

Python Official Tutorial

- <https://docs.python.org/ko/3.8/tutorial/index.html>

Python » Korean 3.8.3 설명서 »

이전 항목
Changelog

다음 항목
1. 입맛 돋우기

현재 문서
버그 보고하기
소스 보기

파이썬 자습서

파이썬은 배우기 쉽고, 강력한 프로그래밍 언어입니다. 효율적인 자료 구조들과 객체 지향 프로그래밍에 대해 간단하고도 효과적인 접근법을 제공합니다. 우아한 문법과 동적 타이핑(typing)은, 인터프리터 적인 특징들과 더불어, 대부분 플랫폼과 다양한 문제 영역에서 스크립트 작성과 빠른 응용 프로그램 개발에 이상적인 환경을 제공합니다.

파이썬 인터프리터와 풍부한 표준 라이브러리는 소스나 바이너리 형태로 파이썬 웹 사이트, <https://www.python.org/>, 에서 무료로 제공되고, 자유롭게 배포할 수 있습니다. 같은 사이트는 제삼자들이 무료로 제공하는 확장 모듈, 프로그램, 도구, 문서들의 배포판이나 링크를 포함합니다.

파이썬 인터프리터는 C 나 C++ (또는 C에서 호출 가능한 다른 언어들)로 구현된 새 함수나 자료 구조를 쉽게 추가할 수 있습니다. 파이썬은 고객화 가능한 응용 프로그램을 위한 확장 언어로도 적합합니다.

이 학습서는 파이썬 언어와 시스템의 기본 개념과 기능들을 격식 없이 소개합니다. 파이썬 인터프리터를 직접 만져볼 수 있도록 돕지만, 모든 예제가 독립적이기 때문에 오프라인에서 읽기에도 적합합니다.

표준 객체들과 모듈들에 대한 설명은 [파이썬 표준 라이브러리](#) 를 보세요. [파이썬 언어 레퍼런스](#) 는 언어에 대한 좀 더 형식적인 정의를 제공합니다. C 나 C++ 로 확장하려면 [파이썬 인터프리터 확장 및 내장](#) 와 [파이썬/C API 레퍼런스 설명서](#) 를 읽으세요. 파이썬을 깊이 있게 다룬 책들도 많습니다.

이 자습서는 포괄적이라고 시도하지 않습니다. 모든 기능을 다루지는 않는데, 심지어 자주 사용되는 기능조차도 그렇습니다. 대신에, 파이썬의 가장 주목할만한 기능들을 소개하고, 언어의 맛과 스타일에 대한 전체적인 인상을 제공합니다. 이 학습서를 읽은 후에는 파이썬 모듈과 프로그램을 작성할 수 있고, [파이썬 표준 라이브러리](#) 에 기술된 다양한 파이썬 라이브러리 모듈들에 대해 학습할 수 있는 준비가 될 것입니다.

용어집 또한 흠여볼 만한 가치가 있습니다.

- 1. 입맛 돋우기
- 2. 파이썬 인터프리터 사용하기
 - 2.1. 인터프리터 실행하기
 - 2.1.1. 인자 전달
 - 2.1.2. 대화형 모드
 - 2.2. 인터프리터와 환경
 - 2.2.1. 소스 코드 인코딩

Python containers - Lists

- <https://docs.python.org/ko/3/tutorial/datastructures.html>
- Lists – 배열 형태로 데이터를 표현
- `dummy_list = []` 형태로 선언

```
xs = [3, 1, 2]      # Create a list
print(xs, xs[2])    # Prints "[3, 1, 2] 2"
print(xs[-1])       # Negative indices count from the end of the list; prints "2"
xs[2] = 'foo'       # Lists can contain elements of different types
print(xs)           # Prints "[3, 1, 'foo']"
xs.append('bar')     # Add a new element to the end of the list
print(xs)           # Prints "[3, 1, 'foo', 'bar']"
x = xs.pop()        # Remove and return the last element of the list
print(x, xs)        # Prints "bar [3, 1, 'foo']"
```

Lists Slicing

- Slicing을 통해 손쉽게 List의 요소들을 접근할 수 있다.

```
nums = list(range(5))      # range is a built-in function that creates a list of integers
print(nums)               # Prints "[0, 1, 2, 3, 4]"
print(nums[2:4])          # Get a slice from index 2 to 4 (exclusive); prints "[2, 3]"
print(nums[2:])            # Get a slice from index 2 to the end; prints "[2, 3, 4]"
print(nums[:2])            # Get a slice from the start to index 2 (exclusive); prints "[0, 1]"
print(nums[:])             # Get a slice of the whole list; prints "[0, 1, 2, 3, 4]"
print(nums[:-1])          # Slice indices can be negative; prints "[0, 1, 2, 3]"
nums[2:4] = [8, 9]        # Assign a new sublist to a slice
print(nums)               # Prints "[0, 1, 8, 9, 4]"
```

Python containers - Dictionaries

- <https://docs.python.org/ko/3.8/tutorial/datastructures.html#dictionaries>
- Dictionary – Key : Value 형태로 데이터 표현
- dummy_dictionary = {} 형태로 선언

```
d = {'cat': 'cute', 'dog': 'furry'} # Create a new dictionary with some data
print(d['cat']) # Get an entry from a dictionary; prints "cute"
print('cat' in d) # Check if a dictionary has a given key; prints "True"
d['fish'] = 'wet' # Set an entry in a dictionary
print(d['fish']) # Prints "wet"
# print(d['monkey']) # KeyError: 'monkey' not a key of d
print(d.get('monkey', 'N/A')) # Get an element with a default; prints "N/A"
print(d.get('fish', 'N/A')) # Get an element with a default; prints "wet"
del d['fish'] # Remove an element from a dictionary
print(d.get('fish', 'N/A')) # "fish" is no longer a key; prints "N/A"
```

Python containers - Tuples

- <https://docs.python.org/ko/3/tutorial/datastructures.html#tuples-and-sequences>
- Tuple – Lists와 비슷하지만 불변의 값을 가짐
- dummy_tuples = () 형태로 선언

```
d = {(x, x + 1): x for x in range(10)} # Create a dictionary with tuple keys
t = (5, 6) # Create a tuple
print(type(t)) # Prints "<class 'tuple'>"
print(d[t]) # Prints "5"
print(d[(1, 2)]) # Prints "1"
```

Python Function

- <https://docs.python.org/ko/3/tutorial/controlflow.html#defining-functions>
- Python에서 함수 작성법
- def foo(): 형태로 작성

```
def sign(x):  
    if x > 0:  
        return 'positive'  
    elif x < 0:  
        return 'negative'  
    else:  
        return 'zero'  
  
for x in [-1, 0, 1]:  
    print(sign(x))  
# Prints "negative", "zero", "positive"
```

Python Function

- 함수의 argument는 아래와 같은 형태로 지정

```
def hello(name, loud=False):  
    if loud:  
        print('HELLO, %s!' % name.upper())  
    else:  
        print('Hello, %s' % name)  
  
hello('Bob') # Prints "Hello, Bob"  
hello('Fred', loud=True) # Prints "HELLO, FRED!"
```

Python Class

- <https://docs.python.org/ko/3/tutorial/classes.html>
- Python에서 class 작성법
- class foo(): 형태로 작성

```
class Greeter(object):  
  
    # Constructor  
    def __init__(self, name):  
        self.name = name # Create an instance variable  
  
    # Instance method  
    def greet(self, loud=False):  
        if loud:  
            print('HELLO, %s!' % self.name.upper())  
        else:  
            print('Hello, %s' % self.name)  
  
g = Greeter('Fred') # Construct an instance of the Greeter class  
g.greet()           # Call an instance method; prints "Hello, Fred"  
g.greet(loud=True)  # Call an instance method; prints "HELLO, FRED!"
```


Python & Numpy Basic Tutorial

- <https://cs231n.github.io/python-numpy-tutorial/>
- <https://colab.research.google.com/github/cs231n/cs231n.github.io/blob/master/python-colab.ipynb#scrollTo=U4Jl8K0tL9e4>

CS231n Convolutional Neural Networks for Visual Recognition

Python Numpy Tutorial (with Jupyter and Colab)

 Open in Colab

This tutorial was originally contributed by [Justin Johnson](#).

We will use the Python programming language for all assignments in this course. Python is a great general-purpose programming language on its own, but with the help of a few popular libraries (numpy, scipy, matplotlib) it becomes a powerful environment for scientific computing.

We expect that many of you will have some experience with Python and numpy; for the rest of you, this section will serve as a quick crash course on both the Python programming language and its use for scientific computing. We'll also introduce notebooks, which are a very convenient way of tinkering with Python code. Some of you may have previous knowledge in a different language, in which case we also recommend referencing: [NumPy for Matlab users](#), [Python for R users](#), and/or [Python for SAS users](#).

Table of Contents


- [Jupyter and Colab Notebooks](#)
- [Python](#)
 - [Python versions](#)
 - [Basic data types](#)

Numpy

- <https://numpy.org/>
- <https://numpy.org/doc/stable/reference/>

[Install](#) [Documentation](#) [Learn](#) [Community](#) [About Us](#) [Contribute](#)

NumPy



The fundamental package for scientific computing with Python

GET STARTED

NumPy v1.18.0 A new C-API for numpy.random - Basic infrastructure for linking with 64-bit BLAS and LAPACK

POWERFUL N-DIMENSIONAL ARRAYS

Fast and versatile, the NumPy vectorization, indexing, and broadcasting concepts are the de-facto standards of array computing today.

NUMERICAL COMPUTING TOOLS

NumPy offers comprehensive mathematical functions, random number generators, linear algebra routines, Fourier transforms, and more.

INTEROPERABLE

NumPy supports a wide range of hardware and computing platforms, and plays well with distributed, GPU, and sparse array libraries.

Numpy Example

- Numpy 배열 초기화 방법

```
import numpy as np

a = np.array([1, 2, 3])    # Create a rank 1 array
print(type(a))            # Prints "<class 'numpy.ndarray'>"
print(a.shape)            # Prints "(3,)"
print(a[0], a[1], a[2])   # Prints "1 2 3"
a[0] = 5                  # Change an element of the array
print(a)                  # Prints "[5, 2, 3]"

b = np.array([[1,2,3],[4,5,6]]) # Create a rank 2 array
print(b.shape)            # Prints "(2, 3)"
print(b[0, 0], b[0, 1], b[1, 0]) # Prints "1 2 4"
```

Matplotlib

- <https://matplotlib.org/>



The screenshot shows the Matplotlib website homepage. At the top, the Matplotlib logo is displayed with the text "Version 3.2.2" below it. A navigation bar contains links for "Installation", "Documentation", "Examples", "Tutorials", and "Contributing". A search bar is located on the right side of the navigation bar. Below the navigation bar, a breadcrumb trail reads "home | contents » Matplotlib: Python plotting". On the right side of the page, there are links for "modules" and "index". The main content area features the heading "Matplotlib: Visualization with Python" followed by a paragraph: "Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python." Below this text are four small images: a line plot with multiple peaks, a histogram with a normal distribution curve, a 2D heatmap, and a 3D surface plot. A quote states: "Matplotlib makes easy things easy and hard things possible." The page is divided into three columns: "Create", "Customize", and "Extend". The "Create" column lists: "Develop publication quality plots with just a few lines of code" and "Use interactive figures that can zoom, pan, update...". The "Customize" column lists: "Take full control of line styles, font properties, axes properties..." and "Export and embed to a number of file formats and interactive environments". The "Extend" column lists: "Explore tailored functionality provided by third party packages" and "Learn more about Matplotlib through the many external learning resources". On the right side, there is a box containing release information: "Latest release candidate 3.3.0rc1: docs | changelog", "Latest release 3.2.2: docs | changelog", "Last release for Python 2 2.2.5: docs | changelog", and "Development version docs". A button labeled "Support Matplotlib" is located at the bottom right of the page.

matplotlib
Version 3.2.2

Installation Documentation Examples Tutorials Contributing

Search

home | contents » Matplotlib: Python plotting

modules | index

Matplotlib: Visualization with Python

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.



Matplotlib makes easy things easy and hard things possible.

Create

- Develop **publication quality plots** with just a few lines of code
- Use **interactive figures** that can zoom, pan, update...

Customize

- **Take full control** of line styles, font properties, axes properties...
- **Export and embed** to a number of file formats and interactive environments

Extend

- Explore tailored functionality provided by **third party packages**
- Learn more about Matplotlib through the many **external learning resources**

Latest release candidate
3.3.0rc1: [docs](#) | [changelog](#)

Latest release
3.2.2: [docs](#) | [changelog](#)

Last release for Python 2
2.2.5: [docs](#) | [changelog](#)

Development version
[docs](#)

Support Matplotlib

matplotlib – imshow, show

```
import numpy as np
from scipy.misc import imread, imresize
import matplotlib.pyplot as plt

img = imread('assets/cat.jpg')
img_tinted = img * [1, 0.95, 0.9]

# Show the original image
plt.subplot(1, 2, 1)
plt.imshow(img)


# Show the tinted image
plt.subplot(1, 2, 2)

# A slight gotcha with imshow is that it might give strange results
# if presented with data that is not uint8. To work around this, we
# explicitly cast the image to uint8 before displaying it.
plt.imshow(np.uint8(img_tinted))
plt.show()
```



Pandas

- <https://pandas.pydata.org/>

 pandas

About us ▾ Getting started Documentation Community ▾ Contribute

pandas

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

[Install pandas now!](#)

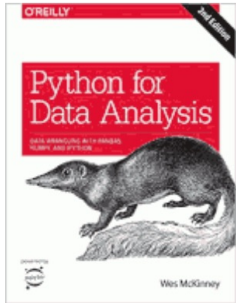
Latest version: 1.0.5

- What's new in 1.0.5
- Release date:
Jun 17, 2020
- Documentation (web)
- Documentation (pdf)
- Download source code

Follow us

[Follow @pandas_dev](#)

Get the book



Getting started

- Install pandas
- Getting started

Documentation

- User guide
- API reference
- Contributing to pandas
- Release notes

Community

- About pandas
- Ask a question
- Ecosystem

With the support of:

Pandas Basic methods – read_csv

- https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html
- read_csv()

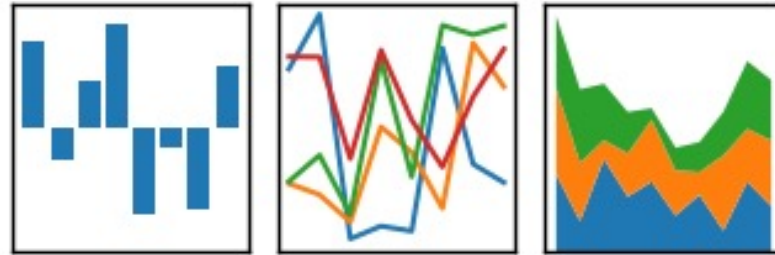
```
> > pd.read_csv('data.csv')
```

Pandas Tutorial

- <https://github.com/adeshpande3/Pandas-Tutorial/blob/master/Pandas%20Tutorial.ipynb>

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



Jupyter notebook

- <https://jupyter.org/>
- <https://nbviewer.jupyter.org/>




[Install](#) [About Us](#) [Community](#) [Documentation](#) [NBViewer](#) [JupyterHub](#) [Widgets](#) [Blog](#)



Scikit-learn

- <https://scikit-learn.org/stable/>

 [Install](#) [User Guide](#) [API](#) [Examples](#) [More ▾](#)

scikit-learn

Machine Learning in Python

[Getting Started](#) [Release Highlights for 0.23](#) [GitHub](#)

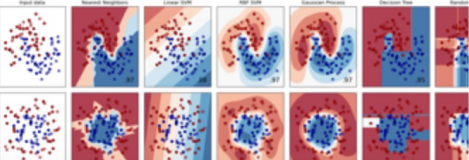
- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.

Algorithms: SVM, nearest neighbors, random forest, and more...

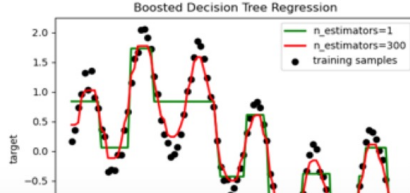


Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, nearest neighbors, random forest, and more...




Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, and more...



Scikit-learn Basic methods – train_test_split

- sklearn.model_selection.train_test_split
- https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

Examples

```
>>> import numpy as np
>>> from sklearn.model_selection import train_test_split
>>> X, y = np.arange(10).reshape((5, 2)), range(5)
>>> X
array([[0, 1],
       [2, 3],
       [4, 5],
       [6, 7],
       [8, 9]])
>>> list(y)
[0, 1, 2, 3, 4]
```

```
>>> X_train, X_test, y_train, y_test = train_test_split(
...     X, y, test_size=0.33, random_state=42)
...
>>> X_train
array([[4, 5],
       [0, 1],
       [6, 7]])
>>> y_train
[2, 0, 3]
>>> X_test
array([[2, 3],
       [8, 9]])
>>> y_test
[1, 4]
```

Thank you!
