day1. python 기초

머신러닝학습에 필요한 python 기초

리스트(list)

- 여러 개의 자료를 하나의 변수로 관리하는 자료형
- 대괄호 [ ] 를 사용해서 생성
- 인덱스를 사용해서 위치값을 표시, 선택하고 출력
- 인덱스는 0부터 counting
- 인덱스의 범위를 지정 [1:N]
- 인덱스 위치의 값을 변경 가능
- 리스트는 연결해서 하나의 리스트로 조립 가능 : list1+list2
- 리스트의 요소를 반복해서 출력가능 : list1*3

튜플(tuple)

- 여러 개의 자료를 하나의 변수로 관리할 때 사용
- 튜플은 괄호 ( )를 사용해서 데이터집합을 생성
- 리스트(list)와의 차이점은 데이터를 변경할 수 없다 점
- .append() 등 값을 변경하는 메소드는 사용할 수 없고, 값을 조회하는 .count(), .index() 메소드만 사용 가능
- 인덱스를 사용해서 요소를 출력
- 인덱스의 범위를 지정해서 해당 요소를 출력
- 튜플을 연결해서 출력 가능: tuple1 + tuple2
- 튜플을 반복해서 요소값을 생성 가능 : tuple1*3
- 요소값의 개수를 반환 : .count()
- 요소값의 인덱스를 반환 .index()

집합(set)

- 여러 개의 자료를 하나의 변수로 관리하는 자료형
- 수학의 집합과 유사
- 집합내의 요소는 중복된 데이터를 가질 수 없고, 순서가 없다.
- 순서와 관련된 인덱스기호([ ])를 사용할 수 없고, 중복 데이터를 만드는 +, *를 사용할 수 없다.
- 하지만, in, not in, len()은 사용할 수 있습니다.

사전(dic)

- 사전은 집합의 일종, 키와 값이 하나의 데이터 요소를 구성
- 사전의 요소는 순서가 없고 중복된 데이터를 갖지 않아서, 중복 데이터를 만드는 +, *를 사용할 수 없다
- 키를 이용하여 요소의 인덱스([ ])처럼 처리

리스트(list) - example

```
# 리스트 생성하기
>>> list1 = [1,'two',3,'four',5,'six']
>>> list1 = []
# 리스트 출력하기
>>> print(list1)
>>> print(list1[3])
>>> print(list1[-2])
# 범위출력
>>> print(list1[1:3])
>>> print(list1[3:])
# 값변경
>>> list1[1] = 2
>>> print(list1)
```

리스트(list) - example

```
# +로 연결, *로 반복
>>> list1 = [1,2,3]
>>> list2 = [4,5,6]
>>> list3 = list1 + list2
>>> print(list3)
>>> list1 = [1,2,3]
>>> list4 = list1*3
>>> print(list4)
# 리스트 길이
>>> list1 = [1,'two',3,'four',5,'six']
>>> print(len(list1))
>>> 1 in list1
>>> 2 in list1
>>> 1 not in list1
>>> 2 not in list1
```

리스트(list) - example

```
# .append() ## 끝에 데이터를 추가
>>> list1 = [1,2,3,4,5]
>>> list1.append(6)
# 리스트 요소 지우기
>>> del list1[3:4]
# .clear()   ## 리스트 요소를 모두 지우기
>>> list1.clear()
# .copy()
>>> list1 = [1,2,3,2,3,2]
>>> list2 = list1.copy()
# .count()
>>> list2.count(2)
# .extend() ## 리스트확장
>>> list1 = [1,2,3]
>>> list2 = [4,5,6]
>>> list1.extend(list2)
```

리스트(list) - example

```
# .index ## 끝에 데이터를 추가
>>> list1 = ['a','b','c','d','e'] ## index: 0, 1, 2, 3, 4,
>>> list1.index('c')
>>> list2 = ['a','b','c','a','b'] ##  중복된 값이 있는 경우 첫번째 인덱스 출력
>>> list2.index('b')
# .insert()
>>> list1 = [1,2,3,4,5]
>>> list1.insert(2,100)
# .pop() ## 요소의 마지막 데이터를 출력하고 인덱스 삭제
>>> list1 = [1,2,3,4,5]
>>> list1.pop()
>>> list2 = ['a','b','c','d','e']
>>> list2.pop(3)
```

리스트(list) - example

```
# .remove() # 지정된 값 지우기
>>> list1 = [1,2,3,4,5]
>>> list1.remove(3)
# .reverse() ## 역순으로 변경
>>> jb = [3,5,2,6,7]
>>> jb.reverse()
# .sort() ## 오름차순, 내림자순 정렬
```

- `>>> jb = [3,5,2,6,7]`
- `>>> jb.sort()`
- `>>> jb = [3,5,2,6,7]`
- `>>> jb.sort(reverse=`**`True`**`)`

튜플(tuple) - example

```
# tuple생성하기
>>> tuple1 = (1,2,3,4,5)
>>> type(tuple1)
>>> tuple2 = ()
>>> tuple2 = tuple()
>>> print(tuple1)
# 범위 출력
>>> print(tuple1[2:5])
```

튜플(tuple) - example

```
# 튜플길이
>>> tuple1 = (1,2,3,4,5)
>>> print(len(tuple1))
# 값의 유무확인
>>> 1 in tuple1
>>> 1 not in tuple1
# +연결, *반복
>>> tuple1 = (1,2,3)
>>> tuple2 = (4,5,6)
>>> tuple3 = tuple1 + tuple2
>>> tuple4 = tuple1*3
```

튜플(tuple) - example

```
# 요소 개수
>>> tuple1 = (1,2,3,2,1)
>>> tuple1.count(2)
# 요소 번호
>>> tuple1.index(3)
```

집합(set) - example

```
# 집합생성하기
>> set1 = {1,2,3,4,5}
>> type(set1)
>> set2 = set()
# 집합의 길이
>>> len(set1)
# 값의 유무확인
>>> 1 in set1
>>> 1 not in set1
```

집합(set) - example

```
# 원소 추가
>>> set1 = {1,2,3}
>>> set1.add(4)
>>> set1.update([4,5,6])
# 원소 제거
>>> set1 = {1,2,3}
>>> set1.remove(3)
>>> set1.discard(2)
>>> set1.pop()
>>> set1.clear()
```

집합(set) - example

```
# 교집합, 합집합, 차집합
>>> set1 = {1,2,3}
>>> set2 = {3,4,5}
>>> set1 & set2
>>> set1 | set2
>>> set1.union(set2)
>>> set1 – set2
>>> set1.difference(set2)
```

사전(dic) - example

```
# 사전생성하기
>>> dic1 = {1:"one", 2:"two", 3:"three"}
>>> type(dic1)
>>> dic1 = {}
# 길이구하기
>>> len(dic1)
# 값의 유무확인
>>> 1 in dic1
>>> 1 not in dic1
```

사전(dic) - example

```
# 사전 추가하기
>>> dic1 = {1:"one", 2:"two", 3:"three"}
>>> dic1[4]=“for"
# 값 수정하기
>>> dic1[4]=“four”
# 항목 지우기
>>> del dic1[2]
>>> dic1.clear()
```

사전(dic) - example

```
# 아이템 출력하기
>>> dic1 = {1: " one " , 2: " two " , 3: " three " }
>>> dic1.items()
>>> dic1.keys()
>>> dic1.values()
 # 사전 내에 사전을 추가하기
>>> dic2 = {4:"four", 5:"five", 6:"six"}
>>> dic1.update(dic2)
```

제어구문 if

```
if condition:
    statement
```

```
if condition:
    statement 1
else:
    statement 2
```

```
if condition 1:
    statement 1
elif condition 2:
    statement 2
```

```
if condition 1:
    statement 1
elif condition 2:
    statement 2
else:
    statement 3
```

제어구문 for - 반복

```
>>> for i in range(4):
                print("Hello")
>>> for i in range(2):
                for j in range(2):
                        print(i, j)
# 구구단 출력
>>> for i in range(2, 4):
                for j in range(1, 10):
                        print(i, "X", j, "=", i*j)
# 리스트 출력
>>> list1 = ['one', 'two', 'three']
>>> for i in list1:
  ...  print(i)
```

제어구문 while - 반복

```
>>> i = 1
>>> while i < 10:
 ...     print(i, end=' ')
 ...     i = i + 1
# 구구단
>>> i = 1
>>> while i < 3:
...     i = i + 1
...     j = 1
...     while j < 10:
...         print(i, 'X', j, '=', i*j)
...         j = j + 1
```

제어구문 while - 반복

```
# 구구단, break
>>> x = int(input('Number : '))
>>> i = 1
>>> while i < 3:
...     i = i + 1
...     j = 1
...     while j < 10:
...         print(i, 'X', j, '=', i*j)
...         j = j + 1
...     if i == x:
...         break
```

제어구문 while - 반복

```
# continue
>>> i = 0
>>> while i < 10:
>>>     i = i + 1
>>>     if i % 2 == 0:
>>>         continue
>>>     print(i, end=" ")
```

입력과 출력

```
>>> x = input()
# 입력을 문자열로 저장
>>> x = input('name : ')
# 입력을 숫자로 저장
>>> x = int(input('number : '))
# 입력을 실수로 저장
>>> x = float(input('number : '))
```

입력과 출력

```
# 출력 1
>>> list1 = [1,2,3]
>>> for I in list1:
...  print( ' n: ' , I, end= '    ' )
>>> for I in list1:
...  print( ' n: ' , I, end='₩n ' )
# 출력 2
>>> a = 'abc'
>>> n = 123
>>> print('a:', a, 'n:', n)
# 출력 3
>>> print('text: %s' %a)
>>> print('numb: %d' %n)
>>> print('text: %s, numb: %d' %(a, n))
>>> print('{0}: %s, {1}: %d'.format('text', 'numb') %(a, n))
```

함수

- range(stop)
- range(start, stop)
- range(start, stop, step)

함수 – example

```
# range() 함수를 사용한 리스트 생성
>>> list(range(4))
>>> list(range(1, 11))
>>> list(range(1, 11, 2))
>>> list(range(10, -5, -3))
# 반복구문의 횟수 정할 때 사용 예
>>> for i in range(4):
...   print("Hello!")
```

함수

- 사용자 정의 함수
- **def** function_name(parameter) :
- *# code*

사용자 정의 함수 – example

```
>>> def hello() :
>>>         print('Hello')

>>> def sum(x, y):
...     print(x + y)
>>>
>>> x = int(input('input x: '))
input x: 10
>>> y= int(input('input y: '))
input y: 20
>>> sum(x, y)
```