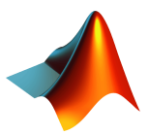


MATLAB 프로그래밍 및 실습

10강. 함수 고급 - part2

(가변 길이 함수 입출력)



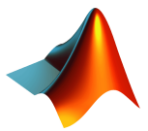
plot 함수 잠깐 복습

	propertyName	의미	값	line specifier?	default
마커	marker	마커 종류	<code>+o*.x^v</code> <code>sdph<></code>	Y	none
	markeredgecolor	마커 테두리 색	<code>rgbmcykw</code> 또는 1x3 벡터	N	auto
	markerfacecolor	마커 면 색		N	none
	markersize	마커 크기	양의 실수	N	6
선	color	선 색	<code>rgbmcykw</code> 또는 1x3 벡터	Y	[0 0.4470 0.7410]
	linestyle	선 스타일	<code>-</code> , <code>--</code> , <code>:</code> , <code>-.</code>	Y	solid (-)
	linewidth	선 두께	양의 실수	N	0.5

```
x = 0:0.1:1;
y = x.^2;
plot(x, y, 'r*- ', 'linewidth', 2, 'markersize', 10, 'markeredgecolor', [0.7 0.7 0.7])
```

- 이걸 사용자 정의 함수로 만들려면...?!

```
function myplot(x, y, lineSpec, prop1, val1, prop2, val2, ...)
```



가변 길이 입력인자 - varargin

varsum(1,2,3,4,5)

{ }

```
+30 varsum.m x
1 function z = varsum(x, y, varargin)
2
3     z = x + y;
4     z = z + sum(cell2mat(varargin));
5
6 end
```

Workspace - varsum		
Name ▲	Value	Size
{ } varargin	1x3 cell	1x3
x	1	1x1
y	2	1x1

* varargin = variable argument input

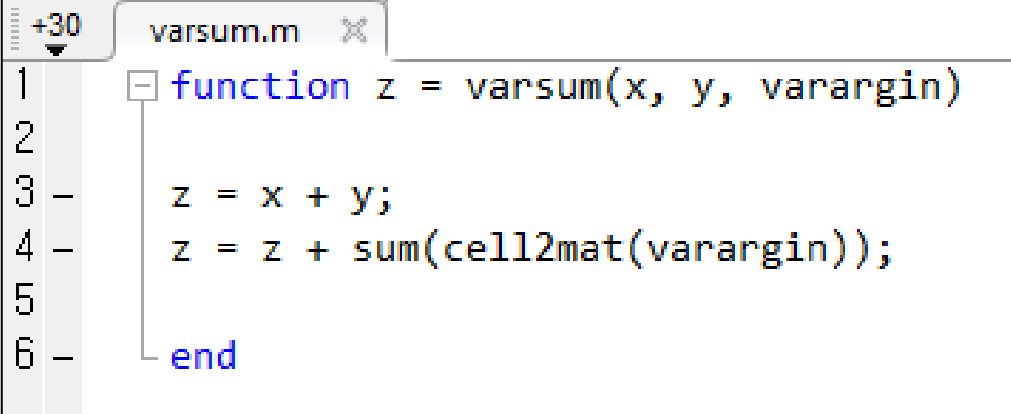
• varargin의 입력이...

- 0, 1개이면?
- 2개이면?
- 3개 이상이면?

varargin 문법

```
function funcname(in1, in2, ..., inN, varargin)
```

- 기본입력인자 – N개
- 함수 호출 시 입력인자 개수가...
 - (입력인자 개수) < N -> 에러 발생
 - (입력인자 개수) == N -> varargin은 빈 셀 배열
 - (입력인자 개수) > N -> N번째 이후의 입력인자는 varargin에 셀 배열로 들어감
- varargin에는 길이 제한이 없음
- 주의할 점
 - 소문자로 정확히 varargin이라고 써야 동작함 (다른 이름 x)
 - varargin은 함수 정의부의 가장 마지막에 와야 함



```
varsum.m x
1 function z = varsum(x, y, varargin)
2
3     z = x + y;
4     z = z + sum(cell2mat(varargin));
5
6 end
```

함수 입력으로 varargin 하나만? 가능하다!

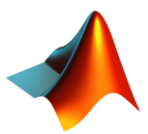
```
>> greeting_all('Hong', 'Hwang', 'Huh')  
Hello, Hong!  
Hello, Hwang!  
Hello, Huh!
```

```
+32  
greeting_all.m  
1 function greeting_all(varargin)  
2  
3 for i=1:length(varargin)  
4     fprintf('Hello, %s!\n', varargin{i});  
5 end  
6  
7 end
```

Workspace - greeting_all		
Name ▲	Value	Size
{ } varargin	1x3 cell	1x3

varargin			
	1	2	3
1	Hong	Hwang	Huh
2			
3			

greeting_all() % 어떻게 될까?



함수 인자에 기본값을 설정할 수 있을까?

```
+34 roll_dices2.m x
1 function nums = roll_dices2(Ndices, Nfaces)
2
3     nums = randi(Nfaces, [1, Ndices]);
4
5 end
```

- 한 종류 주사위 여러 개를 한번에 굴리는 함수
 - Nfaces: 면 개수 (6면체 -> 6)
 - Ndices: 주사위 개수

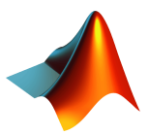
```
>> roll_dices2(5, 6)
ans =
     3     6     4     1     4
>> roll_dices2(3, 6)
ans =
     1     3     4
>> roll_dices2(4, 6)
ans =
     6     4     5     1
```

- 6면체를 기본값으로 설정하고 싶다면?

```
roll_dices2(4, 8) % 8면체 주사위 4개 굴리기
roll_dices2(5) % 6면체 주사위 5개 굴리기
```

- 주사위 개수도 4개를 기본으로 하고 싶다면?

```
roll_dices2 % 6면체 주사위 4개 굴리기
```



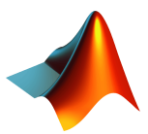
입력 인자의 개수 - nargin

```
+35 roll_dices3.m x
1  function nums = roll_dices3(Ndices, Nfaces)
2
3  -   fprintf('number of input args.: %d\n', nargin)
4
5  -   if nargin<2
6  -       Nfaces = 6;
7  -   end
8  -   if nargin<1
9  -       Ndices = 4;
10 -   end
11
12 -   nums = randi(Nfaces, [1, Ndices]);
13
14 -   end
```

roll_dices3(4, 8)

roll_dices3(5)

roll_dices3



nargin 사용법

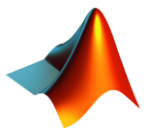
- 함수 호출 시에 입력된 인자 개수를 함수 내부에서 반환
- nargin에 따른 적절한 분기를 통해 기본값 할당 가능
- 함수 내부에 nargin이라는 변수는 없음
- 함수 정의의 varargin 여부와 무관함
 - 함수 "호출" 때의 입력인자 개수

```
function z = varsum(x, y , varargin)
```

varsum()	-> nargin=0
varsum(1)	-> nargin=1
varsum(1,2)	-> nargin=2
varsum(1,2,3)	-> nargin=3
varsum(1,2,3,4,5)	-> nargin=5

- 입력인자의 순서를 바꿀 수는 없음
 - Nfaces를 지정하면서 Ndices 기본값 사용 불가

```
roll_dices3.m
1 function nums = roll_dices3(Ndices, Nfaces)
2
3     fprintf('number of input args.: %d\n', nargin)
4
5     if nargin<2
6         Nfaces = 6;
7     end
8     if nargin<1
9         Ndices = 4;
10    end
11
12    nums = randi(Nfaces, [1, Ndices]);
13
14    end
```



varargin와 nargin를 함께 활용하기

```
roll_dices4('Nfaces',8,'Ndices',5)
```

```
+36 roll_dices4.m x
1 function nums = roll_dices4(varargin)
2
3     inputs.Nfaces = 6;
4     inputs.Ndices = 4;
5
6     for i=1:2:nargin
7         inputs.(varargin{i}) = varargin{i+1};
8     end
9
10    nums = randi(inputs.Nfaces, [1, inputs.Ndices]);
11
12 end
```

varargin =
1x4 cell array
{'Nfaces'} {[8]} {'Ndices'} {[5]}

inputs =
struct with fields:

Nfaces: 6
Ndices: 4

inputs =
struct with fields:

Nfaces: 8
Ndices: 5

```
>> roll_dices4('Ndices',6,'Nfaces',4)
ans =
     3     4     3     3     1     2
>> roll_dices4('Ndices',6)
ans =
     4     3     6     5     2     3
>> roll_dices4()
ans =
     5     4     1     1
```

narginchk

narginchk(minArgs,maxArgs)

↑
입력인자
개수를

↑
확인
해라.

↑
minArgs
이상이고

↑
maxArgs
이하인지

- 입력 0개 가능하다면 -> minArgs = 0
- 입력 최대 개수 미지정 -> maxArgs = inf

```
+36 roll_dices4.m ✕
1 function nums = roll_dices4(varargin)
2
3     narginchk(0,4);
4
5     inputs.Nfaces = 6;
6     inputs.Ndices = 4;
7
8     for i=1:2:nargin
9         inputs.(varargin{i}) = varargin{i+1};
10    end
11
12    nums = randi(inputs.Nfaces, [1, inputs.Ndices]);
13
14 end
```

↔

```
if nargin<minArgs
    error('...')
elseif nargin>maxArgs
    error('...')
end
```

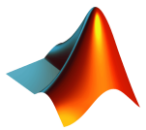
```
>> roll_dices4('Ndices',6,'Nfaces',4)
ans =
     3     4     3     3     1     2
>> roll_dices4('Ndices',6)
ans =
     4     3     6     5     2     3
>> roll_dices4()
ans =
     5     4     1     1
```

```
>> roll_dices4('Ndices',6,'Nfaces',4,6,8)
Error using roll_dices4 (line 3)
Too many input arguments.
```

cylinder, sphere

```
+38 cylinder.m
1 function [xx,yy,zz] = cylinder(varargin)
2 %CYLINDER Generate cylinder.
3 % [X,Y,Z] = CYLINDER(R,N) forms the unit cylinder based on the g
4 % curve in the vector R. Vector R contains the radius at equally
5 % spaced points along the unit height of the cylinder. The cylin
6 % has N points around the circumference. SURF(X,Y,Z) displays th
7 % cylinder.
8 %
9 % [X,Y,Z] = CYLINDER(R), and [X,Y,Z] = CYLINDER default to N = 2
10 % and R = [1 1].
11 %
12 % Omitting output arguments causes the cylinder to be displayed
13 % a SURF command and no outputs to be returned.
14 %
15 % CYLINDER(AX,...) plots into AX instead of GCA.
16 %
17 % See also SPHERE, ELLIPSOID.
18 %
19 % Clay M. Thompson 4-24-91, CBM 8-21-92.
20 % Copyright 1984-2002 The MathWorks, Inc.
21
22 % Parse possible Axes input
23 - narginchk(0,3);
24 - [cax,args,nargs] = axescheck(varargin{:});
```

```
+37 sphere.m
1 function [xx,yy,zz] = sphere(varargin)
2 %SPHERE Generate sphere.
3 % [X,Y,Z] = SPHERE(N) generates three (N+1)-by-(N+1)
4 % matrices so that SURF(X,Y,Z) produces a unit sphere.
5 %
6 % [X,Y,Z] = SPHERE uses N = 20.
7 %
8 % SPHERE(N) and just SPHERE graph the sphere as a SURFACE
9 % and do not return anything.
10 %
11 % SPHERE(AX,...) plots into AX instead of GCA.
12 %
13 % See also ELLIPSOID, CYLINDER.
14
15 % Clay M. Thompson 4-24-91, CBM 8-21-92.
16 % Copyright 1984-2002 The MathWorks, Inc.
17
18 % Parse possible Axes input
19 - narginchk(0,2);
20 - [cax,args,nargs] = axescheck(varargin{:});
21
22 - n = 20;
23 - if nargs > 0, n = args{1}; end
```



가변 길이 출력인자 - varargin, nargout

```
+38  get_matrix_size.m
1    function [s, varargin] = get_matrix_size(x)
2
3    s = size(x);
4    varargin = cell(1,nargout-1);
5
6    for k = 1:nargout-1
7        varargin{k} = s(k);
8    end
9
10   end
```

※ 반환값으로 varargin만 있으면 어떻게 될까?

* varargin = variable argument output

- 함수 호출 시 반환값 개수 가변 가능
- varargin의 값이 s 이후 반환값에 하나씩 할당
- nargout: 함수 호출 시 요청된 반환값 개수
- 소문자로 정확히 varargin이라고 써야 동작함 (다른 이름 x)
- varargin은 반환값 목록 가장 마지막에 와야 함

```
>> get_matrix_size(rand(3,4,5))
ans =
     3     4     5
>> [s, d1] = get_matrix_size(rand(3,4,5))
s =
     3     4     5
d1 =
     3
>> [s, d1, d2] = get_matrix_size(rand(3,4,5))
s =
     3     4     5
d1 =
     3
d2 =
     4
>> [~, d1, d2, d3] = get_matrix_size(rand(3,4,5))
d1 =
     3
d2 =
     4
d3 =
     5
```

get_random_names2

```
function varargout = get_random_names2

lastnames = get_lastnames();
idx = randsample(length(lastnames),nargout);
lastnames = lastnames(idx);

firstnames = get_firstnames();
idx = randsample(length(firstnames),nargout);
firstnames = firstnames(idx);

varargout = num2cell(firstnames + " " + lastnames);

end

function names = get_lastnames

names = [    "Smith" ;    "Johnson";    "Scott";    "Torres";
           "Nguyen";    "Hill";    "Flores";    "Green"];

end

function names = get_firstnames

names = [    "James";    "David";    "Christopher";    "George";
           "Ronald";    "William";    "Thomas";    "Donald"];

end
```

```
>> [name1, name2, name3] = get_random_names2
name1 =
    "Kevin Ramirez"
name2 =
    "Brian King"
name3 =
    "Jason Moore"
```

- num2cell이 쓰였음에 주목하자.

Q&A

