCase(@用户评级@), 试运行输入用户评级'aA', 返回结果为'aa';

4.20 substring(arg1,arg2,arg3)

功能: 截取字符串 **arg1** 的第 **arg2** 位到第 **arg3** 位。

参数详解:

arg1: 被截取的字符串

arg2: 截取的起始位置 (必须 ≥ 0 且 \leq arg1 参数的长度)

arg3: 截取的结束位置 (必须 \leq arg1 参数的长度, 且 \geq arg2)

返回值类型: 字符型

使用举例：

eg1: `substring('abcd',0,3)`返回结果为: `abc`;

eg2: `substring(@身份证号@,6, 10)`, 试运行输入身份证号为'110324198803094556', 返回结果为字符型的出生年份'1988';

4.21 `replace(arg1,arg2,arg3)`

功能：将字符串 `arg1` 中的字符串 `arg2` 替换成字符串 `arg3`。

参数详解：

`arg1`: 需要处理的字符串

`arg2`: 需要替换的字符串

`arg3`: 替换后的字符串

返回值类型：字符型

使用举例：

eg1: `replace('abcde','abc','aaa')`返回结果'`aaade`';

eg2: `replace(@毕业院校@,'学院', '学校')`, 试运行输入毕业院校为'蓝翔技术学院', 运行结果为'蓝翔技术学校';

4.22 `length(arg1)`

功能：获取字符串 `arg1` 的长度。

参数详解：

`arg1`: 字符串

返回值类型：数值型，取整数

使用举例：

eg1: `length('abc')`, 返回结果: 3

eg2: `length(@身份证号@)`, 试运行输入'110324198803094556', 运行结果为 18;

4.23 `lastIndexOf(arg1,arg2)`

功能：从字符串 `arg1` 中匹配字符串 `arg2`, 返回最后一个匹配上的位置索引。

参数详解：

`arg1`: 被比较的字符串

`arg2`: 用来比较的字符串

返回值类型：数值型，取整数

使用举例：

eg1: `lastIndexOf('abcdefge','e')`, 返回结果: 7

eg2: `lastIndexOf('abcdefge','h')`, 返回结果: -1

eg3: `lastIndexOf(@身份证号@,'6')`, 试运行输入身份证号'110326198603094556', 运行结果为 17;

4.24 indexOf(arg1,arg2,arg3)

功能:字符串 arg1 从 arg3 的索引位置开始匹配字符串 arg2,返回匹配到的第一个索引位置,返回-1 表示没有匹配到。

参数详解:

arg1: 被匹配的字符串字符串

arg2: 用例匹配的字符串

arg3: 开始匹配的索引位置 (必须<=arg1 参数的长度)

返回值类型: 数值型, 取整数

使用举例:

eg1: indexOf('abdefg','e',0), 返回结果: 2;

eg2: indexOf('@身份证号@','6',0), 试运行输入身份证号'110326198603094556', 运行结果为 5;

5.数值型函数

5.1 abs(arg1)

功能：求数值型数据的绝对值

参数详解：

arg1: 一个数值

返回值类型：数值型，与输入参数精度一致

使用举例：

eg1: abs(-1.15) 返回结果为 1.15;

eg2: abs(10.2) 返回结果为 10.2

5.2 acos(arg1)

功能：反余弦函数，以 -1 到 1 的一个数为参数，返回一个 0 到 π （弧度）的数值

参数详解：

arg1: 一个数值

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

eg1: acos(-1) 返回值为 3.1416

eg2: acos(1) 返回值为 0

5.3 cos(arg1)

功能：余弦函数，是三角函数中的一员，返回某一角度的余弦值

参数详解：

arg1: number 为需要求余弦值的一个角度，必须用弧度表示。如果 number 的单位是度，可以乘以 $\pi()/180$ 转换为弧度。

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

使用举例：

eg1: cos(1) 返回值为 0.5403

eg2: cos(0) 返回值为 1

5.4 asin(arg1)

功能：反正弦函数，以 -1 到 1 之间的数值作为参数，返回一个介于 $-\pi/2$ 到 $\pi/2$ 弧度的数值

参数详解：

arg1: 一个数值

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

使用举例：

eg1: `asin(1)` 返回值为 1.5708

eg2: `asin(-1)` 返回值为 -1.5708

5.5 `sin(arg1)`

功能：正弦函数，是三角函数中的一员，返回某一角度的正弦值

参数详解：

arg1: `number` 为需要求正弦值的一个角度，必须用弧度表示。如果 `number` 的单位是度，可以乘以 $\pi()/180$ 转换为弧度。

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

使用举例：

eg1: `sin(0)` 返回值为 0

eg2: `sin(90)` 返回值为 0.894

5.6 `atan(arg1)`

功能：反正切值函数，返回 $-\pi/2$ 到 $\pi/2$ 弧度之间的数值

参数详解：

arg1: 一个数值

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

使用举例：

eg1: `atan(10)` 返回值 1.4711

eg2: `atan(-1)` 返回值 -0.7854

5.7 `tan(arg1)`

功能：正切值函数，是三角函数中的一员，返回某一角度的正切值

参数详解：

arg1: `number` 为需要求正切值的一个角度，必须用弧度表示。如果 `number` 的单位是度，可以乘以 $\pi()/180$ 转换为弧度。

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

使用举例：

eg1: `tan(-10)` 返回值 0.6484

eg2: `tan(10)` 返回值 -0.6484

5.8 exp(arg1)

功能：以自然常数 e 为底的指数函数

参数详情：

arg1: 一个数值

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

使用举例：

eg1: exp(10) 返回值 22026.4658

eg2: exp(0) 返回值 1

5.9 avg(arg1,arg2,.....)

功能：求多个数值的平均值

参数详情：

arg1: 需要求平均数的第一个数值

arg2: 需要求平均数的第二个数值

.....

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

使用举例：

eg1: avg(1,2,3,4) 返回值 2.5

eg1: avg(1,2,3,-4) 返回值 0.5

5.10 ceil(arg1)

功能：向上取整，返回大于或者等于指定表达式的最小整数

参数详情：

arg1: 一个数值

返回值类型：数值型，取整数

使用举例：

eg1: ceil(-7.004) 返回值 -7

eg2: ceil(7.004) 返回值 8

5.11 floor(arg1)

功能：向下取整,或者说“向下舍入”,返回小于或等于指定数字的最大整数的数值

参数详情：

arg1: 一个数值

返回值类型：数值型，取整数

使用举例：

eg1: floor(45.95) 返回值 45

eg2: floor(-45.95) 返回值 -46

5.12 round(arg1)

功能：在原来数值上四舍五入取整

参数详情：

arg1: 一个数值

返回值类型：数值型，取整数

使用举例：

eg1: round(-1.1) 返回值 -1

eg2: round(-1.6) 返回值 -2

5.13 ln(arg1)

功能：取自然对数（以 e 为底）

参数详情：

arg1: 一个数值，取值(0,+)

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

使用举例：

eg1: ln(10) 返回值 2.3026

eg2: ln(0.1) 返回值 -2.3026

5.14 log(arg1)

功能：以 10 为底的对数函数

参数详情：

arg1: 10 的倍数

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

使用举例：

eg1: log(10) 返回值 1

eg2: log(13) 返回值 1.1139

5.15 max(arg1,arg2,.....)

功能：求多个数值的最大值

参数详情：

arg1: 求最大值的第一个数值

arg2: 求最大值的第二个数值

.....

返回值类型: 数值型, 不改变精度, 与输入数据一致

使用举例:

eg1: max(0.23456,-45678.987,-1.234567) 返回值 0.23456

5.16 min(arg1,arg2,.....)

功能: 求多个数值的最小值

参数详情:

arg1: 求最小值的第一个数值

arg2: 求最小值的第二个数值

.....

返回值类型: 数值型, 不改变精度, 与输入数据一致

使用举例:

eg1: max(0.23456,-45678.987,-1.234567) 返回值 -45678.987

5.17 pow(arg1,arg2)

功能: 返回数值乘幂的结果

参数详情:

arg1: 幂函数底数

arg2: 幂函数指数

返回值类型: 数值型, 计算结果小数位大于四位, 则四舍五入保留 4 位小数; 小于 4 位则原样输出。

使用举例:

eg1: pow(10,2) 返回值 100

eg2: pow(-10,-2) 返回值 0.01

5.18 random()

功能: 随机数, 一个浮点型伪随机数字在 0 (包括 0) 和 1 (不包括) 之间

参数详情:

该指标不用输入参数

返回值类型: 数值型, 精度为 4 位

使用举例:

eg1: rand() 返回值 0.0509

5.19 rint(arg1)

功能: 函数返回最接近参数的整数, 如果有 2 个数同样接近, 则会返回偶数的那个
参数详情:

arg1: 一个数值
返回值类型: 数值型, 取整数
使用举例:
eg1: rint(-1.5) 返回值 -2
eg2: rint(0.1) 返回值 0

5.20 sqrt(arg1)

功能: 数值的平方根计算
参数详情:
arg1: 一个数值, 范围 $[0, +\infty)$
返回值类型: 数值型, 计算结果小数位大于四位, 则四舍五入保留 4 位小数; 小于 4 位则原样输出。
使用举例:
eg1: sqrt(4) 返回值 2
eg2: sqrt(3) 返回值 1.7321

5.21 sum(arg1,arg2,.....)

功能: 计算数值的和
参数详情:
arg1: 需要求和的第一个数值
arg2: 需要求和的第二个数值
.....
返回值类型: 数值型, 计算结果小数位大于四位, 则四舍五入保留 4 位小数; 小于 4 位则原样输出。
使用举例:
eg1: sum(1,2,3) 返回值 6
eg2: sum(1.0092,2,3) 返回值 6.0092

5.22 toDegrees(arg1)

功能: 将以弧度测量的角度转换为以度为单位的近似等效角度
参数详情:
arg1: $-\pi/2$ 到 $\pi/2$ 弧度
返回值类型: 数值型, 计算结果小数位大于四位, 则四舍五入保留 4 位小数; 小于 4 位则原样输出。
使用举例:
eg1: toDegrees(45)返回值 2578.3101
eg1: toDegrees(0)返回值 0

5.23 toRadians(arg1)

功能：将以度为单位的角度转换为以弧度测量的大致相等的角度

参数详情：

arg1: 一个数值

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

使用举例：

eg1: toRadians (45)返回值 0.7854

eg2: toRadians (0) 返回值 0

6.日期函数

6.1 dateif(arg1,arg2,'Y')

功能：计算两个日期期间的整数年，不够 1 年不计入。

参数详解：

arg1：日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

arg2：日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

返回值类型：数值型，取整数

使用举例：

eg1: dateif('2017-03-15','2018-04-16','Y') 返回结果为: 1;

eg2: dateif('2017-03-15','2016-02-16','Y') 返回结果为: 1;

eg3: dateif(@申请日期@, @审批日期@, 'Y') ，试运行输入申请日期为'2017-03-15'，审批日期为'2018-04-16'，返回结果为 1。

6.2 dateif(arg1,arg2,'M')

功能：计算两个日期期间的整数月，不够 1 月不计入。

参数详解：

arg1：日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

arg2：日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

返回值类型：数值型，取整数

使用举例：

eg1: dateif('2017-03-15','2018-04-16','M') 返回结果为 13;

eg2: dateif(@申请日期@, @审批日期@, 'M') ，试运行输入申请日期为'2017-03-15'，审批日期为'2018-04-16'，返回结果为 13。

6.3 dateif(arg1,arg2,'D')

功能：计算两个日期之间的天数差。

参数详解：

arg1：日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

arg2：日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

返回值类型：数值型，取整数

使用举例：

eg1: dateif('2018-03-15','2018-04-15','D') 返回结果为 396;

eg2: dateif(@申请日期@, @审批日期@, 'D') ，试运行输入申请日期为'2017-03-15'，审批日期为'2018-04-15'，返回结果为 396。

6.4 dateif(arg1,arg2,'YM')

功能：计算两个日期中天数差、忽略日期中的月和年

参数详解：

arg1: 日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

arg2: 日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

返回值类型：数值型，取整数

使用举例：

eg1: dateif('2017-03-15','2018-04-20','YM') 返回结果为：5；

eg2: dateif(@申请日期@, @审批日期@, 'YM') ，试运行输入申请日期为'2017-03-15'，审批日期为'2018-04-20'，返回结果为 5。

6.5 dateif(arg1,arg2,'YD')

功能：计算两个日期中月数差、忽略日期中的日和年；

参数详解：

arg1: 日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

arg2: 日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

返回值类型：数值型，取整数

使用举例：

eg1: dateif('2017-03-15','2018-05-20','YD') 返回结果为：2；

eg2: dateif(@申请日期@, @审批日期@, 'YD') ，试运行输入申请日期为'2017-03-15'，审批日期为'2018-05-20'，返回结果为 2。

6.6 dateif(arg1,arg2,'MD')

功能：计算两个日期的年份差、忽略日期中的月和日

参数详解：

arg1: 日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

arg2: 日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

返回值类型：数值型，取整数

使用举例：

eg1: dateif('2017-03-15','2018-05-20','MD') 返回结果为：1；

eg2: dateif(@申请日期@, @审批日期@, 'MD') ，试运行输入申请日期为'2017-03-15'，审批日期为'2018-05-20'，返回结果为 1。

6.7 dateif(arg1,arg2,'DD')

功能：忽略天，计算两个日期期间的月份差

参数详解：

arg1: 日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

arg2: 日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

返回值类型: 数值型, 取整数

使用举例:

eg1: dateif('2017-03-15','2018-05-20','DD') 返回结果为: 14;

eg2: dateif(@申请日期@, @审批日期@, 'DD') , 试运行输入申请日期为'2017-03-15', 审批日期为'2018-05-20', 返回结果为 14。

6.7 day(arg1)

功能: 获取 arg1 的当前所在天;

参数详解:

arg1: 日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

返回值类型: 数值型, 取整数

使用举例:

eg1: day('2018-03-15') 返回结果为 15;

eg2: day('2018-03-15 12:30:00') 返回结果为 15;

eg3: day(@申请日期@) , 试运行输入申请日期为'2018-03-15', 返回结果为 15。

6.8 month(arg1)

功能: 获取 arg1 的当前所在月;

参数详解:

arg1: 日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

返回值类型: 数值型, 取整数

使用举例:

eg1: month('2018-03-15') 返回结果为 3;

eg2: month('2018-03-15 12:30:00') 返回结果为 3;

eg2: month(@申请日期@) , 试运行输入申请日期为'2018-03-15', 返回结果为 3。

6.9 now()

功能: 获取当前时间;

参数详解: 无参数

返回值类型: 日期型, 取值格式为 yyyy-MM-dd HH:mm:ss

使用举例:

eg1: now() 返回结果为 2018-03-15 11:55:31。

6.10 today()

功能: 获取当前日期;

参数详解: 无参数

返回值类型：日期型，取值格式为 yyyy-MM-dd

使用举例：

eg1: today() 返回结果为 2018-03-15;

6.11 year(arg1)

功能：获取 arg1 的当前所在年份;

参数详解：

arg1: 日期格式字符串(yyyy-MM-dd 或 yyyy-MM-dd HH:mm:ss)

返回值类型：数值型，取整数

使用举例：

eg1: year('2018-03-15') 返回结果为 2018;

eg2: year(@申请日期@) ， 试运行输入申请日期为'2018-03-15'，返回结果为 2018。

7.运算符

7.1 数学运算符

系统支持的数学运算符有：+，-，*，/，%

运算符	说明	示例
+	加	编辑公式 “@决策额度@+@年龄盖帽@”；输入决策额度=20000，年龄盖帽=500；输出结果=20500
-	减	编辑公式 “@决策额度@-@年龄盖帽@”；输入决策额度=20000，年龄盖帽=500；输出结果=19500
*	乘	编辑公式 “@决策额度@*@年龄盖帽@”；输入决策额度=20000，年龄盖帽=1.5；输出结果=30000
/	除	编辑公式 “@决策额度@/@年龄盖帽@”；输入决策额度=20000，年龄盖帽=2；输出结果=10000
%	取余	编辑公式 “@决策额度%@%@年龄盖帽@”；输入决策额度=20003，年龄盖帽=10000；输出结果=3

7.2 关系运算符

系统支持的关系运算符有：==，!=，>，>=，<，<=

使用场景：决策选项节点的决策公式；评分卡选择枚举型输出字段时；枚举型衍生字段配置衍生条件时，运算结果为真时输出对应的枚举型值。

运算符	说明	示例
==	等于	编辑公式 “@决策额度@==@基础额度@”；输入决策额度=20000，基础额度=500；运算结果为逻辑假
!=	不等于	编辑公式 “@决策额度@!=@基础额度@”；输入决策额度=20000，基础额度=500；运算结果为逻辑真
>	大于	编辑公式 “@决策额度@>@基础额度@”；输入决策额度=20000，基础额度=500；运算结果为逻辑真
<	小于	编辑公式 “@决策额度@<@基础额度@”；输入决策额度=20000，基础额度=500；运算结果为逻辑假
>=	大于等于	编辑公式 “@决策额度@>=@基础额度@”；输入决策额度=20000，基础额度=500；运算结果为逻辑真
<=	小于等于	编辑公式 “@决策额度@<=@基础额度@”；输入决策额度=20000，基础额度=500；运算结果为逻辑假

7.3 逻辑运算符

系统支持的逻辑运算符有：&&，||，!

使用场景：决策选项节点的决策公式；评分卡选择枚举型输出字段时；枚举型衍生字段配置衍生条件时，运算结果为真时输出对应的枚举型值

运算符	说明	示例
&&	与	编辑公式 “@年龄@>18 && @年龄@<55”；输入年龄=20；运算结果为逻辑真
	或	编辑公式 “equals(@学历@,'本科') equals(@学历@,'研究生')”；输入学历= ‘本科’；运算结果为逻辑真
!	非	编辑公式 “! equals(@学历@,'本科')”；输入学历= ‘本科’；运算结果为逻辑假

8.数组函数

本节中所有数组函数使用举例均使用以下二维数组：

数组名称：array1

包含参数：

int_pram1, 数值型；
str_pram1, 字符型；
date_pram1, 日期型；

8.1 gavg(arg1)

功能：求二维数组中某一个数值型参数的平均值

参数详情：

arg1: 二维数组中需要求平均数的参数名称，仅数值参数

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

使用举例：

eg1: gavg(@int_pram1@), 二维数组 array1 的参数 int_pram1 的取值如果为：100,-1,1,0；则计算结果为 25。

8.2 gmax(arg1)

功能：求二维数组中某一个数值型参数的最大值

参数详情：

arg1: 二维数组中需要求最大值的参数名称，仅数值参数

返回值类型：数值型，不改变精度，与输入数据一致

使用举例：

eg1: gmax(@int_pram1@), 二维数组 array1 的参数 int_pram1 的取值如果为：100,-1,1,0；则计算结果为 100。

8.3 gmin(arg1)

功能：求二维数组中某一个数值型参数的最小值

参数详情：

arg1: 二维数组中需要求最小值的参数名称，仅数值参数

返回值类型：数值型，不改变精度，与输入数据一致

eg1: gmin(@int_pram1@), 二维数组 array1 的参数 int_pram1 的取值如果为：100,-1,1,0；则计算结果为-1。

8.4 gsum(arg1)

功能：求二维数组中某一个数值型参数的和

参数详情：

arg1: 维数组中要求和的参数名称，仅数值参数

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

eg1: gsum(@int_pram1@)，二维数组 array1 的参数 int_pram1 的取值如果为：100,-1,1,0；则计算结果为 100。

8.5 count(arg1)

功能：对二维数组中某一个参数进行计数

参数详情：

arg1: 维数组中需要计数的参数名称，支持所有参数类型

返回值类型：数值型，取整数

eg1: count(@date_pram1@)，二维数组 array1 的参数 date_pram1 的取值如果为：2016-09-01,2016-09-01 09:00:00,,2016-09-01；则计算结果为 4。

8.6 distinct (arg1)

功能：去重统计，对二维数组中某一个参数进行计数，重复数据不计入

参数详情：

arg1: 维数组中需要去重计数的参数名称，支持所有参数类型

返回值类型：数值型，整数

eg1: distinct(@date_pram1@)，二维数组 array1 的参数 date_pram1 的取值如果为：2016-09-01,2016-09-01 09:00:00,,2016-09-01；则计算结果为 3。

8.7 median (arg1)

功能：求二维数组中某一个数值型参数的中位

参数详情：

arg1: 维数组中要求中位数的参数名称，仅数值参数

返回值类型：数值型，计算结果小数位大于四位，则四舍五入保留 4 位小数；小于 4 位则原样输出。

eg1: median(@int_pram1@)，二维数组 array1 的参数 int_pram1 的取值如果为：100,-1,1,0；则计算结果为 0.5。

中位数：满足条件的数组数据，某列字段取值按照由小到大排序，找出最中间的一个取值；如果满足条件的取值个数为偶数，则取最中间的两个取值的平均值。