

"따라하며 배우는 파이썬과 데이터과학(개정판)_1쇄" 정오표

책 내용에 있는 오류를 정리한 페이지 입니다. 불편을 끼쳐드려 대단히 죄송합니다. 다음 인쇄 때
수정하여 반영하겠습니다.(2025년 4월 28일)

오류 : 잘못된 코드 오류, 실행 결과 오류, 잘못된 설명, 잘못된 참조 번호(그림, 코드 등)

오류 페이지 : 34 page

오류 위치와 오류 : 아래 6줄(파이썬 코드의 오류시 출력되는 결과를 보여주어야 함)

```
>>> import turtle          # turtle 모듈을 사용하기 위해 준비한다
>>> t = turtle.Turtle()    # turtle 모듈에 있는 Turtle 클래스 객체를 t로 생성한다
>>> t.shape('turtle')      # Turtle 클래스 객체인 t의 모양을 거북이 모양을 설정한다
```

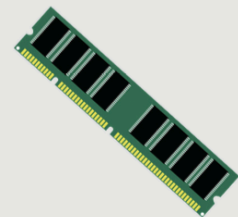
으로

오류 페이지 : 49 page

오류 위치와 오류 :

메인 메모리와 주소 그리고 변수
잠깐 - 변수를 사용하면 좋은 이유 1

컴퓨터의 데이터는 오른쪽 그림에 나타난 것과 같은 컴퓨터의 핵심적인 부품인 **메인 메모리**에 저장한다. 이렇게 메모리에 데이터를 저장한 곳의 위치를 **메모리 주소** **memory address**라고 한다. 이 주소는 보통 0x12FF06이라는 형식의 16진수로 표현한다. 컴퓨터의 입장에서는 0x12FF06라는 주소를 이해하는 것이 전혀 어려운 일이 아니지만, 사람의 입장에서는 그 뜻을 이해하기 힘들고, 다른 주소와 구별도 어렵다. 이 메모리 공간 0x12FF06의 위치에 사람이 인지하기 쉬운 'radius'라는 이름을 붙여보자. 반지름을 의미하는 이 이름을 붙여주면 이 공간에 무엇을 저장해야 하는지, 그리고 읽은 값이 무엇을 의미하는지 다루기가 훨씬 편리하다. 이것이 변수가 사용되는 이유이다.



오류 페이지 : 59 page

오류 위치와 오류 :

```

>>> print("Hello\tWorld!") # \t는 띄어쓰기를 위한 탭문자이다
Hello      World!
>>> "Hello".upper()      # .upper()는 문자열을 대문자로 만드는 기능이 있다
'HELLO'
>>> "Hello".lower()      # .lower()는 문자열을 소문자로 만드는 기능이 있다
'hello'

```

오류 페이지 : 61 page

오류 위치와 오류 :

```

>>> message = '철수가 "안녕"이라고 말했습니다.'
>>> message
'철수가 "안녕"이라고 말했습니다.'
>>> print(message)
철수가 "안녕"이라고 말했습니다.

```

오류 페이지 : 75 page

오류 위치와 오류 :

```

>>> x = y = 100          # 여러 변수에 동일한 값을 할당하는 다중 할당문
>>> x, y
(100, 100)
>>> n1, n2 = 100, 200    # 여러 변수에 한꺼번에 여러 값을 할당하는 동시 할당문
>>> n1, n2
(100, 200)

```

오류 페이지 : 82 page

오류 위치와 오류 :

```

>>> num /= 2             # 4000 / 2 연산을 수행하여 그 결과를 num에 할당
>>> num
2000.0
>>> num //= 2            # 4000 / 2 연산을 수행하여 그 결과를 num에 할당
>>> num
1000.0

```

오류 페이지 : 87 page

오류 위치와 오류 :

```

>>> num = 2    # 정수 2는 이진수 10(2)으로 표현됨
>>> num <<= 1   # 정수 2를 1비트 왼쪽 이동시키면 100(4)가 됨
>>> num
4
>>> num <<= 1   # 정수 4를 1비트 더 왼쪽 이동시키면 1000(8)이 됨
>>> num
8
>>> num >>= 1   # 정수 8를 1비트 오른쪽 이동시키면 100(4)가 됨
>>> num
4
>>> num >>= 2   # 정수 4를 2비트 오른쪽 이동시키면 001(1)이 됨
>>> num
1

```

오류 페이지 : 94 page

오류 위치와 오류 : 오타 : 길이 -> 깊이

- 1) 모서리의 길이가 13인 정육면체
- 2) 모서리의 길이가 22인 정육면체
- 3) 가로, 세로, **길이**가 각각 17, 25, 16인 직육면체

오류 페이지 : 106 page

오류 위치와 오류 :

```

import turtle                # turtle 모듈을 가져온다

tur1 = turtle.Turtle()      # 터틀 객체를 생성함
tur1.shapesize(5, 3)        # 가로 크기가 원래 크기의 5배, 세로 크기가 원래 크기의 3배로 커짐
tur1.forward(100)

```

오류 페이지 : 109 page

오류 위치와 오류 :

```
s = turtle.textinput("", "숫자를 입력하시오: ") # turtle 모듈의 textinput()함수
n = int(s)

if n > 0 :
    t.goto(100, 100)
if n == 0 :
    t.goto(100, 0)
if : n < 0:
    t.goto(100, -100)

turtle.done() # t.done()이 아닌 터틀 스크린의 done() 함수 호출임
```

오류 페이지 : 168 page

오류 위치와 오류 :

```
def weeklyPay(rate, hour):
    if (hour > 30):
        money = rate*30 + 1.5*rate*(hour-30)
    else:
        money = rate*hour
    return money

r = int(input("시급을 입력하시오: ")) # 시급입력받기
h = int(input("근무 시간을 입력하시오: ")) # 근무시간 입력받기
print("주급은 " + str(weeklyPay(rate = r, hour = h))) # 키워드 인자 사용
```

공백 한 칸으로 수정

오류 페이지 : 197 page

오류 위치와 오류 :

```
>>> print('s[-2:] =', s[-2:]) # 마지막 두 문자열을 추출해보자
s[0:2] = on
>>> print('s[-1:-3:-1] =', s[-1:-3:-1]) # 마지막 두 문자열을 꺼꾸로 추출
s[-1:-3:-1] = no
```

오류 페이지 : 227 page

오류 위치와 오류 :

```

        print(dictionary[inputStr])
    else :
        print('{}가 사전에 없습니다.'.format(inputStr))
    elif command == 'q':
        break
    else :
        print('입력 오류가 발생했습니다.')
print("사전 프로그램을 종료합니다.")

```

오류 페이지 : 267 page

오류 위치와 오류 :

- \.는 이스케이프 문자로서, . 문자를 나타낸다. 이는 이메일 주소에서 도메인을 구분하는 . 문자를 나타낸다.
- 마지막으로 \w+가 나오는데, 이는 앞에서와 같이 알파벳, 숫자, 언더스코어를 나타내는 메타 문자인 \w가 1번 이상 반복됨을 나타낸다.

오류 페이지 : 288 page

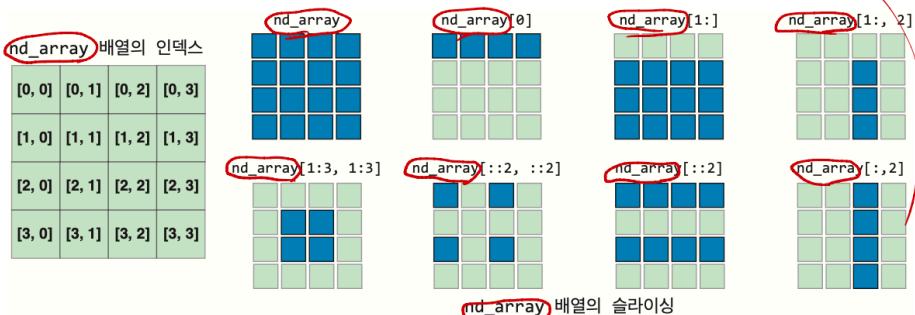
오류 위치와 오류 :

```

>>> np_array[1, 1:3]
array([6, 7])

```

아래의 그림은 4x4 크기의 np_array라는 ndarray와 이 ndarray의 인덱싱 및 슬라이싱 결과를 보여준다.



오류 페이지 : 289 page

오류 위치와 오류 :

남을 알 수 있다.

```
>>> np_array = np.array([[1,2,3], [4,5,6], [7,8,9]])
>>> np_array > 5
array([[False, False, False],
       [False, False, True],
```

<[True, True, True]>

추가

오류 페이지 : 289 page

오류 위치와 오류 :

```
>>> np_array[0] % 2 == 0
array([[False, True, False],
       [ True, False, True],
       [False, True, False]])
```

오류 페이지 : 360 page

오류 위치와 오류 :

```
df_1 = pd.DataFrame({'item' : ['ring0', 'ring0', 'ring1', 'ring1'],
                      'type' : ['Gold', 'Silver', 'Gold', 'Bronze'],
                      'price' : [20000, 10000, 50000, 30000]})
```

print(df_1)

	item	type	price
0	ring0	Gold	20000
1	ring0	Silver	10000
2	ring1	Gold	50000
3	ring1	Bronze	30000

pivot 이전의 데이터프레임

pivot 이후의 데이터프레임

앞의 코드에 아래 코드를 이어서 만들면 *df-2* **df2**는 변경된 테이블을 담게 될 것이다.

```
df_2 = df_1.pivot(index='item', columns='type', values='price')
print(df_2)
```

	type	Bronze	Gold	Silver
item				
ring0		NaN	20000.0	10000.0
ring1		30000.0	50000.0	NaN

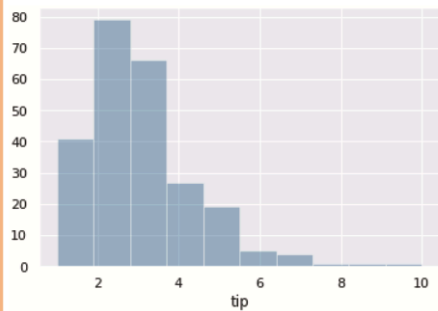
오류 페이지 : 384 page

오류 위치와 오류 :

주로 어느 정도의 요금을 팁으로 주는지 그 분포를 살펴볼 수 있을 것이다.



```
sns.distplot(tips['tip'], kde=False, bins=10);
```



오류 페이지 : 417 page

오류 위치와 오류 :

LAB¹⁴⁻²

주택의 실면적과 대중교통 접근성 그리고 가격

실습 시간



일반적으로 주택의 실면적이 크고 대중교통과의 접근성이 좋을 경우 거래가격은 높은 편이다. 다음과 같이 비슷한 지역의 10개 주택 샘플이 주어졌을 경우에 대하여 선형회귀 모델을 만들고 주택 거래 가격을 예측해 보도록 하자. 주택의 면적은 제곱미터 단위이며, 버스나 지하철과 같은 대중교통과의 접근성은 1에서 10까지로 표시하였다. 이 때 숫자가 높을수록 접근성이 더 좋은 편이다. 그리고 주택의 거래 가격 단위는 억 단위이다.

오류 페이지 : 457 page

오류 위치와 오류 :

```
train_images[0] = [5, 0, 4, 1]
```

맞히는

출력 결과 학습용 이미지의 정답에 해당하는 레이블이 5 0 4 1이라는 것을 확인할 수 있었다. 이런 형태의 이미지와 정답이 6만 개 주어져 있는 것이 MNIST 데이터이므로 우리는 이 필기체 이미지를 학습하여 정답을 알아맞히는 딥러닝 모델을 만들어야 할 것이다. 딥러닝 모델의 정확도는 학습에 사용되지 않은 새로운 데이터인 test_images를 얼마나 정확하게 인식하는가로 판별할 것이다.