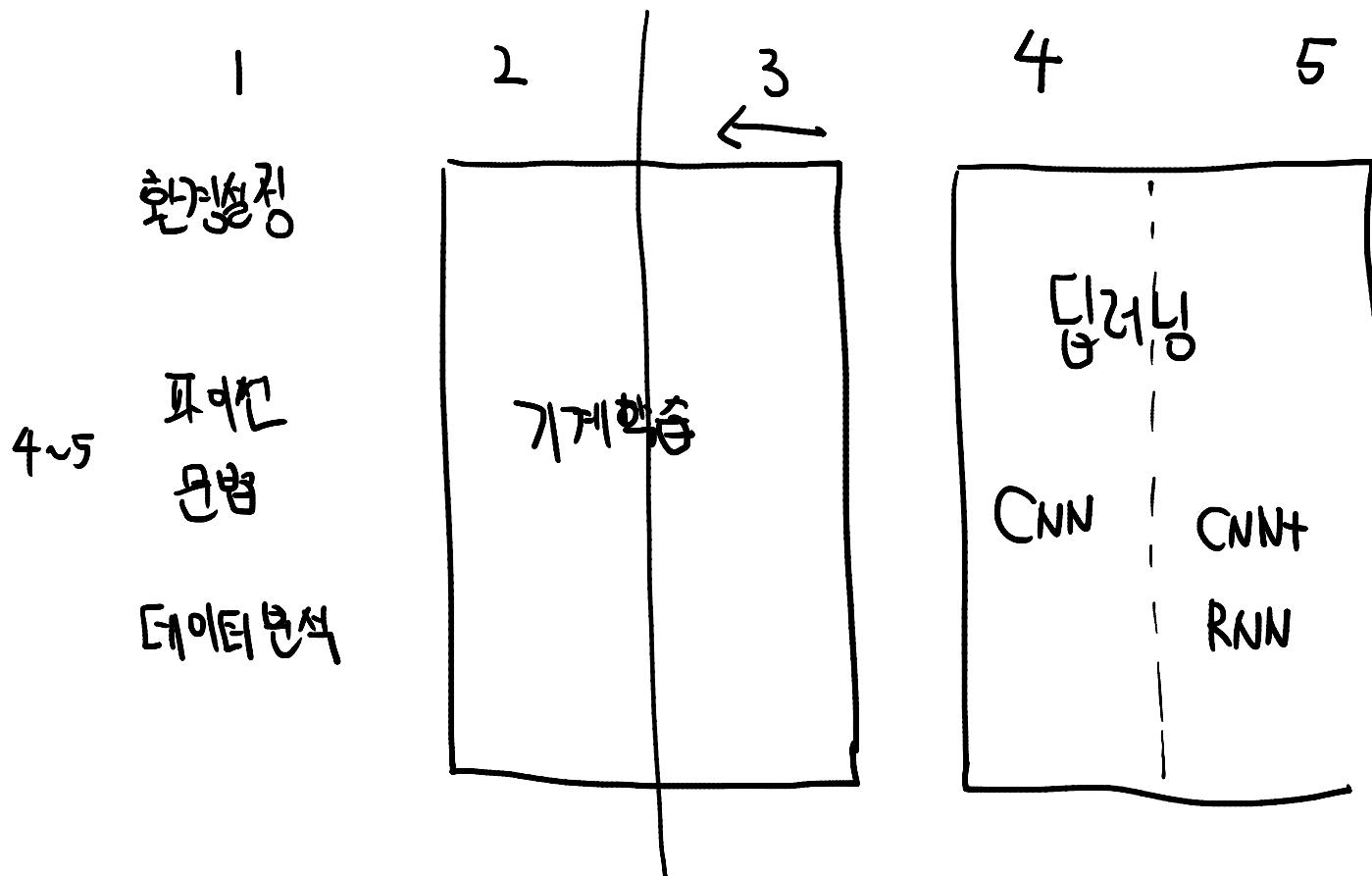
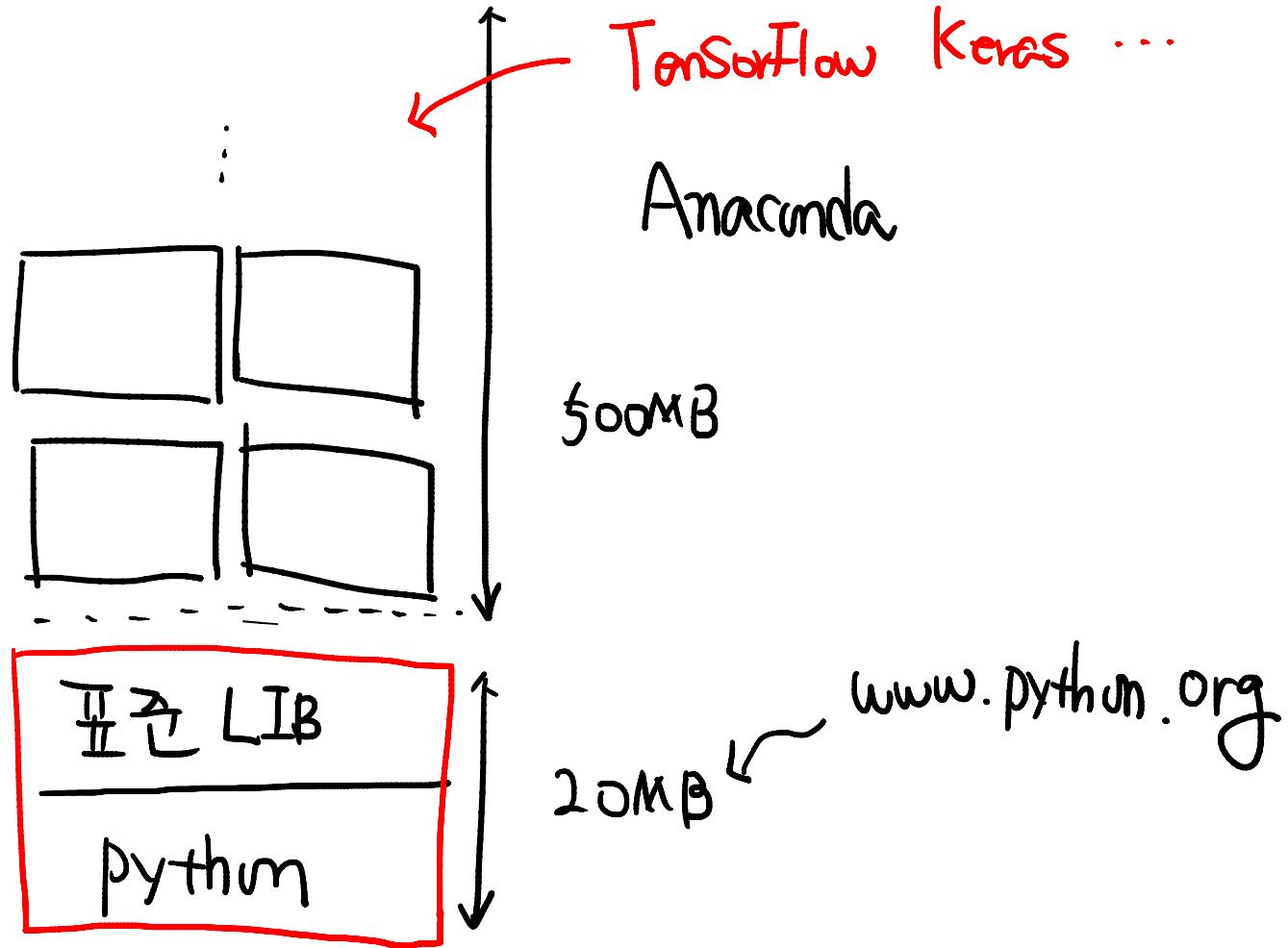


기계학습 > 딥러닝







Anaconda Cloud Documentation Blog Contact



What is Anaconda? Products Support Community About Resources

**Download**

Windows



macOS



Linux

✓  
3.X

## Anaconda 5.0.1 For Windows Installer

Python 3.6 version \*

Download

[64-Bit Graphical Installer \(515 MB\)](#) (?)[32-Bit Graphical Installer \(420 MB\)](#)

2.7

Python 2.7 version \*

Download

[64-Bit Graphical Installer \(500 MB\)](#) (?)[32-Bit Graphical Installer \(403 MB\)](#)[Behind a firewall?](#)[How to get Python 3.5 or other Python versions](#)[How to Install ANACONDA](#)

## Get Started

[Anaconda Documentation](#)[How to Use Anaconda Navigator](#)[Packages Included in Anaconda](#)



## License Agreement

Please review the license terms before installing Anaconda3 5.0.1 (64-bit).

Press Page Down to see the rest of the agreement.

=====

Anaconda End User License Agreement

=====

Copyright 2015, Anaconda, Inc.

무료 오픈 공개의무x

All rights reserved under the 3-clause BSD license:

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 5.0.1 (64-bit).

Anaconda, Inc.

< Back

I Agree

Cancel



## Advanced Installation Options

Customize how Anaconda integrates with Windows

### Advanced Options

- Add Anaconda to my PATH environment variable

환경변수 PATH

- Register Anaconda as my default Python 3.6

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.6 on the system.

Anaconda, Inc.

< Back

Install

Cancel



## Anacoda Prompt

\$ where python

\$ python

>>> print('hello')



파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도구(T) 도움말(H)



Logout

Files

Running

Clusters

Select items to perform actions on them.

현재 작업 디렉토리 "상승폴더"



Upload

New ▾



<input type="checkbox"/>	<input type="checkbox"/>	Documents / pydeep	Name ↑	Last Modified ↑
		..		seconds ago
		data.zip		2 months ago
		사전확인.txt		2 months ago





"IPython Notebook"

Trusted

Python 3

File Edit View Insert Cell Kernel Widgets Help



셀(cell)

실행

In [1]: `print('새해 복 많이 받으세요!')` ↩  
`print('셀 실행은 shift+Enter')`

실행번호

새해 복 많이 받으세요!  
셀 실행은 shift+Enter ] 흐름

In [ ]:

Trusted

Python 3

File Edit View Insert Cell Kernel Widgets Help

In [2]: `print(x)`

-----  
-----  
1) **NameError** Trace  
2) back (most recent call last)  
**<ipython-input-2-81745ac23551>** in **<module>()**  
----> 1 `print(x)`

3) **NameError:** name 'x' is not defined

In [ ]:

Trusted

Python 3

File Edit View Insert Cell Kernel Widgets Help



In [1]: `print('새해 복 많이 받으세요!')`  
`print('셀 실행은 shift+Enter')`

새해 복 많이 받으세요!  
셀 실행은 shift+Enter

[1] → [3] → [4]

In [4]: `print(x)`

1

In [3]: `x = 1`

In [ ]:

Trusted

Python 3

File Edit View Insert Cell Kernel Widgets Help

In [5]: `print(x)`

셀 유형

제목1

## # 파이썬 딥러닝

파이썬 기반으로 딥러닝을 수행하는 기법을 학습합니다.

본문

### ## 진행

제목2

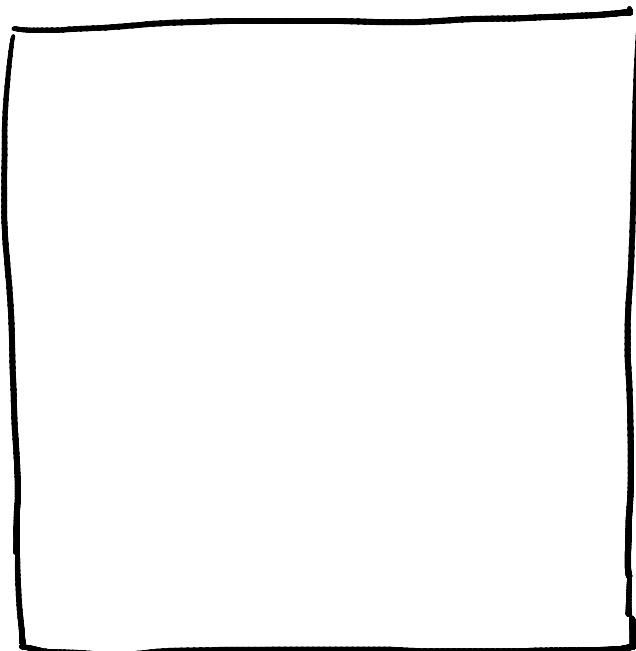
1. 주피터 노트북
1. 파이썬 필수 문법
1. 데이터 분석 기본
1. 기계학습
1. 인공 신경망
1. 딥러닝 프레임워크

차례

Shift + Enter

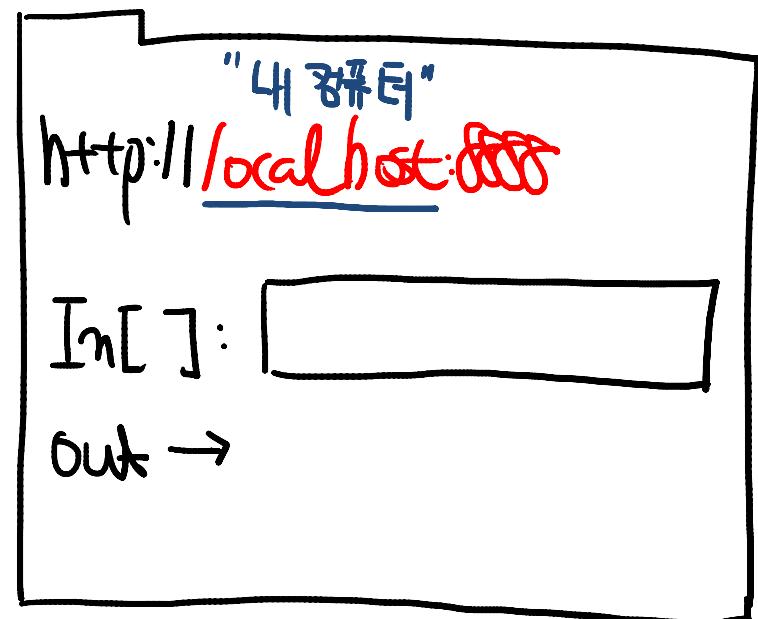
Win / Mac / Linux  $\Rightarrow$  "Web"

Jupyter Server



$\downarrow \uparrow$   
Python.exe

Jupyter Notebook



http://localhost:8888/?token=...

```
Jupyter Notebook
```

```
[I 10:33:04.649 NotebookApp] The Jupyter Notebook is running at: http://localhost:8888/?token=79ed7c750f6a28e275d6114d91da0cbc593a4165f75145a5
[I 10:33:04.649 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 10:33:04.651 NotebookApp]
```

Trusted

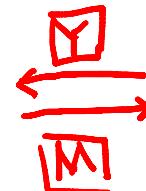
Python 3

File Edit View Insert Cell Kernel Widgets Help



A red circle highlights the copy icon (a clipboard with a plus sign).

새해 복 많이 받으세요!  
셀 실행은 shift+Enter



셀 실행 번호는 실행할 때마다 부여됩니다.

In [2]: `x = 1`

A red circle highlights the 'Above' icon (a square with an upward arrow).

A red circle highlights the 'Below' icon (a square with a downward arrow).

~~In [ ]:~~ \*In [3]: `print(x)`

1

In [ ]:

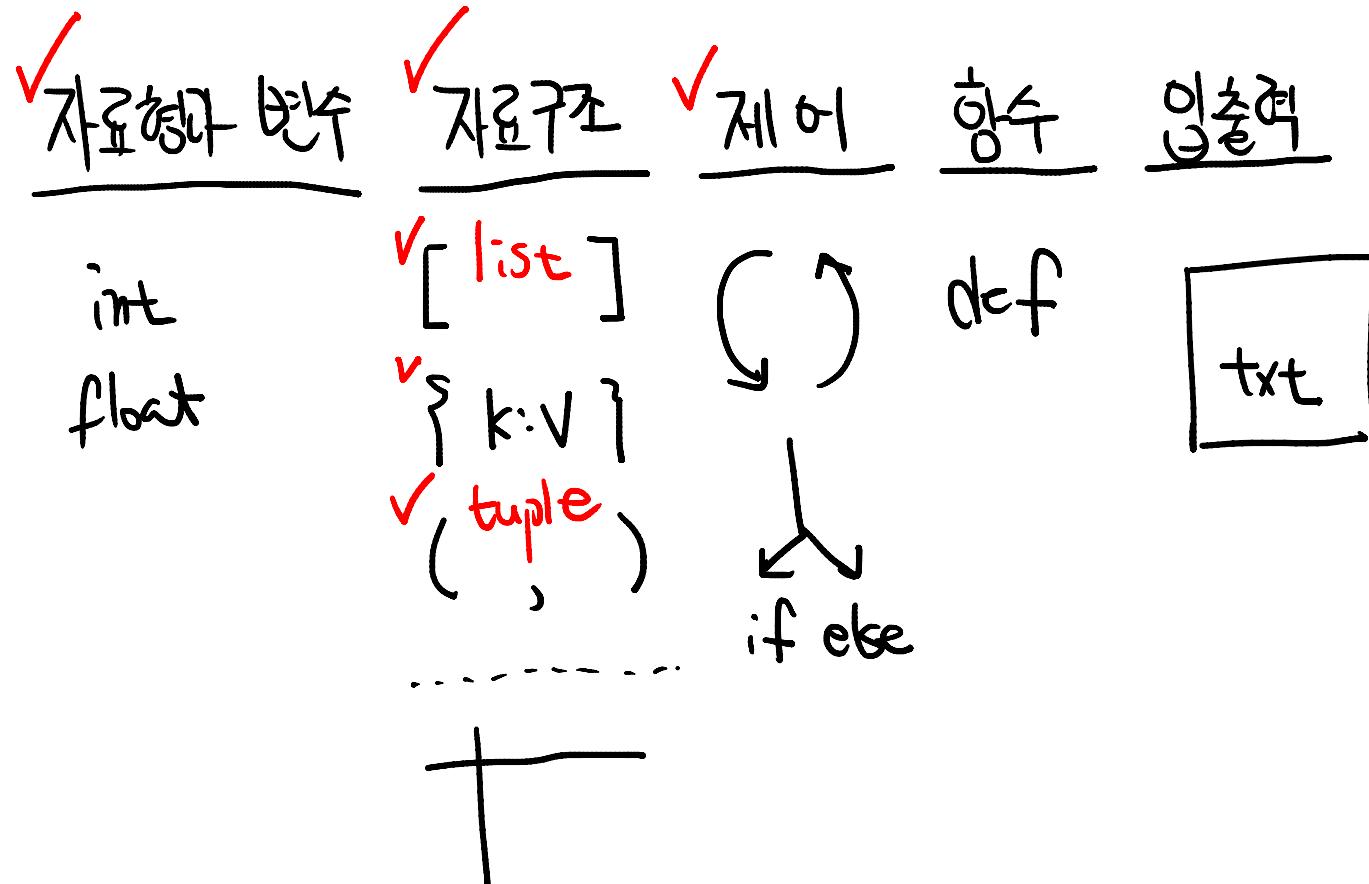
Anacoda prompt  
\$ jupyter notebook password

Enter password: —

(jupyter server restart)  
http://localhost:8888 ← 사용중인 포트  
다른 수로 바꾸자.

password:

# 파이썬



File Edit View Insert Cell Kernel Widgets Help



In [4]: `x = 1`

In [5]: `type(x)`

Out[5]: int 정수  
~~long~~

In [6]: `x = 3.14`

In [7]: `type(x)`

Out[7]: float 무정수

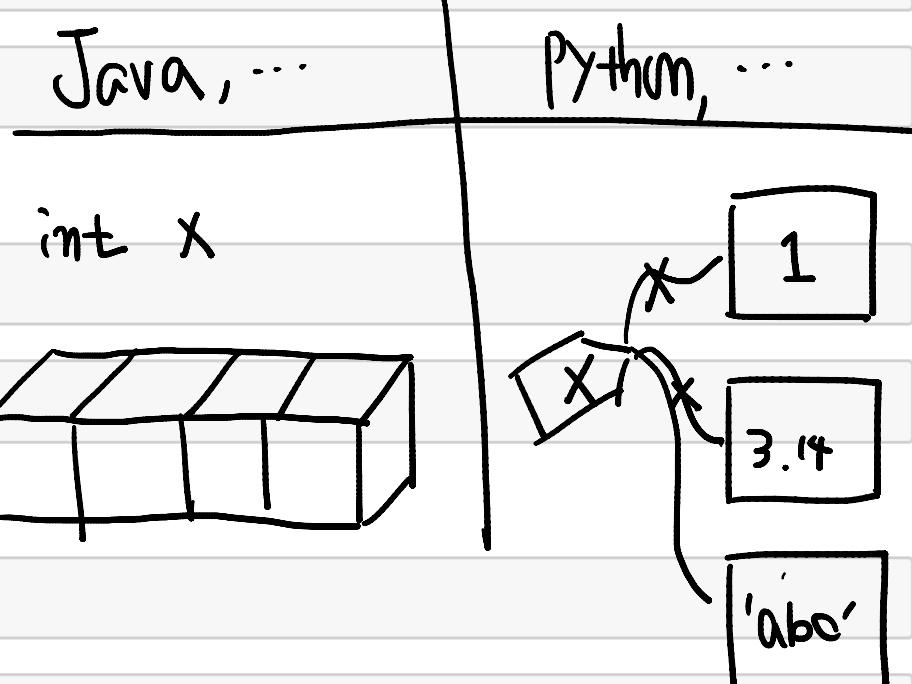
In [8]: `x = 'abc'`

In [9]: `type(x)`

Out[9]: str 문자열(String)

In [ ]:

In [ ]:



File Edit View Insert Cell Kernel Widgets Help



In [11]: 세록 = "성수's 샌드위치"

여러 줄은 원래 안 되지만 이렇게 하면 됩니다.

In [12]: text = """안녕하세요,  
반갑습니다."""

In [13]: print(text)

← 문자열 표현(사용자)

안녕하세요,  
반갑습니다.

→ 객체 표현(프로그래머)

In [14]: text



Out[14]: '안녕하세요, 반갑습니다.'

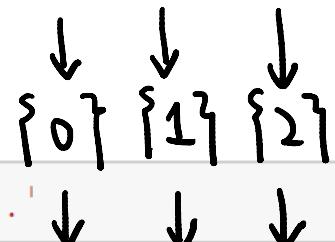
In [ ]:



## 문자열 양식

양식문자열.format(0, 1, 2, ...)

자료를 활용해 문자열을 조합할 때 권장됩니다.



In [26]: 양식 = '{0}에서는 {1} {2}.x 버전을 활용합니다.'

In [28]: 양식 "평범한 문자열"

Out[28]: '{0}에서는 {1} {2}.x 버전을 활용합니다.'

In [27]: 문구 = 양식.format(제목, 언어, 버전)  
print(문구)

자료형 변환 X

양식 채우기

파이썬 딥러닝에서는 파이썬 3.x 버전을 활용합니다.

In [ ]:

File Edit View Insert Cell Kernel Widgets Help



In [27]: 문구 = 양식.format(제목, 언어, 버전)  
print(문구)

파이썬 딥러닝에서는 파이썬 3.x 버전을 활용합니다.

문자열 양식의 형식 지정

{색인[:형식]}

In [29]: 잔고 = 1234567890

In [30]: print('{0:,}'.format(잔고))

1,234,567,890

e.g. {0:,}

.format(x<sub>1</sub>, x<sub>2</sub>, x<sub>3</sub>, ...)  
"색인" 0 1 2

In [ ]:

File Edit View Insert Cell Kernel Widgets Help



In [29]: 잔고 = 1234567890

←

.2f

In [30]: print('{0:,}'.format(잔고))

1,234,567,890

float

제작

In [34]: 양식 = '{0} | {0:f} | {0:.2f} | {0:.0f} | {0:}% | {0:.2}%'

←

In [37]: 출력 = 양식.format(1/3)  
print(출력)

0.3333333333333333 | 0.333333 | 0.33 | 0 | 33.333333% | 33.33%

.6 .2

In [ ]:

File Edit View Insert Cell Kernel Widgets Help



In [38]: 제목

0 6

Out [38]: '파이썬 딥러닝'  
-1

len( ) -1

In [39]: len(제목)

"length"

Out [39]: 7

0 > -1

In [40]: 제목[0]

0 < -1

Out [40]: '파'

In [41]: 제목[6]

Out [41]: '닝'

In [42]: 제목[-1]

Out [42]: '닝'

In [ ]:

File Edit View Insert Cell Kernel Widgets Help



Out [42]:

슬라이스

범위 선택

[0 1 2] 3 4 5 6  
파이썬 딥러닝

In [43]: 제목[0:3] "0≤색인&lt;3"

Out [43]: '파이썬'

연습

마지막 세 글자 선택

In [ ]:

File Edit View Insert Cell Kernel Widgets Help



## 연습

마지막 세 글자 선택

In [46]: 제목[4:] 4≤ 색인 ≤ 끝

Out [46]: '딥러닝'

In [47]: 제목[-3:] 3≤ 색인 ≤ 끝

Out [47]: '딥러닝'

객체의 길이가 바뀌는 경우에 역순 색인이 유용할 수 있습니다.

In [49]: 문구 = 제목 + 제목 + 제목

In [51]: 문구[4:]

Out [51]: '딥러닝파이썬 딥러닝파이썬 딥러닝'

In [52]: 문구[-3:]

Out [52]: '딥러닝'

연산자

+ - \* /

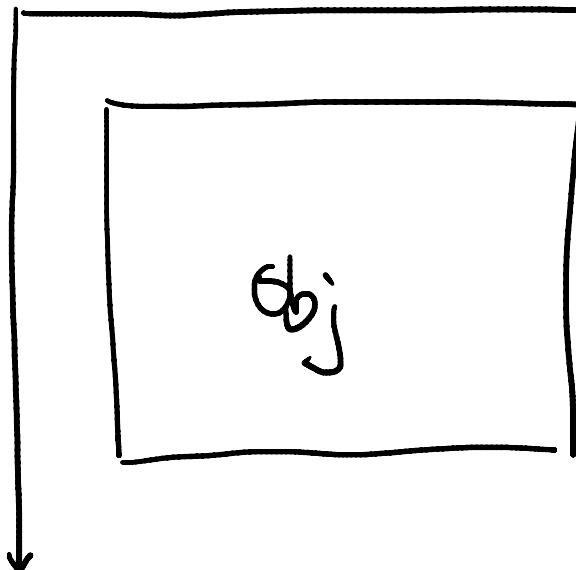
[색인]

[슬라이스]

:

obj

메소드



File Edit View Insert Cell Kernel Widgets Help



In [52]: 문구[-3:]

Out[52]: '딥러닝'

[ ] "기호"

## 자료구조

In [59]: nums = [1,2,3,4,5,6]

New List( ) X

In [60]: type(nums)

Out[60]: list

In [ ]:

File Edit View Insert Cell Kernel Widgets Help



## 자료구조

In [59]: `nums = [1,2,3,4,5,6]`

In [60]: `type(nums)`

Out[60]: `list`

기호로 객체가 생성되는 경우, 약간 혗갈릴 수 있습니다.

In [61]: `[1,2,3][-1]`

Out[61]: `3`

In [ ]:

[ ] [ ]  
"list 생성" "선택" 연산자

배열형	매핑형
[ 0 1 2 3 ...	key → Value
list	dict

File Edit View Insert Cell Kernel Widgets Help



사전

{ 색인: 값, ... }

In [86]: 연락처 = {'이름': '이성주', '전화번호': '010-1234-5678'}

In [87]: type(연락처)

Out[87]: dict

In [88]: 연락처['이름']

Out[88]: '이성주'

포함 여부는 색인을 기준으로 합니다.

In [89]: '이메일' in 연락처

Out[89]: False

값의 포함 여부는 기본적으로는 안 됩니다.

File Edit View Insert Cell Kernel Widgets Help



Markdown



In [96]: `nums.append(7)`

In [97]: `nums`

Out[97]: `[1, 2, 3, 4, 5, 6, 7]`

In [98]: `nums.pop()` “우리 기반”

Out[98]: `7`

In [99]: `nums`

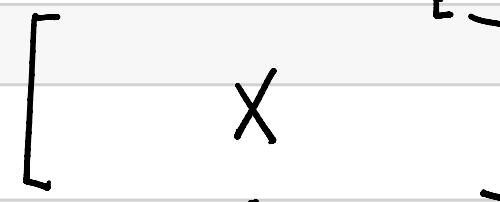
Out[99]: `[1, 2, 3, 4, 5, 6]`

In [100]: `nums.remove(4)` “값 기반”

In [101]: `nums`

Out[101]: `[1, 2, 3, 5, 6]`

append(X)



remove(X) · pop()



File Edit View Insert Cell Kernel Widgets Help



Code



풀기 (unpacking) ~ 대입방법

$x, y, z = (1.2, 2.3, 3.4)$

In [138]:  $x, y, z =$  좌표

In [139]:  $x$

Out[139]: 1.2

In [140]:  $y$

Out[140]: 2.3

In [141]:  $z$

Out[141]: 3.4

In [ ]:

File Edit View Insert Cell Kernel Widgets Help



In [142]: nums

Out[142]: [-1, 2, 3, 5, 6]



In [143]: a, b, \*나머지 = nums

In [144]: a

Out[144]: -1

In [145]: b

Out[145]: 2

In [146]: 나머지

Out[146]: [3, 5, 6]

In [ ]:

File Edit View Insert Cell Kernel Widgets Help



Code



Out[146]: [3, 5, 6]

## 제어

```
In [147]: for n in [0,1,2,3,4]:  
    → n2 = n ** 2  
    → print(n2) 제공
```

0  
1  
4  
9  
16

In [ ]:

File Edit View Insert Cell Kernel Widgets Help



Code

In [163]: `nums * 2`

Out[163]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

In [164]: `nums2 = []``for n in nums:` `nums2.append(n*2)` $f$  $[x_1, f(x_1)]$  $f$  $x_2, f(x_2)$  $f$  $x_3, f(x_3)$  $\vdots$ In [165]: `nums2`

Out[165]: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

In [166]: `nums2 = [n*2 for n in nums]`

Out[166]: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

In [ ]:



```
print(둘)
print('라면 널기')
```

```
100
200
300
400
500
라면 널기
```

"Ctrl+C"  
Kernel Interrupt

"실행 중"

주피터 노트북에서 무한 반복 해결하기

In [\*]: while True:
 pass ← {}

In [ ]:

KERNEL ✓ BUSY  
IDLE

File Edit View Insert Cell Kernel Widgets Help



## 조건

In [181]: 돈 = 8

```
if 돈 > 9:  
    메뉴 = '스테이크'  
elif 돈 > 7:  
    메뉴 = '국밥'  
elif 돈 > 5:  
    메뉴 = '우동'  
else:  
    메뉴 = '호떡'  
  
print(메뉴)
```

국밥

In [ ]:



# 함수

인자  
기본값

```
In [189]: def 주스기(재료, 휘핑크림=True):
    주스 = '{0}주스'.format(재료)
    if 휘핑크림:
        주스 = '휘핑크림을 올린 ' + 주스
    return 주스
```

```
In [190]: 주스기('딸기')
```

```
Out[190]: '휘핑크림을 올린 딸기주스'
```

```
In [ ]:
```



Code



## 고계 함수

High-Order function

?

```
In [232]: def 집계(방법, data):  
    결과 = 방법(data)  
    return 결과
```

$$\text{Sum(data)} \rightarrow \sum_{i=1}^{10} i$$

제곱합 (data) →  $\sum_{i=1}^{10} i^2$

"호출 가능한 것"

```
In [233]: def 제곱합(data):  
    return sum(n**2 for n in data)
```

```
In [234]: data = list(range(1, 11))
```

```
In [235]: 집계(sum, data)
```

Out[235]: 55

↑ 빙도인 함수

```
In [236]: 집계(제곱합, data)
```

Out[236]: 385

In [ ]:

File Edit View Insert Cell Kernel Widgets Help



In [235]: `집계(SUM, data)`

Out[235]: 55

In [236]: `집계(제곱합, data)`

Out[236]: 385

유의할 점

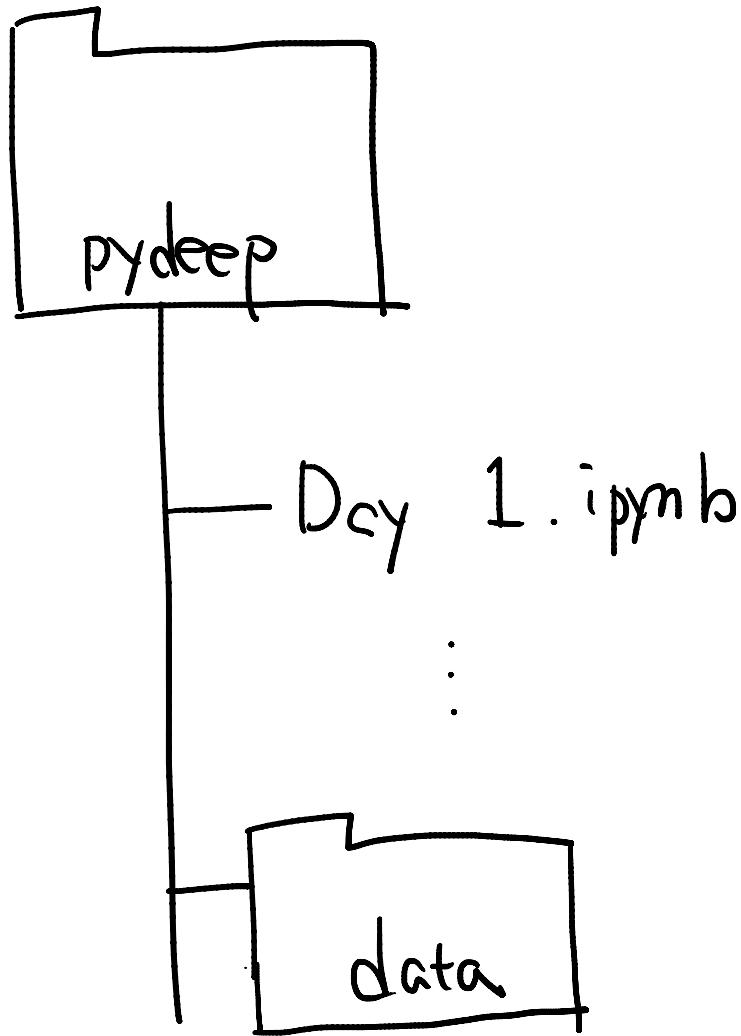
변수공간  
Sum

In [238]: `sum = 0  
for n in range(1, 11):  
 sum += n  
print(sum)`

55

빌트인 공간  
print, type, SUM, ...

In [ ]:



가능

"바닐라"

[ , ]

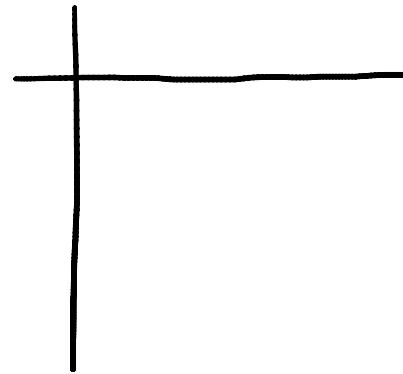
{ k:v, }

( , )

가능

pandas

제어



DataFrame

제어  $\propto$  줄수  $\propto$  작업시간