

Python® 사용자를 위한 MATLAB®

MATLAB 언어는 주로 수학 집약적인 계산과학을 위해 설계되었습니다. MATLAB은 반복 분석에 맞게 조정된 데스크탑 환경과 행렬 및 배열 수학을 직접적으로 표현하는 프로그래밍 언어를 결합합니다. 이런 철학과 API 설계를 이해하면 MATLAB을 배우는 데 도움이 될 수 있습니다.

» 일반 동작

Python 구문	MATLAB 구문	목적	MATLAB 예시
#	%	주석	%hello
print	;(세미콜론)으로 종료하지 않음	출력값 출력	x
/	...	다음 라인에서 계속	x = 1+...2;
os	!	운영 체제 명령	! echo hi
+ - * /	+ - * /	수학 연산자	x = 1+2
**	^	지수	x = y^2
* / **	.* ./ .^	요소별 연산자	x = [1 2].* [3 4]
not, and, or	~ &	NOT, AND, OR 논리 연산자	if x<2 & x>2
del	clear	메모리에서 변수 지우기	clear x y
clear	clc	명령 창 지우기	clc

» 참조

MATLAB 구문	목적	예시
()	인덱싱 (쓰기 시 복사)	x(1,1)
[]	배열 만들기	x = [1 2 3]
	배열 결합	z = [x ; y]
{}	셀 배열 만들기	x = {42; "hello world"}
	컨테이너에서 내용 추출	x{1,1}
.	클래스 속성 또는 메서드에 액세스	obj.Data
	참조 테이블 또는 구조체 필드	t.FieldName

- 첫 요소의 인덱스는 1입니다.
- 인덱싱은 구간의 왼쪽 및 오른쪽 끝 값을 포함합니다.
- 인덱싱 옵션으로는 N차원 인덱싱(행, 열), 선형 인덱싱(요소 번호), 논리형 인덱싱(조건문)이 있습니다.

» 함수

함수 만들기	함수 파일 내에 함수를 선언할 수 있습니다. 입력 인수는 괄호로 묶습니다.	function z = foo(x,y) ... end
	여러 개의 출력값은 대괄호로 묶습니다.	function [a,b] = foo(x,y) ... end
입력 인수 및 이름-값 쌍으로 함수 호출		y = foo(x,y,"Name",Value)

» 데이터형

유사한 데이터형:

Python	MATLAB
float	double, single
complex	complex single, complex double
int	(u)int8, (u)int16, (u)int32, (u)int64
float(nan)	NaN
float(inf)	inf
str	str, char
bool	logical
dict	struct
list, tuple	cell
pandas.dataframe	table

MATLAB은 기본적으로 모든 숫자형 값을 배정밀도 부동소수점 숫자로 저장합니다. Python의 경우 정수로 저장되는 숫자도 있고, 부동소수점으로 저장되는 숫자도 있습니다. MATLAB에서는 $x=4$ 이고 $y=4.0$ 인 경우 x 는 항상 y 와 동일합니다.

» 제어 흐름

명령문	예시
for	<pre>for i = 1:10 ... end</pre>
if	<pre>if x<3 ... elseif x == 2 else ... end</pre>
while	<pre>while x<3 ... end</pre>
switch-case	<pre>switch switch_arg ... case case_arg ... end</pre>
try-catch	<pre>try ... catch ... end</pre>

» 객체

클래스 정의

```
classdef MyClass
    properties
        MyProp
    end
    methods
        function obj = MyClass(val)
            end
        function y = MyMethod(obj,x)
            end
        end
    end
end
```

클래스 사용

- 클래스 정의를 클래스와 동일한 이름으로 저장 `MyClass.m`
 - 클래스의 객체 만들기 `a = MyClass`
 - 속성에 액세스 `a.MyProp`
 - 메서드를 호출하여 연산 수행 `b = MyMethod(a,val)`
 - 참조 방식으로 전달하기 위해 "핸들" 클래스 만들기 `classdef myclass < handle`
- ```
...
end
```