

# docoGen markdown support

kevin

lu

kevinbir61@gmail.com

lu@gmail.com

November 23, 2017

## Abstract

## 1 Text format demo

None type demo

sentence 1

sentence 2

Text type demo

- sentence 1

- sentence 2

## 2 List format demo

list type demo

1. list 1

2. list 2

- (a) list 2-1

- i. list 2-1-1

- ii. list 2-1-2

- (b) list 2-2

3. list 3

4. list 4

### 3 Table format demo

table type demo		
User Name	Age	Job
Kevin	20	Student
Lu	22	Engineer
Eric	31	Programmer

### 4 Formula format demo

formula type demo

A simple equation!  $x^2 + y^2 = z^2$

$$x^n + y^n = z^n$$

$$E = mc^2 \quad (1)$$

### 5 Code format demo

code type demo

```
1 #include<stdio.h>\n\nint main() {\n\treturn 0;\n}\n
```

Listing 1: Simple Demo C program

```
1 import numpy as np
2
3 def incmatrix(genl1, genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None #to become the incidence matrix
7     VT = np.zeros((n*m,1), int) #dummy variable
8
9     #compute the bitwise xor matrix
10    M1 = bitxormatrix(genl1)
11    M2 = np.triu(bitxormatrix(genl2),1)
12
13    for i in range(m-1):
14        for j in range(i+1, m):
15            [r, c] = np.where(M2 == M1[i, j])
16            for k in range(len(r)):
17                VT[(i)*n + r[k]] = 1;
18                VT[(i)*n + c[k]] = 1;
19                VT[(j)*n + r[k]] = 1;
20                VT[(j)*n + c[k]] = 1;
21
22    if M is None:
```

```

23         M = np.copy(VT)
24     else:
25         M = np.concatenate((M, VT), 1)
26
27     VT = np.zeros((n*m, 1), int)
28
29     return M

```

Listing 2: Simple Demo Python program

## 6 Figure format demo

figure type demo

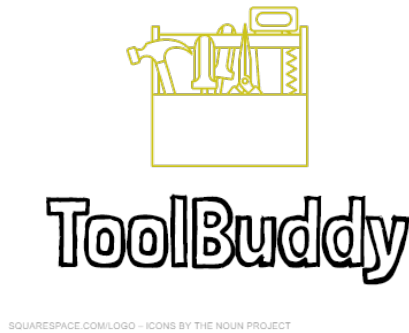


Figure 1: toolbuddy

## 7 Graphviz format demo

Graphviz

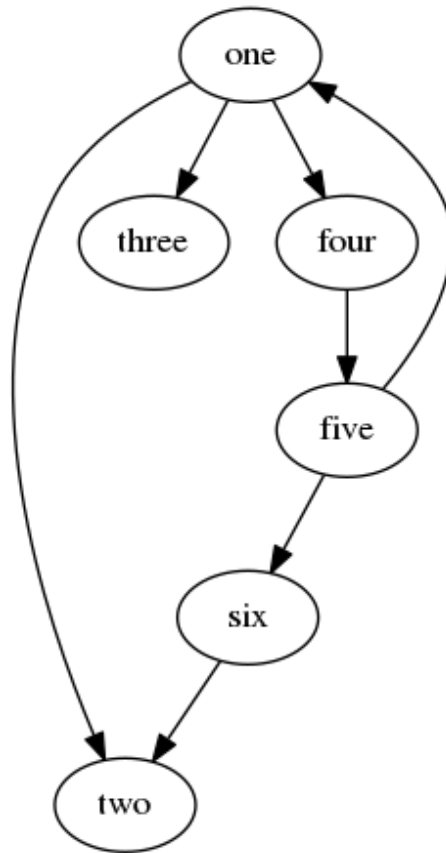


Figure 2: graphviz src demo

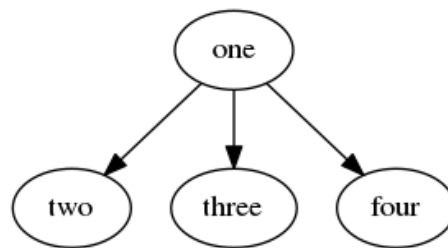


Figure 3: graphviz raw demo

## 8 Restful API Support

restful-api type demo

**[Online] Register New User**

**Method:** post  
**Url:** https  
**Description:** Enroll new user to local service  
**Parameter:**

Field Name	Data Type
username	string
username	string

**Error Msg:** duplicated internal error  
**Success Msg:** passed noped

## 9 Outside Figure + SubArticle Test

Text Default

- sentence 3
- sentence 4

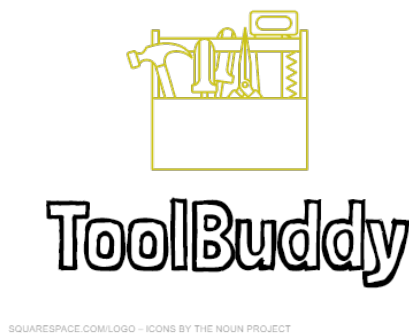


Figure 4: caption of image

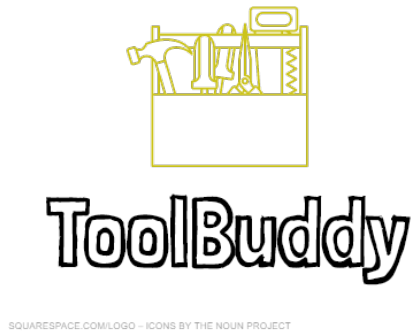


Figure 5: caption of image

## 9.1 Nested

Nested Text

- sentence nested 1
- sentence nested 2

Nested None

sentence nested 3  
sentence nested 4

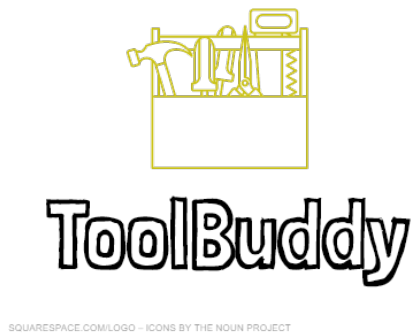
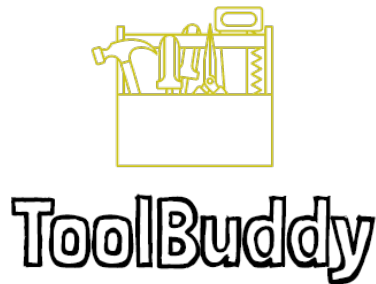


Figure 6: caption of image



SQUARESPACE.COM/LOGO - ICONS BY THE NOUN PROJECT

Figure 7: caption of image

### 9.1.1 Nested Nested

Nested Nested Text

- sentence nested nested 1
- sentence nested nested 2

### 9.1.2 Nested Nested Code Block

Code Block

```
1  /* Prime modulus multiplicative linear congruential generator
2  Z[i] = (630360016 * Z[i-1]) (mod(pow(2,31) - 1)), based on Marse
   and Roberts'
3  portable FORTRAN random-number generator UNIRAN. Multiple (100)
   streams are
4  supported, with seeds spaced 100,000 apart. Throughout, input
   argument
5  "stream" must be an int giving the desired stream number. The
   header file
6  lcgrand.h must be included in the calling program (#include "
   lcgrand.h")
7  before using these functions.
8
9  Usage: (Three functions)
10
11  1. To obtain the next U(0,1) random number from stream "stream,"
   execute
12     u = lcgrand(stream);
13     where lcgrand is a float function. The float variable u will
   contain the
14     next random number.
15
```

```

16 2. To set the seed for stream "stream" to a desired value zset,
    execute
17     lcgrandst(zset, stream);
18     where lcgrandst is a void function and zset must be a long
    set to the
19     desired seed, a number between 1 and 2147483646 (inclusive).
    Default
20     seeds for all 100 streams are given in the code.
21
22 3. To get the current (most recently used) integer in the
    sequence being
23     generated for stream "stream" into the long variable zget,
    execute
24     zget = lcgrandgt(stream);
25     where lcgrandgt is a long function. */
26
27 /* Define the constants. */
28
29 #define MODLUS 2147483647
30 #define MULT1 24112
31 #define MULT2 26143
32
33 /* Set the default seeds for all 100 streams. */
34
35 static long zrng[] =
36 {
37     1,
38     1973272912, 281629770, 20006270, 1280689831, 2096730329, 1933576050,
39     913566091, 246780520, 1363774876, 604901985, 1511192140, 1259851944,
40     824064364, 150493284, 242708531, 75253171, 1964472944, 1202299975,
41     233217322, 1911216000, 726370533, 403498145, 993232223, 1103205531,
42     762430696, 1922803170, 1385516923, 76271663, 413682397, 726466604,
43     336157058, 1432650381, 1120463904, 595778810, 877722890, 1046574445,
44     68911991, 2088367019, 748545416, 622401386, 2122378830, 640690903,
45     1774806513, 2132545692, 2079249579, 78130110, 852776735, 1187867272,
46     1351423507, 1645973084, 1997049139, 922510944, 2045512870, 898585771,
47     243649545, 1004818771, 773686062, 403188473, 372279877, 1901633463,
48     498067494, 2087759558, 493157915, 597104727, 1530940798, 1814496276,
49     536444882, 1663153658, 855503735, 67784357, 1432404475, 619691088,
50     119025595, 880802310, 176192644, 1116780070, 277854671, 1366580350,
51     1142483975, 2026948561, 1053920743, 786262391, 1792203830, 1494667770,
52     1923011392, 1433700034, 1244184613, 1147297105, 539712780, 1545929719,
53     190641742, 1645390429, 264907697, 620389253, 1502074852, 927711160,
54     364849192, 2049576050, 638580085, 547070247 };
55
56 /* Generate the next random number. */
57
58 float lcgrand(int stream)
59 {
60     long zi, lowprd, hi31;
61
62     zi = zrng[stream];
63     lowprd = (zi & 65535) * MULT1;
64     hi31 = (zi >> 16) * MULT1 + (lowprd >> 16);
65     zi = ((lowprd & 65535) - MODLUS) +
66         ((hi31 & 32767) << 16) + (hi31 >> 15);
67     if (zi < 0) zi += MODLUS;
68     lowprd = (zi & 65535) * MULT2;

```



```

68     hi31    = (zi >> 16) * MULT2 + (lowprd >> 16);
69     zi      = ((lowprd & 65535) - MODLUS) +
70               ((hi31 & 32767) << 16) + (hi31 >> 15);
71     if (zi < 0) zi += MODLUS;
72     zrng[stream] = zi;
73     return (zi >> 7 | 1) / 16777216.0;
74 }
75
76
77 void lcgrandst (long zset, int stream) /* Set the current zrng for
78     stream                                     "stream" to zset. */
79 {
80     zrng[stream] = zset;
81 }
82
83
84 long lcgrandgt (int stream) /* Return the current zrng for stream "
85     stream". */
86 {
87     return zrng[stream];
88 }

```

Listing 3: C example

## References

- [1] kevin, NCKU student