Chapter 1. 헬로 파이썬

1.1 파이썬 인터프리터

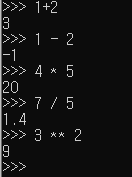
Python은 배우기 쉽고, 수치 계산과 통계 처리를 다루는 라이브러리(numpy, scipy)등과 같이 사용되며 기계학습과 데이터 과학 분야에서 뛰어난 성능을 보이고 있는 프로그래밍 언어입니다. 배우고 사용해본 적이 있지만 정리하고 넘어가는 것도 좋다고 생각해서 ch01부터 정리해보려 합니다. 설치하는 부분은 생략하고 간단한 연산 예제부터 시작해보기로 했습니다. Python은 3.x 버전을 사용하고 아나콘다를 통해 설치했습니다.





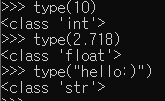
설치된 파이썬 버전과 인터프리터를 실행시킨 화면입니다. 인터프리터를 실행하면 간단하게 코드를 실행할 수 있습니다.

1.1.1 산술연산

 Python에서 사용하는 산술 연산들입니다.

1.1.2 자료형

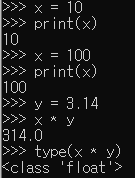
프로그래밍 언어에는 자료형(data type)이라는 것이 있는데, Python은 다른 언어와 다르게 자료형을 선언하지 않아도 알아서 자료형을 정해줍니다. 이런 언어를 동적 언어라고 합니다. 다른 언어를 사용하다 Python을 사용하면 굉장히 편하게 느껴지는 특징입니다. 자료형은 type() 함수로 출력해볼 수 있습니다.



출력을 보면, 10은 int(정수), 2.718은 float(실수), “hello😊”는 string(문자열)형임을 알 수 있습니다.

1.1.3 변수

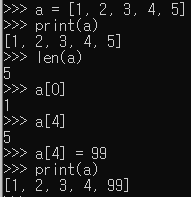
a, b, …, x, y 같은 알파벳을 사용하여 변수(variable)를 정의할 수 있습니다. 변수를 사용하여 계산, 값을 대입할 수 있습니다.



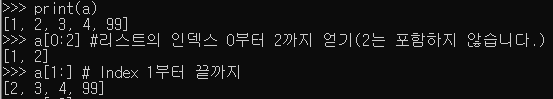
Python은 동적언어이기 때문에, x 와 y에 대해 자료형을 선언해주지 않아도 자동으로 x(int)와 y(float)을 곱한 결과에 대한 자료형이 float으로 출력되는 것을 확인할 수 있습니다.

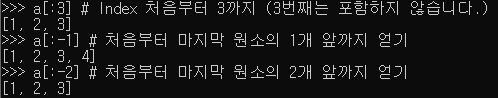
1.1.4 리스트

리스트(List)는 여러 데이터를 인덱스에 따라 저장해두는 자료형입니다.



리스트의 원소에 접근할 때는 a[0], a[4]처럼 합니다. []안의 수를 인덱스(Index)라 하고, 0부터 시작합니다. 위의 예의 경우에는 a[0]~a[4]까지 있는 길이가 5인 리스트 입니다. Python의 리스트에는 슬라이싱(Slicing)이라는 기능이 있습니다. 원하는 범위를 지정해 부분적인 리스트를 얻을 수 있습니다.





a[:3]같은 슬라이싱을 사용할 때는 항상 index가 3인 원소는 포함하지 않는 것을 주의해야 합니다. 항상 어디까지 가져오는지 까먹기도 합니다.

1.1.5 딕셔너리

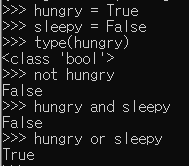
리스트가 index 번호 순으로 값을 저장한다면, 딕셔너리(Dictionary)는 키(Key)와 값(Value)을 한 쌍으로 저장합니다.





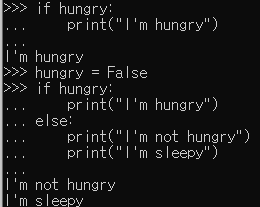
1.1.6 bool

불 또는 불리언(Bool or Boolean)은 참(True)와 거짓(False) 둘 중 하나의 값을 취하는 자료형입니다. bool에는 and, or, not 같은 논리 연산을 할 수 있습니다.



1.1.7 if 문

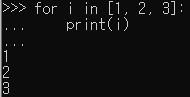
If/else는 조건에 따라 실행되느냐 실행되지 않는가를 정할 수 있습니다.



Python에서 공백문자는 중요한 의미를 지닌다고 합니다. If hungry: 다음 줄은 앞 쪽에 4개의 공백 문자가 있는데, 이 들여쓰기는 지난 조건(if hungry:)가 충족될 때 실행되는 코드를 표현합니다. 공백 대신 탭(Tab)을 사용해도 되지만 공백 문자 쪽을 권장하고, 한단계에 공백문자 4개가 일반적이라고 합니다.

1.1.8 for 문

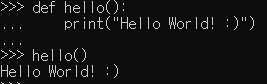
반복(루프(Loop)) 처리에는 for 문을 사용합니다.



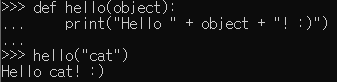
위 반복문은 [1, 2, 3] list에 i가 차례로 접근해 하나씩 출력하는 예입니다.

1.1.9 함수

특정 기능을 수행하는 일련의 명령들을 묶어 하나의 함수(Function)으로 정의할 수 있습니다.



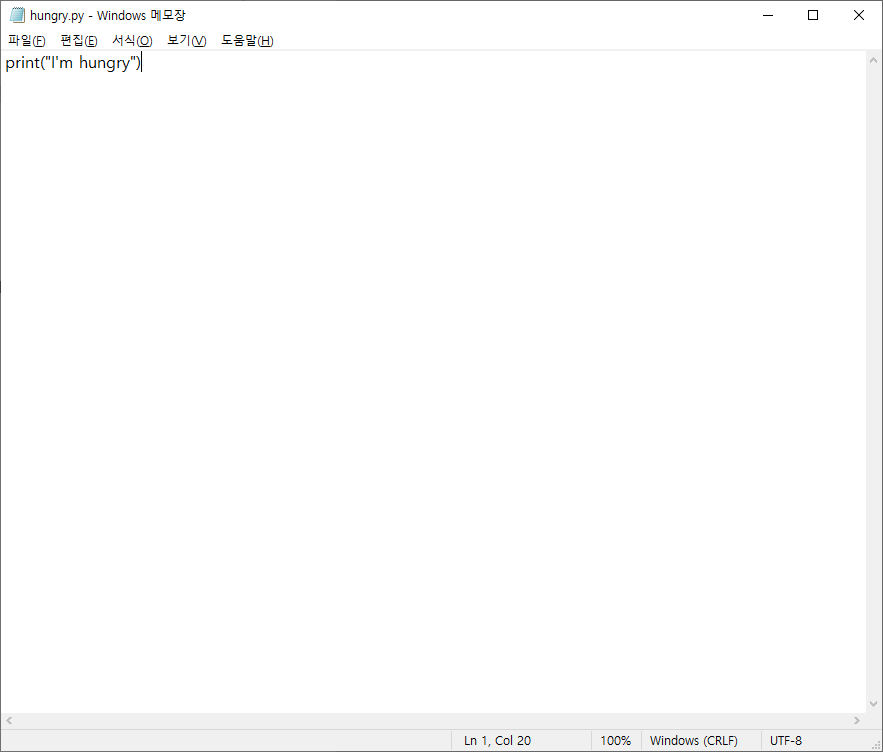
함수는 인수를 받을 수 있고, + 연산자를 이용하여 문자열을 이어 붙일 수 있습니다.



1.2 파이썬 스크립트 파일

지금까지는 파이썬 인터프리터를 활용한 예입니다. 인터프리터는 대화형으로 파이썬 코드를 실행해보며 간단한 실험을 하기에 좋습니다. 긴 작업을 수행하기에는 불편하기 때문에 이럴 때는 파이썬 프로그램을 파일로 저장하고 실행하는 방법이 있습니다. (파이썬 스크립트 파일)

1.2.1 파일로 저장하기



텍스트 편집기를 열어 hungry.py 라는 파일을 작성합니다. 그 다음에 파일을 저장한 디렉토리로 이동하여 파이썬 파일을 실행시켜봅시다.

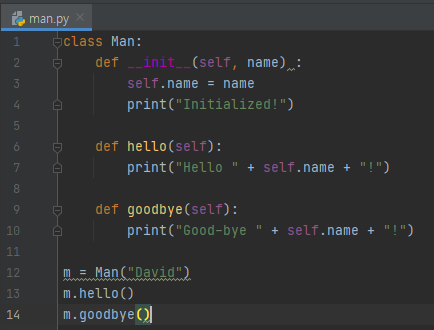


저는 바탕화면에 python example 이라는 폴더를 만들어 그곳에 파일을 저장했습니다.

(~\Desktop\python example)

1.2.2 클래스

클래스(Class)는 개발자가 독자적인 자료형을 만든다고 생각할 수 있습니다. Python에서는 class라는 키워드를 사용하여 클래스를 정의합니다. 이 부분부터는 pycharm이라는 Python IDE tool을 사용하여 코드를 작성했습니다.



클래스 정의에는 \_\_init\_\_라는 특별한 메서드가 있습니다. 이 메서드는 클래스를 초기화하는 방법을 정의합니다. 이를 생성자(Constructor)라고도 하며, 클래스의 인스턴스가 만들어질 때 한번만 불립니다. 그리고 첫번째 인수로 자신(자신의 인스턴스)을 나타내는 self를 명시적으로 쓰는 것이 특징입니다.

위의 예제 코드는 Man이라는 class를 정의했습니다. 그리고 Man class에서 m이라는 객체를 생성합니다. Man의 생성자는 name(“David”)라는 인수를 받고, 그 인수로 self.name을 초기화합니다. self.name 같은 변수를 인스턴스 변수라 합니다.

1.3 넘파이

딥러닝을 구현하다 보면 배열이나 행렬 계산이 많이 등장합니다. 넘파이의 배열 클래스인 numpy.array에는 편리한 메서드가 많이 준비되어 있어, 딥러닝을 구현할 때 이용합니다.

1.3.1 넘파이 가져오기

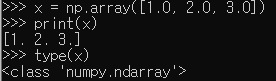
넘파이(numpy)는 외부 라이브러리이기 때문에, python에 기본 내장되어 있지 않습니다. 그래서 넘파이 라이브러리를 사용할 수 있도록 가져와야(import)합니다.



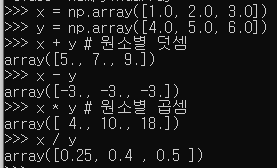
Python에서는 다른 라이브러리를 읽기 위해 import 문을 사용합니다. ‘Numpy 라이브러리를 np라는 이름으로 가져와라’라는 의미를 가지고 있습니다. 이제부터 numpy가 제공하는 메서드를 np를 통해 참조할 수 있습니다.

1.3.2 넘파이 배열 생성하기

np.array() 메서드를 이용하여 넘파이 배열을 만듭니다. np.array()는 파이썬의 리스트를 인수로 받아 numpy의 특수한 형태의 배열(numpy.ndarray)을 반환합니다.



1.3.3 넘파이의 산술 연산

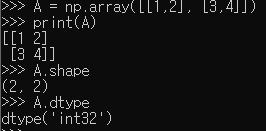


여기서 주의해야할 점은 원소 수가 같지 않으면 산술 연산은 행해지지 않는다는 것입니다. 넘파이는 원소별(“element-wise”) 뿐만 아니라 스칼라값과의 연산도 가능합니다. 이 경우 스칼라값과의 계산이 넘파이 배열의 원소별로 한 번씩 수행되며, 이를 브로드캐스트(Broadcast)라 합니다.

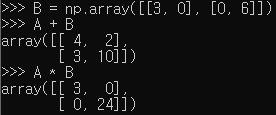


1.3.4 넘파이의 N차원 배열

넘파이는 다차원 배열도 작성할 수 있습니다.



A에 2x2 행렬을 작성했습니다. 행렬의 형상은 shape로, 원소의 자료형은 dtype으로 알 수 있습니다.



형상이 같은 행렬끼리면 산술 연산도 대응하는 원소별로 계산됩니다. 이때도 배열과 마찬가지로 브로드캐스트 기능이 작동합니다.



1.3.5 브로드캐스트

넘파이에서는 형상이 다른 배열 간의 계산도 지원합니다. 2x2행렬 A에 스칼라값 10을 곱했습니다.

10

=

30

40

10

20

10

10

10

10

\*

3

4

1

2

=

\*

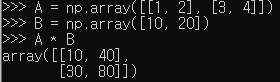
3

4

1

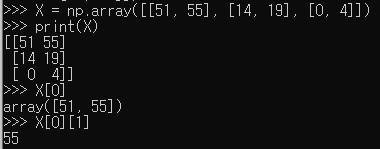
2

위 그림처럼 연산이 이뤄지는데, 이 기능을 브로드캐스트(Broadcast)라고 합니다.

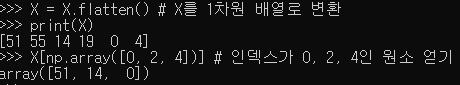


1차원 배열인 B가 2차원 A와 똑 같은 형상인 [[10, 20], [10, 20]]으로 변형된 후 연산이 이뤄집니다.

1.3.6 원소 접근



배열에 접근하는 것처럼 원소의 인덱스를 통해 원소에 접근할 수 있습니다.

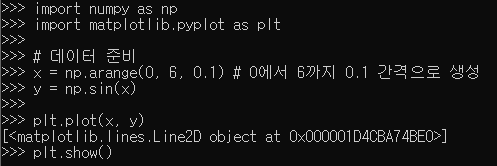


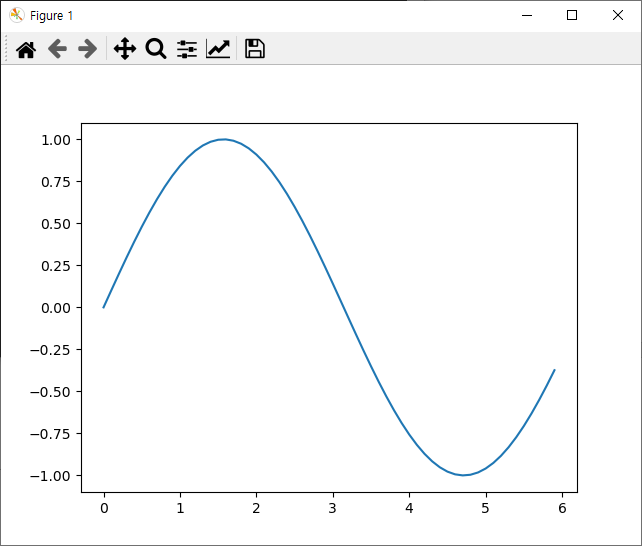


1.4 matplotlib

Matplotlib는 그래프를 그려주는 라이브러리로 데이터 시각화를 위해 사용합니다.

1.4.1 단순한 그래프 그리기





1.4.2 pyplot의 기능

