

파이썬1 목차

2018년 8월 17일 금요일 오후 3:54

- 파이썬 149제

- 1. 대화식 모드로 프로그래밍 하기
- 2. 텍스트 에디터로 프로그래밍 하기
- 3. 변수명 만들기
 - 변수 이름 규칙
 - 일반 변수
 - 리스트 변수
- 4. 변수에 값 대입하기
- 5. 주석 처리하기(#)
- 6. 자료형 개념 배우기
- 7. 자료형 출력 개념 배우기(print)
- 8. 들여쓰기 개념 배우기

- 파이썬 149제(if, for, while)

- 9. if문 개념 배우기
- 10. if문 개념배우기2
- 11. for문 개념배우기1
- 14. for문 개념배우기2(for ~ continue ~ break)
- 15 while문 개념 배우기 (while ~ continue ~ break)
- 16. None의 개념배우기
- 17. 정수형 자료 이해하기
- 18. 실수형 자료 이해하기
- 19. 대입 연산자 이해하기
- 20. 사칙연산 이해하기(+, -, *, /, **)

- 파이썬 149제(연산자)

- 21. 연산자 축약 이해하기(+=, -=, *=, /=)
- 22. True와 False 이해하기
- 23. 관계 연산자 이해하기(==, !=, <, <=, >, >=)
- 24. 논리 연산자 이해하기(and, or, nor)
- 25. 스퀀스 자료형 이해하기

- 파이썬 149제(시퀀스)

- 26. 시퀀스 자료 슬라이싱 이해하기
- 27. 시퀀스 자료 연결하기 (+)

- [28. 시퀀스 자료 반복 이해하기\(*\)](#)
- [29. 시퀀스 자료 크기 이해하기\(len\)](#)

- [파이썬 149제\(문자열\)](#)

- [30. 문자열의 멤버 체크 이해하기 \(in\)](#)
- [31. 문자열 이해하기](#)
- [32. 문자열 포매팅 이해하기](#)
- [33. 이스케이프 문자 이해하기](#)
- [34. 리스트 변수란?](#)
- [35. 파이썬과 오라클 그룹함수 비교](#)
- [36. 튜플 이해하기](#)
- [37. 사전 이해하기](#)

- [파이썬 149제\(함수\)](#)

- [38. 함수 이해하기\(def\)](#)
- [39. 함수 인자\(매개변수\) 이해하기](#)
- [40. 지역변수와 전역변수 이해하기](#)
- [41. 함수 리턴값 이해하기\(return\)](#)
- [42. 파이썬 모듈 이해하기](#)
- [43. 파이썬 패키지 이해하기](#)
- [44. 파이썬 모듈 임포트 이해하기1\(import\)](#)
- [45. 파이썬 모듈 임포트 이해하기2\(from~import\)](#)
- [46. 파이썬 모듈 임포트 이해하기3\(import ~ as\)](#)
- [47. 파일 열고 닫기\(open, close\)](#)

- [파이썬 149제\(클래스\)](#)

- [48. 클래스 이해하기\(class\)](#)
- [49. 클래스 멤버와 인스턴스 멤버 이해하기](#)
- [static method](#)
- [50. 클래스 메소드 이해하기](#)
- [51. 클래스 생성자와 소멸자 이해하기](#)
- [52. 클래스 상속 이해하기](#)

- [파이썬 149제\(예외 처리\)](#)

- [53. 예외처리 이해하기\(try~except~\)](#)
- [54. 예외처리 이해하기2\(try~except~else\)](#)
- [55. 예외처리하기3 \(복수개의 except절 사용하기\)](#)
- [56. 예외처리하기4 \(try~except~finally\)](#)

- [57. 예외처리 이해하기5\(try~except Exception as e\)](#)
- [58. 사용자 정의 예외 처리](#)

- [파이썬 149제\(input,type\)](#)

- [59. 사용자 입력받기\(input\)](#)
- [60. 자료형 확인하기\(type\)](#)

- [파이썬 149제 \(% ,divmod\)](#)

- [61. 나눗셈에서 나머지만 구하기\(%\)](#)
- [62. 몫과 나머지 구하기\(divmod\)](#)

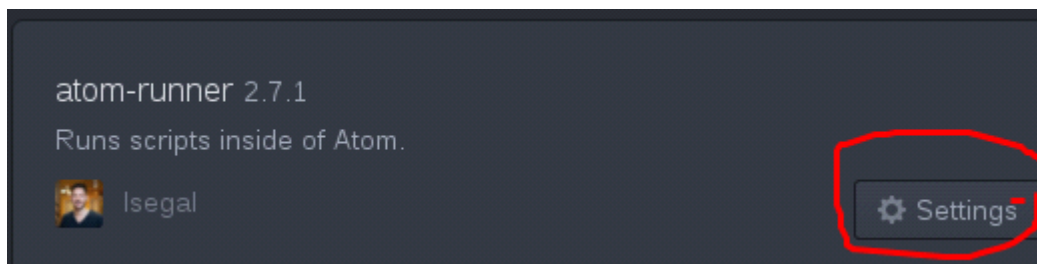
- [파이썬 149제\(pandas\)](#)

- [63. Pandas를 이용한 데이터 검색](#)
- [64. Pandas를 이용해서 조건절 사용 데이터 검색 방법](#)
- [65. Pandas를 이용해서 조건절 사용 데이터 검색방법](#)
- [66. pandas를 이용한 조인](#)
- [67. pandas를 이용한 서브쿼리](#)

Centos에서 Atom설치 하는법

2018년 7월 12일 목요일 오전 9:58

1. vm웨어로 Centos를 실행한다. (xshell로 하면 안됨)
2. 터미널 창을 열고 루트 권한으로 접속한다.
3. `yum install wget git -y`
4. wget https://github.com/atom/atom/releases/download/v1.18.0/atom.x86_64.rpm
5. `yum localinstall atom.x86_64.rpm -y`
6. atom
아톰이라 쳐서 잘 실행 되는지 확인한다.
(아톰은 항상 루트 권한으로 접속을 권장)
7. (트러블 슈팅, 아톰 지우기) 안되는 사람들만 수행
`rpm -qa | grep atom`
`rpm -e atom-1.18.1-0.1.x86_64`
8. 아톰에서
`edit -> preference -> install`
 - 1) atom-runner
 - 2) python tools
 - 3) autocomplete-python
9. settings -> packages _> atom-runner settings -> configuring 부분 찾기
'runner' 부분 ~ 끝까지 복사



```
# Make sure to place 'runner' under '*' object.
'*':
  ...
  'runner':
    'scopes':
      'coffee': 'coffee'
      'js': 'node'
      'ruby': 'ruby'
      'python': 'python'
      'go': 'go run'
      'shell': 'bash'
      'powershell': 'powershell -noninteractive -nopprofile -c -'
    'extensions':
      'spec.coffee': 'mocha'
      'ps1': 'powershell -file'
```

10. edit 탭 -> config... 들어가기 맨 마지막 줄에 복사해서 그대로 붙여 넣기.
(해당 줄의 맨 앞에 공백이 없어야 함.)

```
1  "*":
2    core:
3      telemetryConsent: "no"
4      "exception-reporting":
5        userId: "6f034049-32e2-4fb6-bdbd-4f1fa52710cc"
6      welcome:
7        showOnStartup: false
8      'runner':
9        'scopes':
10         'coffee': 'coffee'
11         'js': 'node'
12         'ruby': 'ruby'
13         'python': 'python'
14         'go': 'go run'
15         'shell': 'bash'
16         'powershell': 'powershell -noninteractive -nopprofile -c -'
17        'extensions':
18         'spec.coffee': 'mocha'
19         'ps1': 'powershell -file'
20
```

11. 아무 파일이나 새로 만들고 오른쪽 아래에서 UTF-8 옆에 보면 언어 설정하는 부분 있음.
거기에서 언어 python으로 변경.



12. 아무 명령어나 입력하고 Alt+R로 실행해서 결과 바로 나타나는지 확인.

끝!!

통계

2018년 7월 23일 월요일 오전 10:14

2장. 데이터 수치 요약

문제1. 중심위치 측정이 필요한 이유는 무엇이고 중심위치 측도로 사용되는 방법에는 무엇이 있는가?

답)

데이터가 중심으로 얼마나 집중되어 있는지, 빈도가 가장 많은 값은 어느 값인지, 다른 데이터와 비교할 때 특이한 데이터는 없는지 살펴볼때 필요하다.

평균값, 중앙값, 최빈값, 범위, 사분위수, 분산 표준편차

numpy 사용법

numpy 모듈이란 ?

1. 숫자배열 연산을 위한 라이브러리 이다.
2. numpy array는 파이썬 리스트보다 속도가 빠르며 리스트 간 연산에 편의성을 제공한다.
3. sorting이나 indexing, intersection 을 찾는 경우에도 속도가 빠르다.
4. 평균, 중앙값, 최빈값, 사분위수, 분산, 표준편차를 찾는경우에도 리스트보다 훨씬 코드가 간단하고 속도가 빠르다.

ex1)

리스트나 튜플로부터 배열을 만들 수 있다.

```
import numpy as np
```

```
b=np.array([2,4,6,8])
```

```
print(b)
```

```
print(type(b))
```

```
[2 4 6 8]  
<class 'numpy.ndarray'>
```

ex2)

b배열의 차원수(랭크)를 확인한다.

```
print(b.ndim)
```

```
[2 4 6 8]  
<class 'numpy.ndarray'>
```

```
1
```

1차원 : 한 행으로 되어져있다.

ex3)

b 배열에 있는 값들의 총 개수를 확인한다.

```
print(b.size)
```

```
4
```

ex4)

b배열의 행렬 모양을 확인한다.

```
print(b.shape)
(4, )
```

문제2. 아래의 행렬을 만드시오.

보기)

```
1 2 3
4 5 6
```

답)

```
import numpy as np
a=np.array( [ [1,2,3], [4,5,6] ] )
print(a)
print(a.shape)
[[1 2 3]
 [4 5 6]]
(2, 3)
```

2행 3열

문제3. 아래의 행렬곱을 numpy로 수행하시오.

보기)

```
1 2 3   3 2 1
4 5 6   6 5 4 = ?
7 8 9   9 8 7
```

답)

```
import numpy as np
a=np.array( [ [1,2,3], [4,5,6], [7,8,9] ] )
b=np.array( [ [3,2,1], [6,5,4], [9,8,7] ] )
```

#두 행렬의 합

```
print(a+b)
[[ 4  4  4]
 [10 10 10]
 [16 16 16]]
```

#두 행렬의 빼기

```
print(a-b)
[[-2  0  2]
 [-2  0  2]
 [-2  0  2]]
```

#두 행렬의 곱(a*b)

```
print(a*b)
[[ 3  4  3]
 [24 25 24]
 [63 64 63]]
```

#두 행렬의 내적

```
print(np.dot(a,b))
[[ 42  36  30]
 [ 96  81  66]
 [150 126 102]]
```

```
#####
x = np.array([ 18,  5, 10, 23, 19, -8, 10,  0,  0,  5,  2, 15,  8,
               2,  5,  4, 15, -1,  4, -7, -24,  7,  9, -6, 23, -13,
               1,  0, 16, 15,  2,  4, -7, -18, -2,  2, 13, 13, -2,
               -2, -9, -13, -16, 20, -4, -3, -11,  8, -15, -1, -7,  4,
               -4, -10,  0,  5,  1,  4, -5, -2, -5, -2, -7, -16,  2,
               -3, -15,  5, -8,  1,  8,  2, 12, -11,  5, -5, -7, -4])
print(x.shape)
print(x.size)
( 78, )
78
#####
```

문제4. 위의 78개 요소를 갖는 numpy 배열의 평균값을 구하시오.

답)

```
np.mean(x)
0.6923076923076923
```

문제5. 책 70페이지의 10가구의 자녀수의 평균값을 numpy로 구하시오.

답)

```
import numpy as np

x=np.array([2,3,0,2,1,0,3,0,1,4])
print(np.mean(x))
1.6
```

문제6. 어느 소아과 병원의 주별 내원 환자수는 1,283 명, 1,002명, 942명, 768명, 295명 이었다. 주당 평균 내원 환자수는 얼마인가?

답)

```
import numpy as np
patient=np.array([1283, 1002, 942, 768, 295])
```



```
print(np.mean(patient))
```

```
858.0
```

ex) 쿵푸 교실 : 17살인 여학생이 자기와 같은 나이대인 수강생이 있는 교육 교실에 등록을 하려했는데 17, 17, 17, 18 83 --> 평균값이 올라가서

문제7. 아래의 데이터의 중앙값을 numpy배열로 구하시오.

보기)

p.76 예제

```
32, 37, 39, 40, 41, 41, 41, 42, 43, 43, 44, 45, 45, 45, 46, 47, 47, 49, 50, 51
```

답)

```
import numpy as np
```

```
b=np.array([32, 37, 39, 40, 41, 41, 41, 42, 43, 43, 44, 45, 45, 45, 46, 47, 47, 49, 50, 51])
```

```
print(np.median(b))
```

```
43.5
```

중앙값에 비해 평균이 가진 가장 큰 장점은 모평균을 추정할 때 평균이 더 정확성이 높다.

문제8. 중앙값이 평균값보다 중심위치를 나타내는 대표값으로 더 적합한 경우가 언제인가?

답)

데이터에 특이점이 포함되어 있는 경우

ex) 쿵푸교실

문제9. 평균값이 중앙값보다 중심위치를 나타내는 대표값으로 갖는 장점이 무엇인가 ?

답)

모 평균을 추정할 때 평균이 더 정확성이 높다. 같은 모집단으로 부터 여러 번 표본을 추출해보면 중앙값보다 평균이 표본 추출에 따른 변화가 더 적다.

문제10. 최빈값은 무엇인가?

답)

데이터 중에서 발생빈도가 가장 높은 값

문제11. p.78의 예제 2-8의 20명의 운전면허 시험점수의 최빈값을 numpy로 구해보시오.

보기)

```
68, 70, 70, 71, 69, 74, 71, 72, 70, 73, 71, 78, 70, 69, 70, 72, 74, 72, 69, 67
```

답)

```
import numpy as np
```

```
from scipy import stats
```

```
a=np.array([68, 70, 70, 71, 69, 74, 71, 72, 70, 73, 71, 78, 70, 69, 70, 72, 74, 72, 69, 67])
```

```
print(stats.mode(a))
```

```
ModeResult(mode=array([70]), count=array([5]))
```

문제12. 엄마와 아기가 함께하는 수영교실에 나이의 최빈값을 파이썬으로 구하시오.

보기)

```
1,1,1,2,2,2,2,3,3,31,31,32,32,32,32
```

답)

```
import numpy as np
```

```
from scipy import stats
```

```
a=np.array([1,1,1,2,2,2,2,3,3,31,31,32,32,32,32])
```

```
print(stats.mode(a))
```

```
ModeResult(mode=array([2]), count=array([4]))
```

문제13. 엄마와 아기와 함께하는 수영교실의 나이 데이터가 책 80페이지에 나오는 쌍봉우리형 그래프인지를 파이썬으로 시각화 하시오.

참조 코드)

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

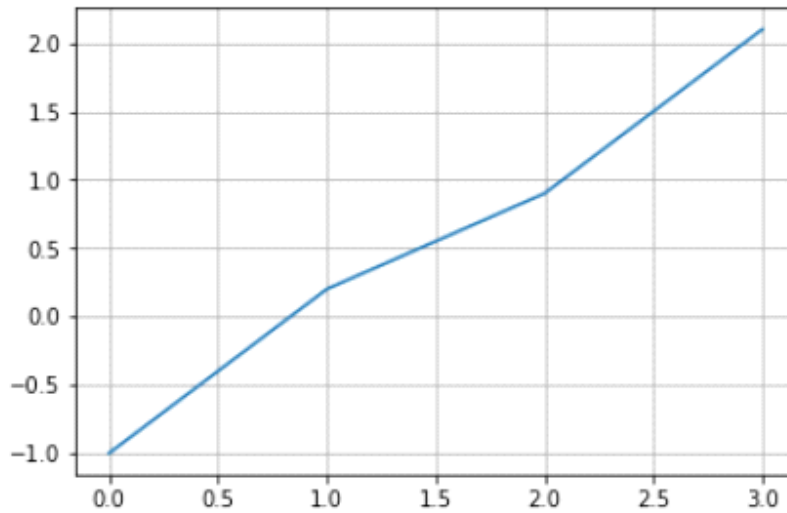
```
x = np.array([0,1,2,3])
```

```
y = np.array([-1,0.2,0.9,2.1])
```

```
plt.plot(x,y)
```

```
plt.grid(True)
```

```
plt.show()
```

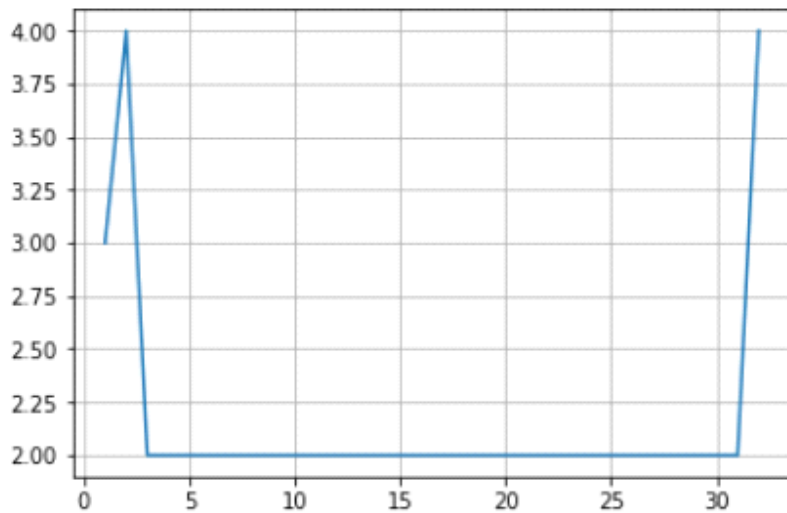


답)

```
import numpy as np
import matplotlib.pyplot as plt
```

```
x = np.array([1,2,3,31,32])
y = np.array([3,4,2,2,4])
```

```
plt.plot(x,y)
plt.grid(True)
plt.show()
```



• 파이썬으로 그래프 그리기

1. 라인 그래프 그리기
2. 원형 그래프 그리기
3. 막대 그래프 그리기
4. 히스토그램 그리기
5. plot 그래프 그리기

6. 사분위수 그래프 그리기

1. 라인 그래프 그리기

```
import numpy as np
import matplotlib.pyplot as plt
```

```
x = np.array([0,1,2,3])
y = np.array([-1,0.2,0.9,2.1])
```

```
plt.plot(x,y)
plt.grid(True)
plt.show()
```

문제14. 아래의 직선의 방정식을 시각화 하시오.

보기)
 $y=4x+10$

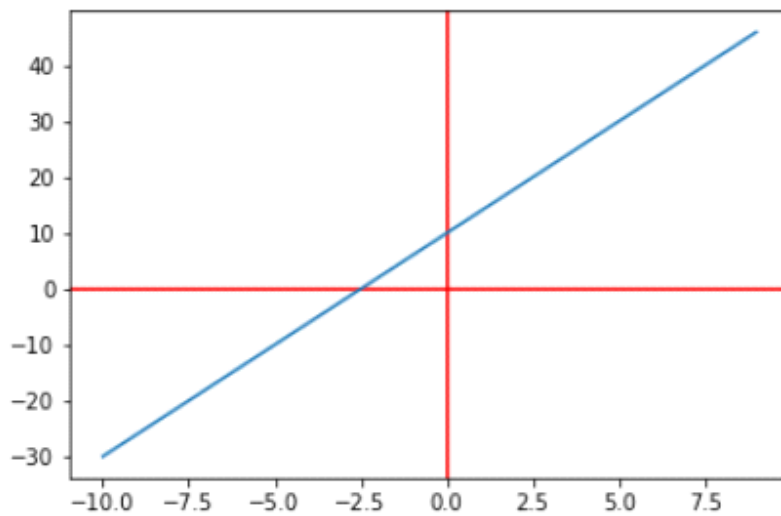
답)
import numpy as np
import matplotlib.pyplot as plt

```
plt.axvline(x=0, color = 'r')
plt.axhline(y=0, color = 'r')
```

```
x = np.arange(-10,10)
y = 4*x + 10
plt.plot(x, y) # draw line
print(x)
```

```
#plt.plot(x, y, 'o') # draw dots #
plt.show()
```

```
[-10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7  
 8 9]
```



`print(x)`

문제15. 아래의 식을 시각화 하시오.

보기)

$$y=2x^2 + 2$$

답)

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
plt.axvline(x=0, color = 'r', linestyle='--')
```

```
plt.axhline(y=0, color = 'r', linestyle='--')
```

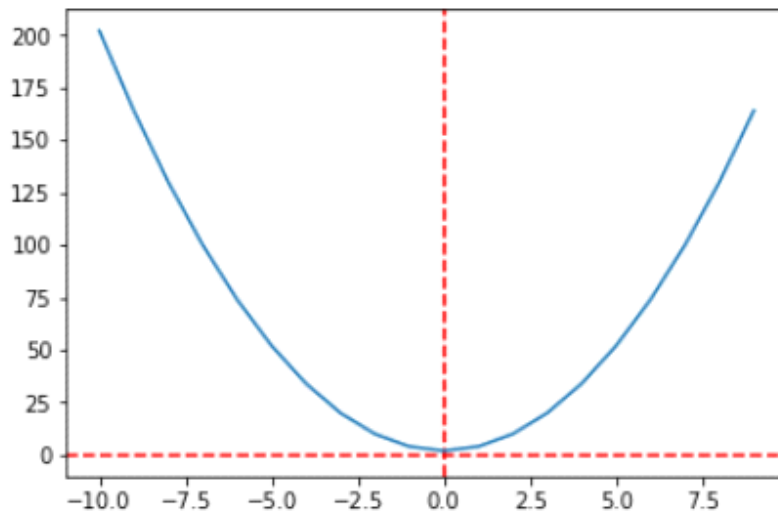
```
x = np.arange(-10,10)
```

```
y = 2*x**2 + 2
```

```
plt.plot(x, y) # draw line
```

```
#plt.plot(x, y, 'o') # draw dots #
```

```
plt.show()
```



2. 원형 그래프 그리기

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
from matplotlib import font_manager, rc
```

```
from matplotlib import style
```

```
#font_name = #font_manager.FontProperties(fname="c:/Windows/Fonts/malgun.ttf").get_name()
```

```
#rc('font', family=font_name)
```

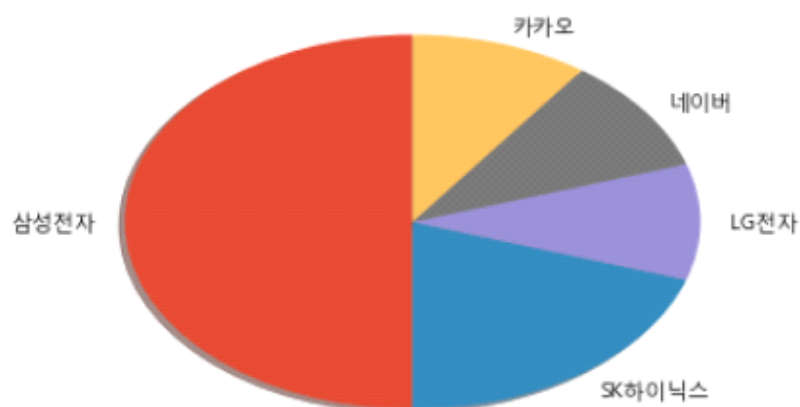
```
style.use('ggplot')
```

```
labels = ['삼성전자', 'SK하이닉스', 'LG전자', '네이버', '카카오']
```

```
ratio = [50, 20, 10, 10, 10]
```

```
plt.pie(ratio, labels=labels, shadow=True, startangle=90)
```

```
plt.show()
```



문제16. emp 테이블의 직업, 직업별 토탈월급을 원형 그래프로 그리시오.

답)

```
import pandas as pd
```

```
emp = pd.read_csv("C:\python_data\emp.csv")
```

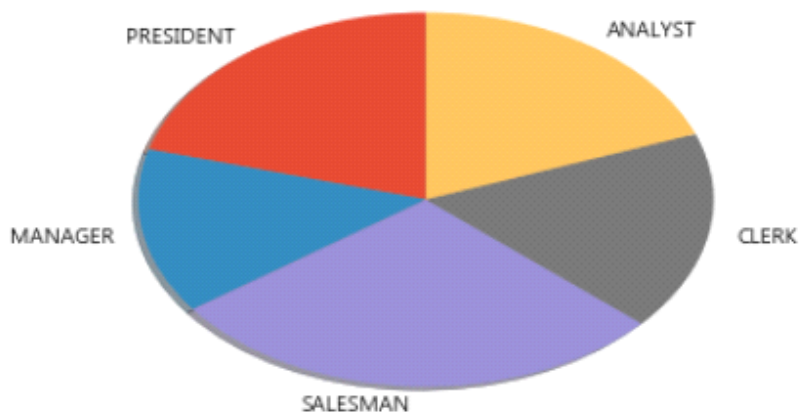
```
result=emp.groupby(['job'])['sal'].sum()
print(result)
```

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib import font_manager, rc
from matplotlib import style
```

```
style.use('ggplot')
job_name = emp['job'].unique()
```

```
plt.pie(result, labels=job_name, shadow=True, startangle=90)
plt.show()
```

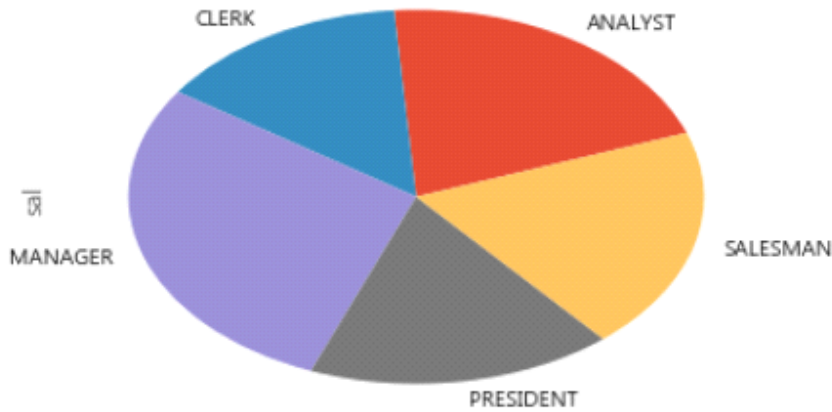
```
job
ANALYST      6000
CLERK        4150
MANAGER      8275
PRESIDENT    5000
SALESMAN     5600
Name: sal, dtype: int64
```



답2)

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

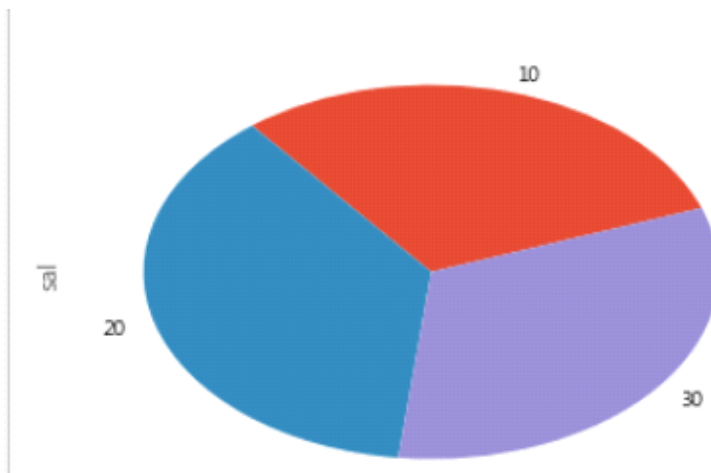
```
emp = pd.read_csv("C:\python_data\emp.csv")
emp.groupby('job').sal.sum().plot.pie(startangle=20)
plt.show()
```



문제17. 부서번호, 부서번호별 토탈월급을 가지고 파이썬으로 원형 그래프를 그리시오,
답)

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

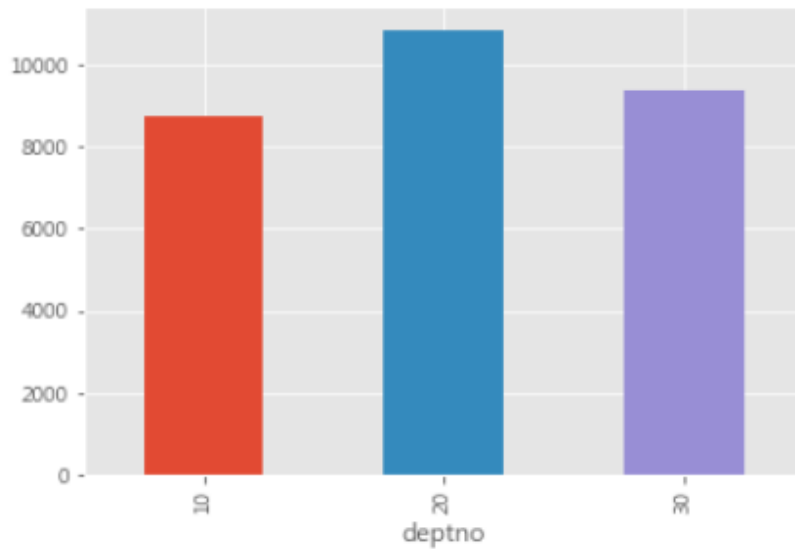
```
emp = pd.read_csv("C:\Wpython_data\emp.csv")
emp.groupby('deptno').sal.sum().plot.pie(startangle=20)
plt.show()
```



3. 막대 그래프 그리기

```
ex1)
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

```
emp = pd.read_csv("C:\Wpython_data\emp.csv")
emp.groupby('deptno').sal.sum().plot.bar()
plt.show()
```

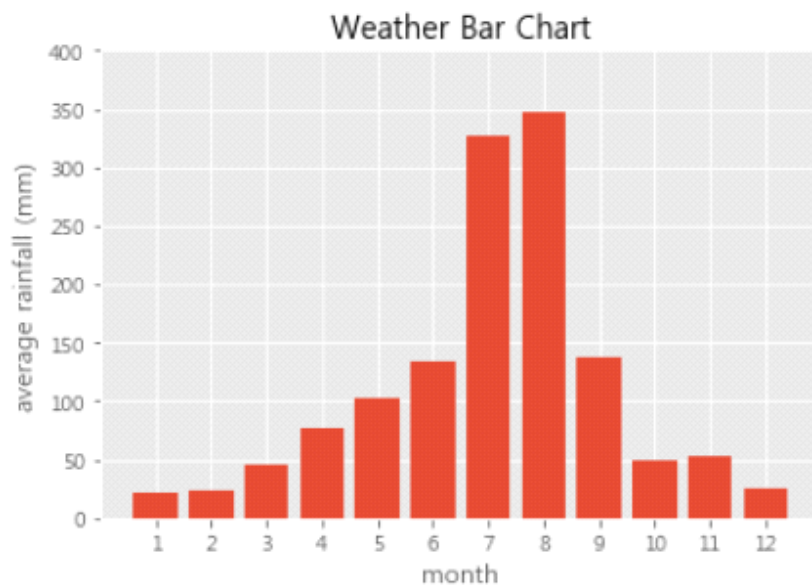



```
ex2)
import numpy as np
import matplotlib.pyplot as plt

y1_value = (21.6, 23.6, 45.8, 77.0, 102.2, 133.3, 327.9, 348.0, 137.6, 49.3, 53.0, 24.9)
x_name=('1', '2', '3', '4', '5','6', '7', '8', '9', '10', '11', '12')
n_groups = len(x_name)
index = np.arange(n_groups)

plt.bar(index, y1_value, tick_label=x_name, align='center')

plt.xlabel('month')
plt.ylabel('average rainfall (mm)')
plt.title('Weather Bar Chart')
plt.xlim( -1, n_groups)
plt.ylim( 0, 400)
plt.show()
```



문제18. 아기와 엄마와의 수영교실을 막대 그래프로 그리시오.

답)

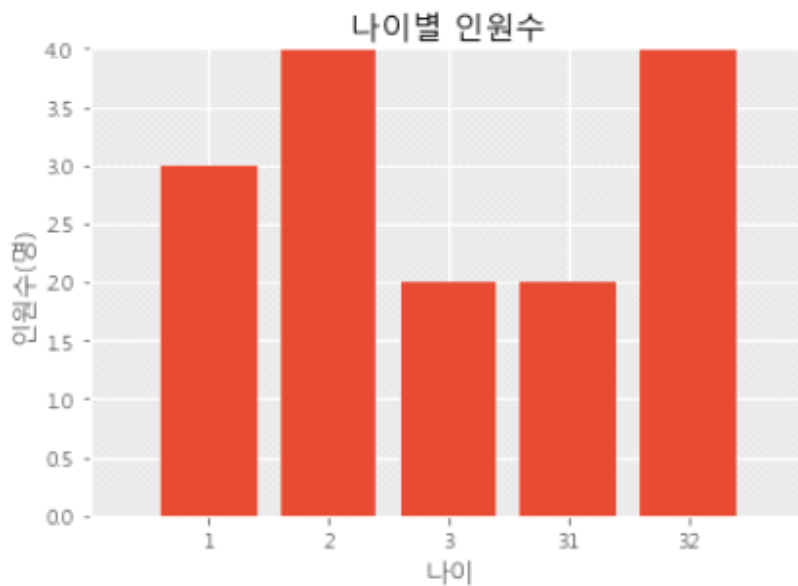
```
import numpy as np
import matplotlib.pyplot as plt

b = np.array([3,4,2,2,4])
a = np.array([1,2,3,31,32])

n_groups = len(a)
index = np.arange(n_groups)

plt.bar(index, b, tick_label=a, align='center')

plt.xlabel('나이')
plt.ylabel('인원수(명)')
plt.title('나이별 인원수')
plt.xlim( -1, n_groups)
plt.ylim( 0, 4)
plt.show()
```



문제19. 81페이지의 자료1 데이터를 막대 그래프로 그리시오.

보기)

자료 1 : 55,56,57,58,59,60,60,60,61,62,63,64,65

답)

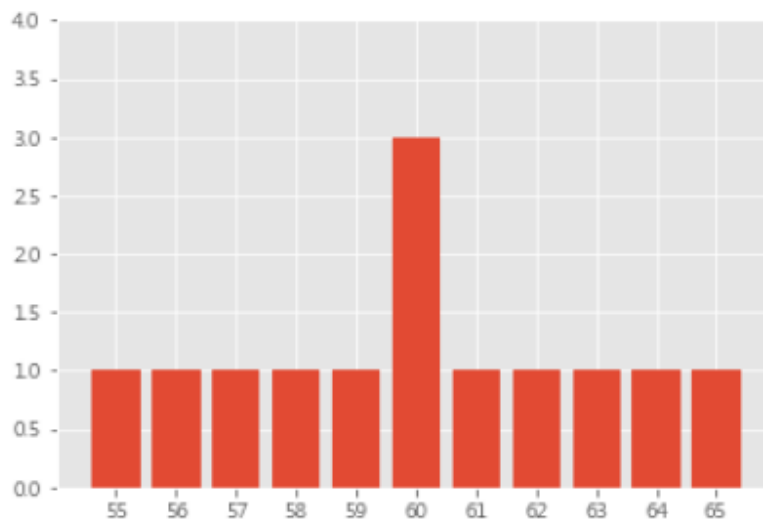
```
import numpy as np
import matplotlib.pyplot as plt
```

```
b = np.array([1,1,1,1,1,3,1,1,1,1])
a = np.array([55,56,57,58,59,60,61,62,63,64,65])
```

```
n_groups = len(a)
index = np.arange(n_groups)
```

```
plt.bar(index, b, tick_label=a, align='center')
```

```
# plt.xlabel('나이')
# plt.ylabel('인원수(명)')
# plt.title('나이별 인원수')
plt.xlim( -1, n_groups)
plt.ylim( 0, 4)
plt.show()
```



답2)

```
import numpy as np
import matplotlib.pyplot as plt
```

```
b = np.array([1,1,1,1,1,3,1,1,1,1])
a = np.array([55,56,57,58,59,60,61,62,63,64,65])
```

```
n_groups = len(a)
#index = np.arange(n_groups)
```

```
plt.bar(a, b, tick_label=a, align='center')
```

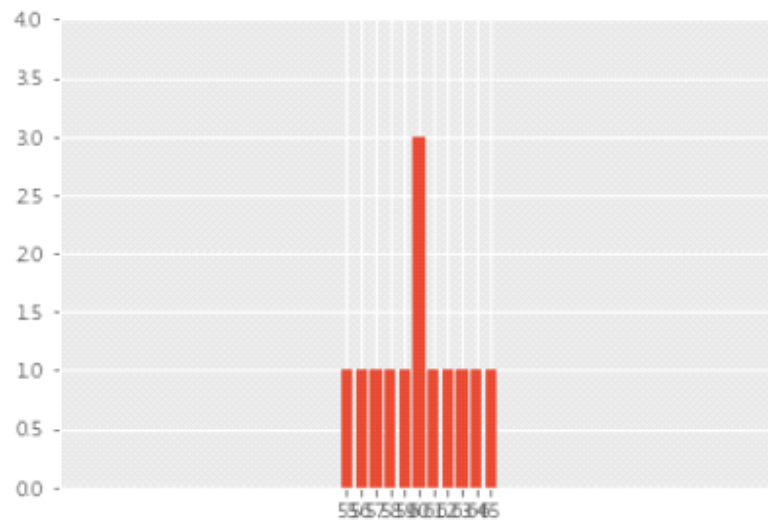
```
# plt.xlabel('나이')
# plt.ylabel('인원수(명)')
```

```
# plt.title('나이별 인원수')
```

```
plt.xlim( 35, 85)
```

```
plt.ylim( 0, 4)
```

```
plt.show()
```



답3)

```
from collections import Counter
```

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
num = [55,56,57,58,59,60,60,60,60,61,62,63,64,65]
```

```
cnt = Counter(num)
```

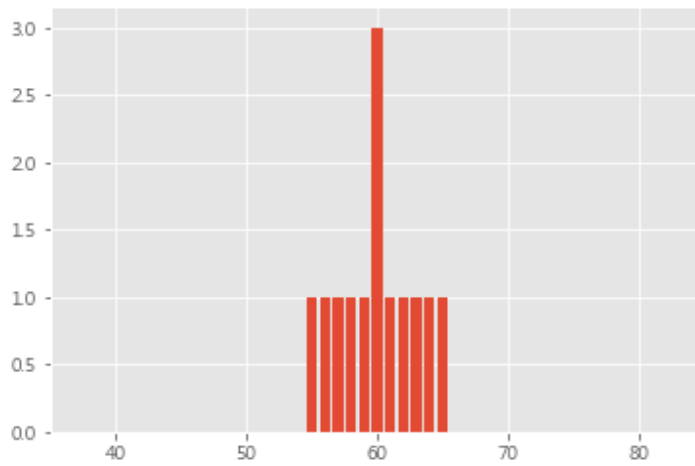
```
y = cnt.values()
```

```
x = cnt.keys()
```

```
plt.bar(x, y, align='center')
```

```
plt.xlim(35, 85)
```

```
plt.show()
```



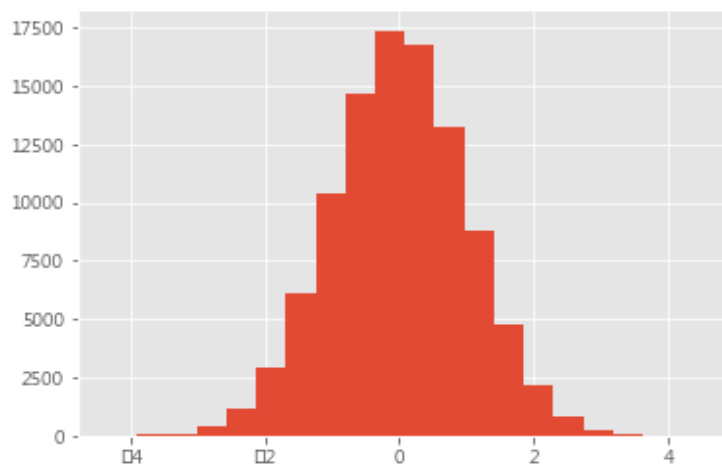
4. 히스토그램 그리기

```
import matplotlib.pyplot as plt
import numpy as np
N_points = 100000
n_bins = 20

# Generate a normal distribution, center at x=0 and y=5
x = np.random.randn(N_points)
y = .4 * x + np.random.randn(100000) + 5

fig, ax = plt.subplots()

# We can set the number of bins with the `bins` kwarg
ax.hist(x, bins=n_bins)
plt.show()
```



문제 20. 아래의 자료를 히스토그램 그래프로 그리시오
(보기)

55, 56, 57, 58, 59, 60, 60, 60, 61, 62, 63, 64, 65

답)

```
num = [55,56,57,58,59,60,60,60,61,62,63,64,65]
```

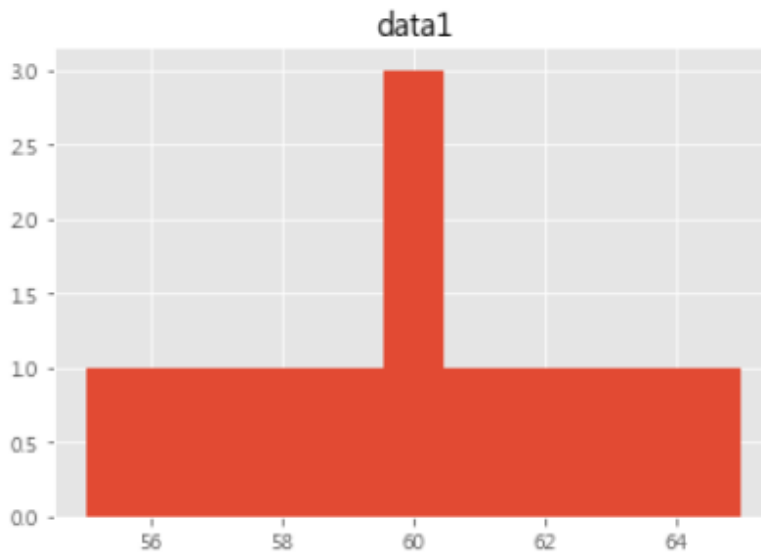
```
import matplotlib.pyplot as plt
```

```
plt.grid(True)
```

```
plt.hist(num, bins=11)
```

```
plt.title('data1')
```

```
plt.show()
```



5) plot 그래프

ex)

```
import numpy as np
```

```
import matplotlib.pyplot as plt
```

```
line = plt.figure()
```

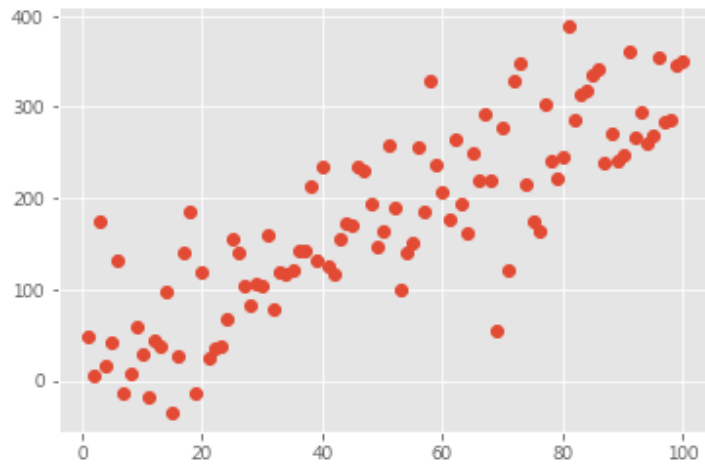
```
np.random.seed(5)
```

```
x = np.arange(1, 101)
```

```
y = 20 + 3 * x + np.random.normal(0, 60, 100)
```

```
plt.plot(x, y, "o")
```

```
plt.show()
```



6) 사분위수그래프

ex)

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
## Create data
```

```
np.random.seed(10)
```

```
collectn_1 = np.random.normal(100, 10, 200)
```

```
collectn_2 = np.random.normal(80, 30, 200)
```

```
collectn_3 = np.random.normal(90, 20, 200)
```

```
collectn_4 = np.random.normal(70, 25, 200)
```

```
## combine these different collections into a list
```

```
data = [collectn_1, collectn_2, collectn_3, collectn_4]
```

```
fig1, ax = plt.subplots(nrows=1,ncols=1, figsize=(8, 4))
```

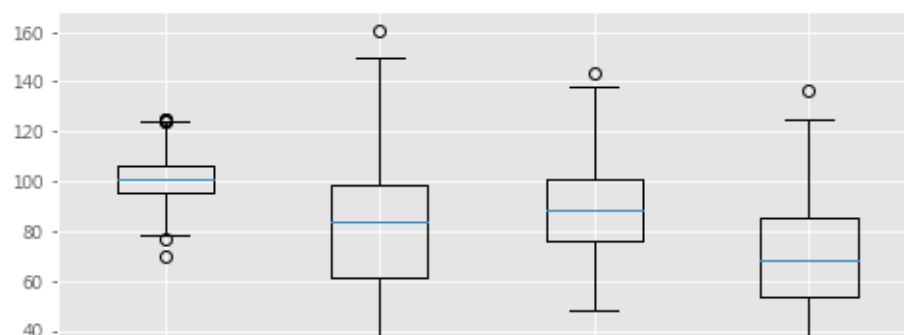
```
# plotting
```

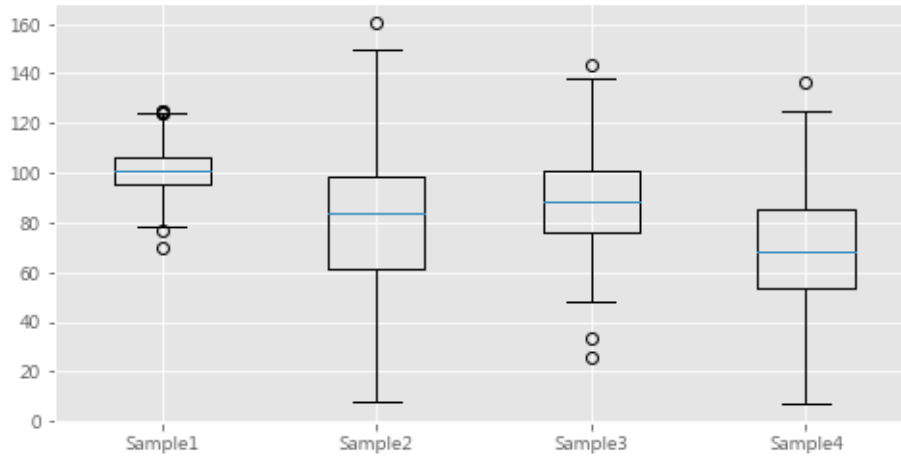
```
ax.boxplot(data)
```

```
## Custom x-axis labels
```

```
ax.set_xticklabels(['Sample1', 'Sample2', 'Sample3', 'Sample4'])
```

```
plt.show()
```





문제 21. 아래의 데이터에 1분위수, 2분위수, 3분위수가 어떻게 되는지 출력하시오
보기)

```
x = np.array([18, 5, 10, 23, 19, -8, 10, 0, 0, 5, 2, 15, 8,
              2, 5, 4, 15, -1, 4, -7, -24, 7, 9, -6, 23, -13,
              1, 0, 16, 15, 2, 4, -7, -18, -2, 2, 13, 13, -2,
              -2, -9, -13, -16, 20, -4, -3, -11, 8, -15, -1, -7, 4,
              -4, -10, 0, 5, 1, 4, -5, -2, -5, -2, -7, -16, 2,
              -3, -15, 5, -8, 1, 8, 2, 12, -11, 5, -5, -7, -4])
```

답)

```
import numpy as np
x = np.array([18, 5, 10, 23, 19, -8, 10, 0, 0, 5, 2, 15, 8,
              2, 5, 4, 15, -1, 4, -7, -24, 7, 9, -6, 23, -13,
              1, 0, 16, 15, 2, 4, -7, -18, -2, 2, 13, 13, -2,
              -2, -9, -13, -16, 20, -4, -3, -11, 8, -15, -1, -7, 4,
              -4, -10, 0, 5, 1, 4, -5, -2, -5, -2, -7, -16, 2,
              -3, -15, 5, -8, 1, 8, 2, 12, -11, 5, -5, -7, -4])

print(np.min(x)) # 최소값
print(np.percentile(x, 25)) # 1분위수
print(np.percentile(x, 50)) # 2분위수(중앙값)
print(np.percentile(x, 75)) # 3분위수
print(np.max(x)) # 최대값
```

```
-24
-5.75
0.5
5.0
23
```

문제 22. 위 21번 문제의 데이터를 4분위수 그래프로 그리시오

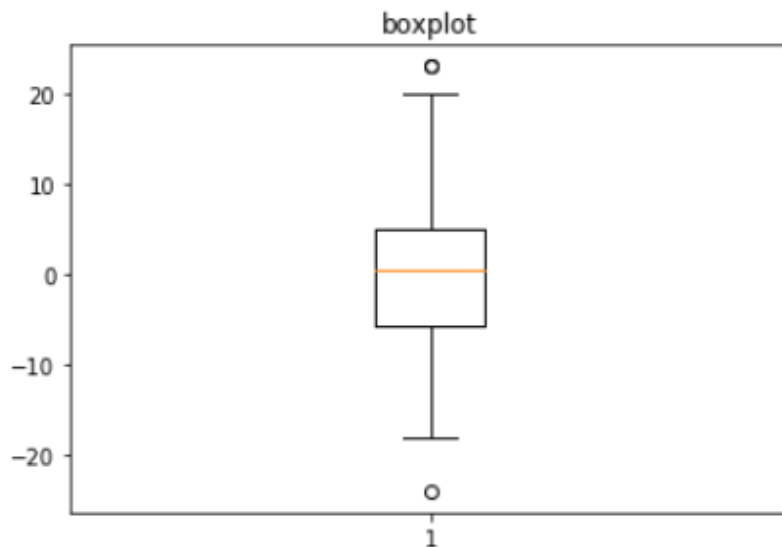
답)

```
import matplotlib.pyplot as plt
```



```
import numpy as np
x = np.array([18, 5, 10, 23, 19, -8, 10, 0, 0, 5, 2, 15, 8,
              2, 5, 4, 15, -1, 4, -7, -24, 7, 9, -6, 23, -13,
              1, 0, 16, 15, 2, 4, -7, -18, -2, 2, 13, 13, -2,
              -2, -9, -13, -16, 20, -4, -3, -11, 8, -15, -1, -7, 4,
              -4, -10, 0, 5, 1, 4, -5, -2, -5, -2, -7, -16, 2,
              -3, -15, 5, -8, 1, 8, 2, 12, -11, 5, -5, -7, -4])

plt.boxplot(x)
plt.title('boxplot')
plt.show()
```



문제 23. 사분위수의 범위가 무엇인가?

답)

범위 = 최대값 - 최소값

사분위수 범위 = 3사분위수 - 1사분위수

문제 24. 전체 범위를 구하는 함수를 파이썬으로 구현하시오

답)

```
def data_range(x):
    return max(x) - min(x)
```

문제 25. 사분위수 범위를 구하는 함수를 파이썬으로 구현하시오

답)

```
import numpy as np
def data_range(x):
    return np.percentile(x, 75) - np.percentile(x, 25)
```

문제26. 분산이란 무엇인가?

답) 평균을 중심으로 관찰값들이 얼마나 떨어져 분포하는가를 나타내는 것

문제27. 표준편차란 무엇인가?

답) 분산의 양의 제곱근으로 원시 데이터의 측정 단위가 같은 단위로 퍼짐정도를 나타내는 것.

문제28. 평균을 구하는 함수를 파이썬으로 구현하시오.(p.108의 18번문제)

보기)

환경 공학자가 납의 농도를 10군데의 강에서 측정하여 다음의 데이터를 얻었다.

```
-----  
0.062 0.071 0.067 0.068 0.066  
0.062 0.068 0.067 0.060 0.065  
-----
```

답)

```
x=[0.062, 0.071, 0.067, 0.068, 0.066, 0.062, 0.068, 0.067, 0.060, 0.065]
```

```
def mean(args):  
    return sum(args)/len(args)
```

```
print(mean(x))  
0.06559999999999999
```

문제29. 분산을 구하는 함수를 파이썬으로 구현하시오.

분산식)

답)

```
x=[0.062, 0.071, 0.067, 0.068, 0.066, 0.062, 0.068, 0.067, 0.060, 0.065]
```

```
def mean(args):  
    return sum(args)/len(args)
```

```
def variance(args):  
    total=0  
    m=mean(args)  
    for i in args:  
        total+=(i-m)**2  
  
    return total/len(args)
```

```
#print(mean(x))  
print(variance(x))  
1.0240000000000004e-05
```

문제30. 표준편차를 구하는 함수를 파이썬으로 구현하시오.

보기)

```
print(stddev(x))
```

답)

```
x=[0.062, 0.071, 0.067, 0.068, 0.066, 0.062, 0.068, 0.067, 0.060, 0.065]
```

```
def mean(args):
```

```
    return sum(args)/len(args)
```

```
def variance(args):
```

```
    total=0
```

```
    m=mean(args)
```

```
    for i in args:
```

```
        total+=(i-m)**2
```

```
    return total/len(args)
```

```
import math
```

```
def stddev(args):
```

```
    v = variance(args)
```

```
    return math.sqrt(v)
```

```
print(mean(x))
```

```
print(variance(x))
```

```
print(stddev(x))
```

```
0.06559999999999999
```

```
1.0240000000000004e-05
```

```
0.0032000000000000006
```

문제31. 아래의 코드를 이용해서 동전의 앞면과 뒷면이 나오는 상대적 도수 확률정의를 테스트 하시오. 1은 앞면, 2는 뒷면, loop문을 이용해서 10000번을 동전을 던지고 앞면이 몇번 나오고 뒷면이 몇번 나오는지 테스트 하시오.

보기)

```
import random
```

```
res=random.randrange(1,3)
```

```
print(res)
```

답)

```
import random
```

```

x=0
y=0
for i in range(10000):
    res=random.randrange(1,3)

    if res==1:
        x+=1
    else:
        y+=1

print('앞:',x,'뒤:',y)
    앞: 5094 뒤: 4906

```

3장. 확률의 정의

문제32. 한 회사원이 어느 도시에 출장을 갔는데 숙소 근처에 2개의 식당 (식당1,식당2) 가 있다. 어느 식당을 갈 것인지 망설이다가 주사위를 던져 홀수가 나오면 식당 1, 짝수가 나오면 식당 2 로 간다고 할때 식당 1이 뽑힐 확률을 코드로 작성하시오 !

답)

표본공간은 1,2,3,4,5,6 이고 홀수가 나올 사건은 1,3,5이므로 3개 그래서 $1/2$ 이다

```

import random
x=0
y=0
for i in range(10000):
    res=random.randrange(1,7)

    if res in (1,3,5):
        x+=1
    else:
        y+=1

print('홀:',x,'짝:',y)
    홀: 4983 짝: 5017

```

문제33. 네 사람 A,B,C,D 를 나란히 있는 네개의 의자에 배치시키려고 한다. 네 사람을 배치시키는 전체의 경우의 수와 그중 A가 가장 왼쪽에 배치될 경우의 수를 구하여라. A가 가장 왼쪽에 배치되는 확률은 얼마인가 ?

답)

전체 경우의수? 4!

A가 왼쪽에 배치될 경우의수 3!

A가 왼쪽에 배치될 경우의수 = $3! / 4! = 0.25$

문제34. 어느 회사에 경비원이 4명(A,B,C,D) 있다. 매일 아침 이 경비원 중 두 사람을 임의로 뽑아 둘 중 한 사람은 정문, 다른 한 사람은 후문의 경비로 배치한다. 4명을 정문과 후문에 배치하는 전체 경우의 수와 그중 A가 정문에 배치되는 경우의 수를 구하라. A가 정문에 배치될 확률은 ?

답)

```
from itertools import permutations
per=permutations(['빨','주','노','초'],2)
print(list(per)) #12
```

```
[('빨', '주'), ('빨', '노'), ('빨', '초'), ('주', '빨'), ('주', '노'), ('주', '초'), ('노', '빨'), ('노', '주'), ('노', '초'), ('초', '빨'), ('초', '주'), ('초', '노')]
```

이중 A를 정문에 배치 시키고 난 나머지 세사람중 한명을 후문에 배치 시키면 되므로

```
from itertools import permutations
per=permutations(['주','노','초'],1)
print(len(list(per)))
```

3

따라서, A가 정문에 배치될 확률은 ${}_3P_1 / {}_4P_2 = 1/4$

문제35. 파이썬으로 확률의 덧셈 법칙을 파이썬으로 시각화 하시오. (벤다이어그램을 파이썬)

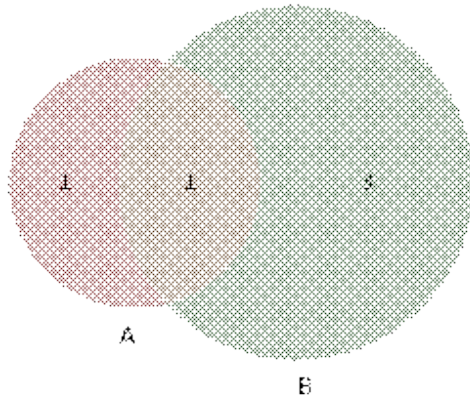
확률의 덧셈 법칙 : $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

답)

pip install matplotlib #아나콘다 프롬프트

```
from matplotlib import pyplot as plt
from matplotlib_venn import venn2, venn2_circles

v = venn2(subsets=[set([1,2]), set([2,3,4,5])], set_labels=('A','B'))
plt.title("ven diagram")
plt.show()
```



설명: 숫자 개수

문제36. 예제 3-5를 파이썬으로 구현 하시오.

보기)

이번 학기에 통계학과 2학년 학생 40명 중 교양선택으로 경제학을 수강 하는 학생이 25명, 경영학을 수강하는 학생이 30명, 두 과목을 모두 수강하는 학생이 20명이다. 통계학과 2학년 학생 한 사람을 만났을 때 이 학생이 경제학 또는 경영학(즉, 둘 중 한 과목이나 두 과목 모두)을 수강할 확률은 ?

확률의 덧셈 법칙 : $P(A \cup B) = P(A) + P(B) - P(A \cap B)$

$$25/40 + 30/40 - 20/40 = 0.875$$

답)

문제37. (조건부 확률) 예제 3-6을 파이썬으로 구현하시오.

3-6. 통계학과 2학년 학생 30명중 남학생이 10명, 여학생이 20명이다.

남학생중 1명, 여학생중 5명이 지방 출신이라 한다.

엑셀로 작성하고 판다스 데이터 프레임으로 구현하시오.

1) 한 학생을 뽑았더니 여학생이었다. 이 학생이 지방 출신일 확률은?

2) 한 학생을 뽑았더니 지방 출신이었다. 이 학생이 남학생일 확률은?

조건부 확률 공식 : $P(A|B) = P(A \cap B) / P(B)$

ex) 비가 온다는 조건하에 우산이 팔릴 확률

이 두사건은 서로 배반이면 안된다.

답)

1. 분할표를 먼저 그린다.

구분	서울	지방	계
남	9	1	10

여	15	5	20
계	24	6	30

```
import pandas as pd
boonhal=pd.read_csv("C:/python_data/aa.csv")
woman=boonhal['sum'][1]
etc=boonhal['etc'][2]
woman_ect=boonhal['etc'][1]
ect_man=boonhal['etc'][0]

a=woman_ect/woman
b=ect_man/etc
print(woman,etc,woman_ect,ect_man)
print('여학생중 지방:',a,'\n지방 중 남학생:',round(b,3))
```

문제38. 예제 3-7문제를 파이썬으로 구현하시오.

3-7) 한 학생을 뽑았을 때 남학생이고, 서울 출신일 확률?

확률의 곱셈법칙 : $P(M \cap S) = P(M)P(S|M)$

$$10/30 * 9/10 = 9/30$$

답)

문제39. 예제 3-8) 통계학과 2학년 학생30명의 남녀별, 출신지역별 분할표는 다음과 같다.

구분	서울 출신	지방 출신	계
남학생	5	5	10
여학생	10	10	20
계	15	15	30

한 학생을 뽑았을때 남학생일 사건과 서울 출신인 사건이 서로 독립인가?

$$P(M \cap S) = P(M) P(S|M)$$

↓

$P(M) P(S)$ 이렇게 나오면 서로 독립이다.

독립이다!!!

문제40. 예제 3-9를 파이썬으로 구현하시오.

6개의 제품이 들어 있는 상자가 있는데, 그중 2개가 불량품이라고 하자. 제품검사를 위해 3개를 추출했을 때 적어도 1개의 불량품이 발견될 확률은? 검사를 위해 한 번 추출한 제품은 다시 넣지 않는 비복원 추출 이라고 가정하자

$$h(x; N, n, k) = \frac{\binom{k}{x} \binom{N-k}{n-x}}{\binom{N}{n}}$$

$$x=1$$

$$N=6$$

n=3
k=2

$$\binom{k}{s} = {}_kP_s / r!$$

3개의 제품검사에서 1개의 불량품이 발견된 확률 = 3/5

3개의 제품검사에서 2개의 불량품이 발견된 확률 = 1/5

따라서, 3/5 + 1/5 = 4/5

1. 아래의 combination 함수를 파이썬으로 생성하시오.

일단 Permutation 함수를 만든다

```
6P3
—— --> Permutation
3!
print(permutation(6,3))
6x5x4=120

print(combination(6,3))
20
```

```
def permutation(a,b):
    cnt=1
    for i in range(b):
        cnt*=a
        a-=1
    print(cnt)
permutation(6,3)
120
```

답)

```
def combination(a,b):
    cnt=1
    for i in range(1,b+1):
        cnt*=i

    cnt2=1
    for j in range(0,b):
        cnt2*=a-j
    return cnt2/cnt
```

```
combination(6,3)
```


$$h(x; N, n, k) = \binom{k}{x} \binom{N-k}{n-x} / \binom{N}{n}$$

x=1

N=6

n=3

k=2

$$\binom{k}{s} = k! / (s! (k-s)!)$$

위의 공식을 파이썬으로 구현하시오.

```
def combination(a,b):
```

```
    cnt=1
```

```
    for i in range(1,b+1):
```

```
        cnt*=i
```

```
    cnt2=1
```

```
    for j in range(0,b):
```

```
        cnt2*=a-j
```

```
    return cnt2/cnt
```

```
def init_range(a,b,c,d):
```

```
    return combination(d,a)*combination(b-d,c-a)/combination(b,c)
```

```
print(init_range(1,6,3,2))
```

```
print(init_range(2,6,3,2))
```

• 이산형 확률 변수

동전 2개를 던졌을때 나올수 있는 표본 공간은 어떻게 되는가?

{ 앞앞, 앞뒤, 뒤앞, 뒤뒤 } 원소개 총4개

표본 공간의 각 원소가 나오는 사건의 확률? 1/4

확률 변수 X 값에 대한 확률

동전 2개를 던졌을때 앞이 나올 확률?

x	0	1	2
P(X=x)	1/4	1/2	1/4

동전2개를 던졌을때 앞이 나올 누적 확률?

x	0	1	2
P(X=x)	1/4	3/4	1

문제41. 한 동네에 200가구가 살고 있다. 각 가구가 지난 1년동안 병원에서 찾은 횟수를 조사 해보니 다음과 같다.

병원 방문횟수	0	1	2	3	4
가구수	74	80	30	10	6

확률 변수를 X ='병원 방문 횟수'로 하여 확률 분포 함수와 누적 확률 분포 함수를 아래와 같이 파이썬으로 구현하시오.

```
보기)
h_cnt=[74, 80, 30, 10, 6]
print(h_function(h_cnt))
[ 0.37, 0.40, 0.15, 0.05, 0.03]

print(h_cumulative_function(h_cnt))
[0.37, 0.77, 0.92, 0.97, 1.00]
```

```
답)
h_cnt=[74, 80, 30, 10, 6]
def h_function(h_list):
    res=[]
    for i in h_list:
        res.append(i/sum(h_list))
    return res

print(h_function(h_cnt))

def h_cumulative_function(h_list):
    res=[]
    sum=0
    for i in h_function(h_list):
        sum+=i
        res.append(round(sum,2))
    return res

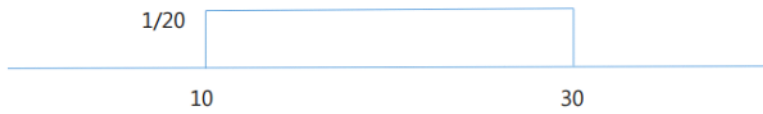
print(h_cumulative_function(h_cnt))
[0.37, 0.4, 0.15, 0.05, 0.03]
[0.37, 0.77, 0.92, 0.97, 1.0]
```

문제42. 예제3-11.

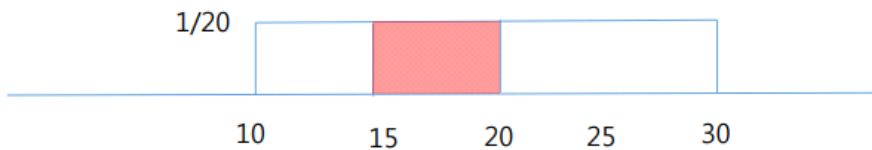
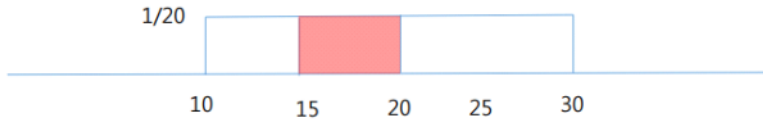
피자를 주문해서 집에 도착할 때까지 걸리는 시간이 10분에서 30분 사이에 어느 시간이나 같은 가능성을 갖는다. 확률변수 X ='피자배달에 걸리는 시간' 이라고 할 때, X 의 확률분포 함수를 구하고 그림을 그려라. 또 15분에서 20분 사이에 배달될 확률을 구하라 !

X 는 10에서 30사이의 어느 수나 같은 가능성을 가지므로 확률분포는

$$f(x) = 1/(30-10) , 10 < x < 30$$



15분에서 20분 사이에 배달될 확률은 그림으로 그리면



이 면적(확률)은 $(20-15) \times (1/20) = 0.25$

• 기댓값과 분산

기댓값? 사건이 벌어졌을때 이득과 그 사건이 벌어질 확률을 곱한것.

딸을 갖고 싶어하는 부모 --> 누군가에 물어봐서 앞으로 낳을 애기가 딸인지 아들인지 물어보았다.

그 누군가에게 1000원 주고 맞추면 1000원을 주는거고 못맞추면 1000원을 돌려받는다.

ex)

부부1	부부2
1000	1000
0.5	0.5
500원	500원

문제43. 어떤 자동차 판매영업소가 1주일동안 자동차 판매대수에 대한 다음과 같은 확률분포표를 얻었다고 하자.

확률변수

x_i	0	1	2	3	4	5	계
$f(x_i)$	0.1	0.1	0.2	0.3	0.2	0.1	1.0

확률밀도함수

확률 변수 X를 판매대수라 할 때, 기대 판매 대수 $E(X)$ 를 구하라.

보기)

car_sell_p = [0.1, 0.1, 0.2, 0.3, 0.2, 0.1]

```
print( expect_value(car_sell_p) )
```

답)

```
def expect_value(c_list):
```

```
    res=0
```

```
    for i in range(len(c_list)):
```

```
        res+=i*c_list[i]
```

```
    return res
```

```
car_sell_p = [ 0.1, 0.1, 0.2, 0.3, 0.2, 0.1 ]
```

```
print( expect_value(car_sell_p) )
```

문제44. 예제3-13)을 파이썬으로 구현하시오.

3-13)확률변수 X를 동전을 2회 던졌을 때 나오는 표면의 수로 정의할 때, $g(X)=(X-1)^2$ 의 기댓값을 구해 보자.

{ '앞앞', '앞뒤', '뒤앞', '뒤뒤' } 표본원소수 4

각 원소가 나올 확률은 1/4

동전이 앞이 나올 확률에 대한 확률 분포표

	앞앞	앞뒤	뒤앞	뒤뒤
동전 앞 표면수	0	1	1	2
확률	1/4	1/4	1/4	1/4

$$0 \times 1/4 + 1 \times 1/4 + 1 \times 1/4 + 2 \times 1/4 = 1$$

동전이 앞이 나올 확률에 대한 확률 분포표 $g(x) = (x-1)^2$

$$(0-1)^2 \times 1/4 + (1-1)^2 \times 1/4 + (1-1)^2 \times 1/4 + (2-1)^2 \times 1/4 = 0.25$$

4장. 확률분포와 표본분포 (총12문제)

확률분포의 종류?

1. 이산 확률 분포 : 이산확률 변수의 확률 분포이다.

ex) 이항분포, 초기하 분포, 포아송 분포

2. 연속 확률 분포 : 연속확률 변수의 확률분포이다.

ex) 정규 분포, t분포, 카이제곱 분포, F분포, 지수분포

이항분포에 앞서 먼저 베르누이 시행에 대해서 이해하자.

베르누이 시행 이란?

: 상호 배반적인 2가지 결과만을 기대할 수 있는 시행

ex) 동전을 던졌을 때 나올 수 있는 앞면 or 뒷면.

축구 경기에서 승자 or 패자.

대통령 선거에서 당선 or 낙선.

베르누이 시행결과로 얻을 수 있는 값을 나타내는 확률변수를 ?

"베르누이 확률변수라 한다"

- 베르누이 시행의 특징 3가지

1. 베르누이 시행의 결과를 나타내는 확률 변수는 1 or 0 이다.
2. 베르누이 시행에서 성공확률과 실패확률의 합은 항상 1 이다.
3. 각각의 베르누이 시행은 서로 독립적이기 때문에 여러 번에 걸친 베르누이 시행간에는 서로 영향을 미치지 않는다.

4-1. 이산 확률 분포

이항분포

"한번 이상 반복 실시한 베르누이 시행 결과의 합을 변수값으로 하는 확률변수의 분포를 이항(확률)분포라고 한다."

ex) 동전의 앞면이 나오면 성공, 뒷면이 나오면 실패로 하는 동전 던지기를 n번 하는 경우, 얻을 수 있는 결과값들의 합을 변수값으로 하는 확률변수의 분포를 이항분포라고 한다.

문제45. 어느 보험회사의 영업사원이 한 고객을 만나 그 사람을 보험에 가입하게 할 확률을 과거의 경험으로 보아 20%이다. 오늘 아침 영업사원이 10명의 고객을 만날 예정이다.

(1) 세 사람이 보험에 가입할 확률은?

$$P(X=x) = \binom{n}{x} p^x (1-p)^{n-x}$$

$$P(X=3) = \binom{10}{3} 0.2^3 (1-0.2)^{10-3}$$

0.23 이 아니라 $(0.2)^3$ 이다.

답)

```
def combination(a,b):  
    cnt=1  
    for i in range(1,b+1):  
        cnt*=i  
  
    cnt2=1  
    for j in range(0,b):  
        cnt2*=a-j  
    return cnt2/cnt
```

combination(6,3)

```
def bin_dist(n,x,p):
    res=combination(n,x) * (p**x)*((1-p)**(n-x))
    return res
```

```
print(round(bin_dist(10,3,0.2),4 ))
print('%0.4f' %bin_dist(10,3,0.2))
```

```
0.2013
0.2013
```

(2) 두 사람 이상이 보험에 가입할 확률은?

$$P(X=x) = \binom{n}{x} p^x (1-p)^{n-x}$$

$$\begin{aligned} P(X \geq 2) &= 1 - (P(X=0) + P(X=1)) \\ &= 1 - \left(\binom{10}{0} 0.2^0 (1-0.2)^{10} + \binom{10}{1} 0.2^1 (1-0.2)^9 \right) \\ &= 1 - (0.1074 + 0.2684) = 0.6242 \end{aligned}$$

1 - (아무도 가입하지 않을 확률 + 1명 가입할 확률)

답)

```
def combination(a,b):
    cnt=1
    for i in range(1,b+1):
        cnt*=i

    cnt2=1
    for j in range(0,b):
        cnt2*=a-j
    return cnt2/cnt
```

```
combination(6,3)
```

```
def bin_dist(n,x,p):
    res=combination(n,x) * (p**x)*((1-p)**(n-x))
    return res
```

```
# print(round(bin_dist(10,3,0.2),4 ))
```

```
# print('%0.4f' %bin_dist(10,3,0.2))
```

```
print(round( 1-(bin_dist(10,0,0.2) + bin_dist(10,1,0.2) ),4 ))
```

```
0.6242
```

(3) 평균 몇 사람이 가입하겠는가? 또 그 표준편차는?

$$E(X) = np$$

$$E(x) = 10 \times 0.2 = 2$$

$$\text{Var}(X) = np(1-p)$$

$$\text{Var}(x) = 10 \times 0.2 \times 0.8 = 1.6$$

$$\sqrt{1.6} = 1.265$$

var(x) : 분산

$\sqrt{\text{분산}}$: 표준 편차

답)

$$10 \times 0.2 = 2$$

문제46. 한 복권의 당첨률이 30%라고 한다. 복권을 8장을 구입했다고 하자.

(1) 4장이 당첨될 확률은?

x	0	1	2	3	4	5	6	7	8
P(X≤x)	0.058	0.255	0.552	0.806	0.942	0.989	0.999	1.00	1.00

답)

$$4\text{장 당첨될 확률} - 3\text{장 당첨될 확률} = 0.136$$

(2) 3장~7장이 당첨될 확률은?

x	0	1	2	3	4	5	6	7	8
P(X≤x)	0.058	0.255	0.552	0.806	0.942	0.989	0.999	1.00	1.00

답)

$$1 - 3\text{장당첨될 확률} = 0.448$$

(3) 6장 이상 당첨될 확률?

x	0	1	2	3	4	5	6	7	8
P(X≤x)	0.058	0.255	0.552	0.806	0.942	0.989	0.999	1.00	1.00

답)

$$1 - 5\text{장 당첨될 확률} = 0.011$$

초기하 분포

" 이항분포에서 베르누이 시행은 서로 독립적이어서 시행에서 성공할 확률은 항상 동일하다.
그러나 각 시행이 서로 독립적이지 않고, 유한 모집단을 대상으로 비복원 추출로 표본을 선정하는 경우, 즉 표본을 추출할 때 마다 표본을 추출하는 표본공간이 매번 변하는 경우의 확률 변수는 어떠한 분포를 하는가? "

이러한 확률 변수에 적용할 수 있는 분포가 바로 초기화 분포이다.

ex) 주머니 속에 노란공 6개 파란공 4개가 들어있다.

10개의 공 중에서 임의로 3개의 공을 꺼낼 경우 뽑힌 2개의 공이 파란색 공일 확률은?

이때 3개의 공을 뽑는 방법이 복원 추출인지, 비복원 추출인지에 따라 확률값이 다르다.

그래서 표본을 추출하는 시행결과가 서로 독립적인 **복원추출 방법**을 사용하는 경우에 확률 변수는 **이항분포**로 하고, 표본을 추출하는 시행 결과가 서로 영향을 미치는 유한 모집단을 대상으로 **비복원 추출방법을 사용하는 경우에 확률 변수는 "초기화 분포"로 한다.**

문제47. 흰공 3개와 검은 공 2개가 들어 있는 주머니에서 임의로 2개를 꺼냈을때 검은공의 개수인 확률 변수로 정의하자. $X=0,1,2$ 의 값을 취할 확률은?

답)

```
def combination(a,b):
    cnt=1
    for i in range(1,b+1):
        cnt*=i
```

```
    cnt2=1
    for j in range(0,b):
        cnt2*=a-j
    return cnt2/cnt
```

```
combination(6,3)
```

```
def init_distrib(w,b,c,bn):
    res=combination(b,bn)*combination(w,c-bn)/combination(w+b,c)
    return res
print(init_distrib(3,2,2,0))
print(init_distrib(3,2,2,1))
print(init_distrib(3,2,2,2))
0.3
0.6
0.1
```

포아송 분포

"일상생활에서는 일정한 시간이나 거리 혹은 면적 등에서 발생하는 특정한 사건의 수에 관심을 보이는 경우가 있다. "

ex) 새로 구입한 자동차에서 발견되는 흠집의 수, 하루동안 고장나는 기계의 대수, 일정 시간 내에는행창구에 방문하는 고객의 수, 1년간 아파트 엘리베이터가 고장난 횟수, 생명보험에 가입한 사람 중에서 단위시간당 사망하는 사람의 수

=> 이러한 경우 발생하는 특정한 사건의 횟수를 확률변수값으로 하는 분포가 바로 **포아송 분포!!**

- **포아송 분포의 기본적인 가정 3가지**

1. **독립성** : 일정한 단위 시간이나 단위 길이 혹은 단위 면적 등에서 발생하는 사건들은 셀 수 있을 정도의 수이며 서로 독립적이어야 한다.

ex) 일정 시간에 은행에 방문하는 고객수의 확률을 조사할때, 그 각각의 사건이 별개여야 한다.

2. **비집락성** : 극히 작은 단위시간과 단위 면적등에서는 한번에 둘 이상의 사건이 동시에 발생할 확률은 매우 작기때문에, 이러한 사건이 발생할 확률은 0으로 간주한다.

ex) 10분동안 은행을 방문하는 고객수가 10명이라고 할때 평균적으로 1분에 1명의 고객이 은행에 방문한 셈이 된다. 그러나 1초동안에 은행을 방문한 고객의 수에 대한 확률은 0이라 할 수 있다.

3. **비례성** : 일정한 단위시간이나 단위 면적에서 발생하는 사건의 수는 단위시간이나 단위 면적의 크기에 비례한다.

ex) 자동차 사고가 1일 평균2건이 발생한다면, 1주일이면 14건이 된다.

포아송분포 공식

$$P(X=x) = \frac{e^{-m} * m^x}{x!}$$

설명 : e는 exponential 의 약자로서 자연로그의 밑이 되는 지수를 의미한다. 그값은 약 2.7이 된다. m은 평균 출현 횟수

문제48.

문제4 “포아송 분포”

화제보험회사에서 1,860명의 보험 계약자를 보유하고 있는데, 1년에 화제가 발생하여 보험금을 청구하는 확률이 1/600 이다. 어느 해에 보험계약자 중 화제보험금을 청구하는 횟수를 확률변수 X로 하여 0,1,2,3 인 경우의 확률을 구하여라

답)

```
import numpy as np
import math
```

```
for i in range(4):
    print(i)
    a = (np.exp(-3.1) * 3.1 ** i)/math.factorial(i)
    print(round(a,4))
```

```
0
0.045
1
0.1397
2
0.2165
3
0.2237
```

문제49. 아래와 같이 포아송 분포 함수를 만들어서 문제 48번을 해결하시오.

보기)

```
poisson_distrib(0,3.1) -> 0.045
poisson_distrib(1,3.1) -> 0.1397
poisson_distrib(2,3.1) -> 0.2165
poisson_distrib(3,3.1) -> 0.2237
```

답)

```
def poisson_distrib(x,e):
    import numpy as np
    import math
    a=(np.exp(-e)*e**x)/math.factorial(x)
    print(round(a,4))
```

```
poisson_distrib(0,3.1)
poisson_distrib(1,3.1)
poisson_distrib(2,3.1)
poisson_distrib(3,3.1)
```

```
0.045
0.1397
0.2165
0.2237
```

문제50.

은행에서 하루 평균 6건의 불량수표를 받게 된다고 할 때, 어떤 특정한 날에 불량수표를 4번 받을 확률은?

답)

평균 출현 횟수(m) = 6

```
def poisson_distrib(x,e):
    a=(np.exp(-e)*e**x)/math.factorial(x)
    print(round(a,4))
```

```
poisson_distrib(4,6)
0.1339
```

4-2 연속 확률 분포

정규분포

"정규분포는 연속 확률분포중에서 가장 대표적인 분포로써 통계학과 조사방법론에서 가장 많이 사용되는 확률분포이다."

예: 사람들의 몸무게와 키, 지능지수등과 같은 대부분의 변수들은 정규분포를 한다고 할 수 있다.

이러한 정규분포의 가장 큰 특징은 좌우 대칭인 **종 모양**을 이루고 있다.

사람들의 몸무게를 모집단으로 해서 추출한 표본의 평균은 **관측치가 많아질수록 정규분포에 가까워진다.**

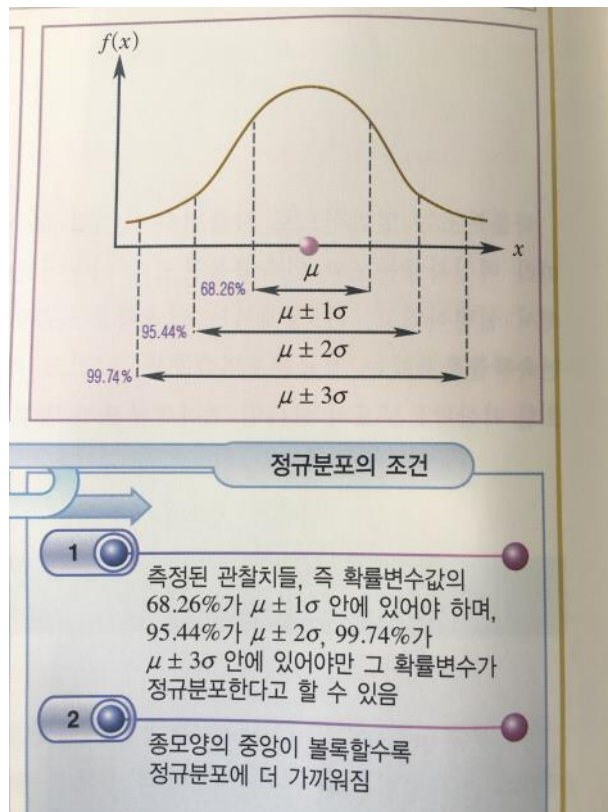
• 정규분포의 특징 3가지

1. 정규분포는 평균을 중심으로 좌우 대칭 종모양을 하고 있다.

2. 정규분포는 평균을 중심으로 좌우 1개의 표준편차 안에 측정된 관측치, 즉 확률변수값들의 68.26%가 포함되어 있어야 한다.

좌우 2개의 표준편차 안에는 95.44%의 확률변수값들이 있어야한다.

좌우 3개의 표준편차 안에는 99.74%에 이르는 대부분의 값들이 있다.



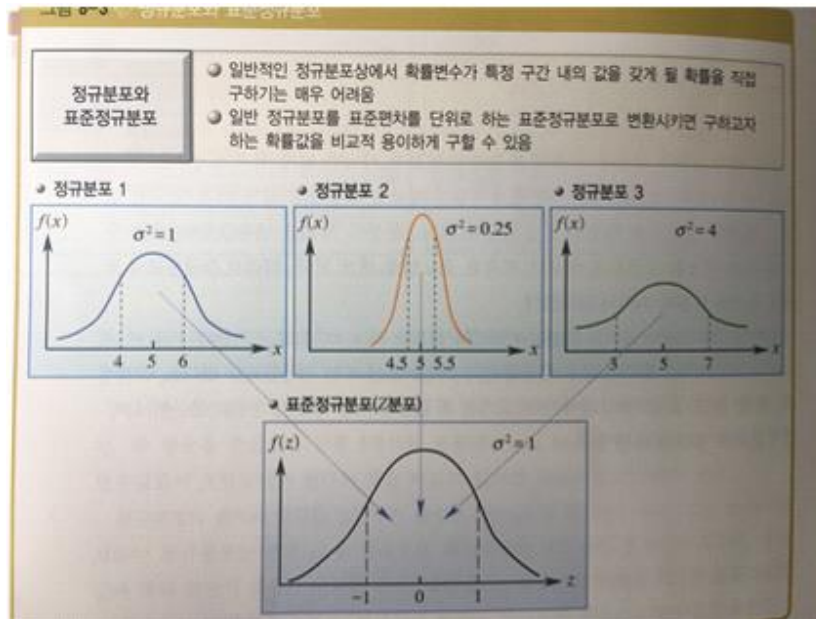
3. 정규분포의 위치와 모양을 결정하는 모수는 평균과 분산이다.

• 표준 정규 분포

"정규분포는 평균과 분산의 크기에 따라서 여러가지 형태를 갖는다. 따라서, 서로 다른 두개의 분포를 비교하거나 확률분포상의 면적의 크기를 계산하여 확률을 알아내는 것이 쉽지 않다.

예를들어, 평균이 5이고 분산이 16인 정규분포에서 $P(2 < x < 8)$ 의 확률을 계산하는 것은 상당히 어렵다. (적분을 해서 근사치를 구해야한다.) <-어렵다


그래서!! 평균과 분산에 따라 변하는 정규분포의 모양과 측정단위에 무관하게 어떠한 정규분포이든 특정 구간 사이의 확률값을 구할 수 있는 방법이 필요하다.



표준편차를 변수값의 단위로 하여 만들어진 정규분포를 표준 정규분포라고 한다. 이러한 표준정규 분포는 평균이 0이고, 분산이 1인 정규분포로써 표준편차를 변수값의 단위로 사용하기 때문에 모든 정규분포는 표준 정규 분포로 전환이 가능하다.

정규분포상의 구간을 표준 정규분포상의 구간으로 바꾼다음 제시된 표를 찾아 봄으로써 해당 구간의 확률값을 손쉽게 구할 수 있다.

표준정규분포표

$$P[Z \leq z] = \int_{-\infty}^z \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$$


z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
-3.5	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0002
-3.4	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003	0.0003
-3.3	0.0005	0.0005	0.0005	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0003
-3.2	0.0007	0.0007	0.0006	0.0006	0.0006	0.0006	0.0006	0.0005	0.0005	0.0005
-3.1	0.0010	0.0009	0.0009	0.0008	0.0008	0.0008	0.0008	0.0008	0.0007	0.0007
-3.0	0.0013	0.0013	0.0013	0.0012	0.0012	0.0011	0.0011	0.0011	0.0010	0.0010
-2.9	0.0019	0.0018	0.0018	0.0017	0.0016	0.0016	0.0015	0.0015	0.0014	0.0014
-2.8	0.0026	0.0025	0.0024	0.0023	0.0023	0.0022	0.0021	0.0021	0.0020	0.0020
-2.7	0.0035	0.0034	0.0033	0.0032	0.0031	0.0030	0.0029	0.0028	0.0027	0.0026
-2.6	0.0047	0.0045	0.0045	0.0043	0.0041	0.0040	0.0039	0.0038	0.0037	0.0036
-2.5	0.0062	0.0060	0.0059	0.0057	0.0055	0.0054	0.0052	0.0051	0.0049	0.0048
-2.4	0.0082	0.0080	0.0078	0.0075	0.0073	0.0071	0.0069	0.0068	0.0066	0.0064
-2.3	0.0107	0.0104	0.0102	0.0099	0.0096	0.0094	0.0091	0.0089	0.0087	0.0084
-2.2	0.0139	0.0136	0.0132	0.0129	0.0125	0.0122	0.0119	0.0116	0.0113	0.0110
-2.1	0.0179	0.0174	0.0170	0.0166	0.0162	0.0158	0.0154	0.0150	0.0145	0.0143
-2.0	0.0228	0.0222	0.0217	0.0212	0.0207	0.0202	0.0197	0.0192	0.0188	0.0183
-1.9	0.0287	0.0281	0.0274	0.0268	0.0262	0.0256	0.0250	0.0244	0.0239	0.0233
-1.8	0.0359	0.0351	0.0344	0.0336	0.0329	0.0322	0.0314	0.0307	0.0301	0.0294
-1.7	0.0446	0.0436	0.0427	0.0418	0.0409	0.0401	0.0392	0.0384	0.0375	0.0367
-1.6	0.0548	0.0537	0.0526	0.0516	0.0505	0.0495	0.0485	0.0475	0.0465	0.0455
-1.5	0.0668	0.0655	0.0643	0.0630	0.0618	0.0606	0.0594	0.0582	0.0571	0.0559
-1.4	0.0808	0.0793	0.0778	0.0764	0.0749	0.0735	0.0721	0.0708	0.0694	0.0681
-1.3	0.0968	0.0951	0.0934	0.0918	0.0901	0.0885	0.0869	0.0853	0.0836	0.0823
-1.2	0.1151	0.1131	0.1112	0.1093	0.1075	0.1056	0.1038	0.1020	0.1003	0.0985
-1.1	0.1357	0.1335	0.1314	0.1292	0.1271	0.1251	0.1230	0.1210	0.1190	0.1170
-1.0	0.1587	0.1562	0.1539	0.1515	0.1492	0.1469	0.1446	0.1423	0.1401	0.1379
-0.9	0.1841	0.1814	0.1788	0.1762	0.1736	0.1711	0.1685	0.1660	0.1635	0.1611
-0.8	0.2119	0.2090	0.2061	0.2033	0.2005	0.1977	0.1949	0.1922	0.1894	0.1867
-0.7	0.2420	0.2389	0.2358	0.2327	0.2297	0.2266	0.2236	0.2206	0.2177	0.2148
-0.6	0.2743	0.2709	0.2676	0.2643	0.2611	0.2578	0.2546	0.2514	0.2483	0.2451
-0.5	0.3085	0.3050	0.3015	0.2981	0.2946	0.2912	0.2877	0.2843	0.2810	0.2776
-0.4	0.3446	0.3409	0.3372	0.3336	0.3300	0.3264	0.3228	0.3192	0.3156	0.3121
-0.3	0.3821	0.3783	0.3745	0.3707	0.3669	0.3632	0.3594	0.3557	0.3520	0.3483
-0.2	0.4207	0.4168	0.4129	0.4090	0.4052	0.4013	0.3974	0.3936	0.3897	0.3859
-0.1	0.4602	0.4562	0.4522	0.4483	0.4443	0.4404	0.4364	0.4325	0.4286	0.4247
-0.0	0.5000	0.4960	0.4920	0.4880	0.4840	0.4801	0.4761	0.4721	0.4681	0.4641

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7703	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9733	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.6	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
2.9	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.0	0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990
3.1	0.9990	0.9991	0.9991	0.9991	0.9992	0.9992	0.9992	0.9992	0.9993	0.9993
3.2	0.9993	0.9993	0.9994	0.9994	0.9994	0.9994	0.9994	0.9995	0.9995	0.9995
3.3	0.9995	0.9995	0.9995	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9997
3.4	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9998
3.5	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998

문제51. Z가 표준정규확률변수일 때 다음의 확률을 구하여라.

보기)

$$1. P(Z < 1.96) \text{ -----} >$$

$$2. P(-1.96 < Z < 1.96) \text{ -----} >$$

$$3. P(Z > 1.96) \text{ -----} >$$

답)

$$1. 0.975$$

$$2. P(Z < 1.96) - P(Z < -1.96) : 0.975 - 0.025 = 0.95$$

$$3. 1 - P(Z < 1.96) = 1 - 0.975 = 0.025$$

문제52. (점심시간) 확률변수 X가 평균이 70, 표준편차가 10인 정규분포를 따를 때 다음의 확률은?

보기)

$$1. P(X < 94.3)$$

$$2. P(X > 57.7)$$

$$3. P(57.7 < X < 94.3)$$

$$Z = \frac{X - \mu}{\sigma}$$

답)

$$P(a < X < b) = P\left(\frac{a - \mu}{\sigma} < Z < \frac{b - \mu}{\sigma}\right)$$

$$1. P(X < 94.3) = P(24.3/10) = P(Z < 2.43) = 0.9925$$

$$2. P(X > 57.7) = P(57.7 - 70/10) = P(Z > -1.23) = 1 - 0.1093 = 0.8907$$

$$3. P(57.7 < X < 94.3) = P(-1.23 < Z < 2.43) : 0.8907 - (1 - 0.9925) = 0.8832$$

문제53. 한 회사의 건전지 수명은 평균 110시간, 표준편차가 10인 정규분포를 따른다고 하자.

1. 건전지 수명이 90시간 이하일 확률은?

2. 건전지 수명이 100시간 이상, 115시간 이하일 확률은?

3. 건전지 수명이 120시간 이상일 확률은?

답)

$$P(a < X < b) = P\left(\frac{a-\mu}{\sigma} < Z < \frac{b-\mu}{\sigma}\right)$$

(1) 건전지 수명이 90시간 이하일 확률은?

$$P(X < 90) = P\left(\frac{X-110}{10} < \frac{90-110}{10}\right) = P(Z < -2) = 0.0228$$

(2) 건전지 수명이 100시간 이상, 115시간 이하일 확률은?

$$\begin{aligned} P(100 < X < 115) &= P\left(\frac{100-110}{10} < \frac{X-110}{10} < \frac{115-110}{10}\right) \\ &= P(-1 < Z < 0.5) = 0.6915 - 0.1587 \\ &= 0.5328 \end{aligned}$$

$$P(a < X < b) = P\left(\frac{a-\mu}{\sigma} < Z < \frac{b-\mu}{\sigma}\right)$$

(3) 건전지 수명이 120시간 이하일 확률은?

$$\begin{aligned} P(X \geq 120) &= P\left(\frac{X-110}{10} \geq \frac{120-110}{10}\right) = P(Z \geq 1) \\ &= 1 - P(Z < 1) = 1 - 0.8413 \\ &= 0.1587 \end{aligned}$$

표본분포

"표본으로 부터 계산되는 함수식을 **통계량**이라고 하고, 통계량은 다른 표본이 추출되면 다른 값을 갖게되는 확률변수로 확률분포를 갖게 되는데, 통계량의 분포를 표본분포라고 한다."

표본분포는 표본으로부터 계산되는 통계량과 모수 사이의 관계를 규명해주기 때문에 모수의 추정과 검정을 가능하게 한다.

문제54. 한 회사의 영업사원 5명을 모집단이라고 하자. 관심 있는 화궐≡변수는 이 회사에서의 근무 연수로 다음과 같다.

보기)

6, 2, 4, 8, 10

1) 이모집단의 평균과 분산을 구하라.

평균 = 6

$$\text{분산} = \frac{(6-\bar{x})^2 + (2-\bar{x})^2 + (4-\bar{x})^2 + (8-\bar{x})^2 + (10-\bar{x})^2}{5} = 8$$

2) 이 모집단에서 표본의 크기가 2인 모든 가능한 표본을 단순 확률 복원추출하여 각각의 표본 평균을 구하라, 또 이러한 표본평균 전체의 평균과 분산을 구하고 모집단의 평균, 분산과 비교하라.

표본의 평균?

모집단에서 추출 가능한 $n = 2$ 인 모든 표본평균의 합
전체 표본의 개수(5X5)

표본	평균	표본	평균
2,2	2	4,2	3
2,4	3	4,4	4
2,6	4	4,6	5
2,8	5	4,8	6
2,10	6	4,10	7

표본의 분산?

문제55. 위의 표본의 평균을 구하는 방법을 파이썬으로 for loop문으로 구현해서 구현하시오.

보기)

표 4-3 모집단에서 추출 가능한 $n=2$ 인 모든 표본과 표본평균

표본	\bar{x}	표본	\bar{x}	표본	\bar{x}	표본	\bar{x}	표본	\bar{x}
2, 2	2	4, 2	3	6, 2	4	8, 2	5	10, 2	6
2, 4	3	4, 4	4	6, 4	5	8, 4	6	10, 4	7
2, 6	4	4, 6	5	6, 6	6	8, 6	7	10, 6	8
2, 8	5	4, 8	6	6, 8	7	8, 8	8	10, 8	9
2, 10	6	4, 10	7	6, 10	8	8, 10	9	10, 10	10

답)

```
u=[2,4,6,8,10]
res1=0
res2=0
for i in u:
    for j in u:
        res1=(i+j)/2
        res2+=res1

print(res2/25)
6.0
```

문제56. 아래표의 표본의 분산을 파이썬으로 구하시오.

보기)

표 4-3 모집단에서 추출 가능한 $n=2$ 인 모든 표본과 표본평균

표본	\bar{x}	표본	\bar{x}	표본	\bar{x}	표본	\bar{x}	표본	\bar{x}
2, 2	2	4, 2	3	6, 2	4	8, 2	5	10, 2	6
2, 4	3	4, 4	4	6, 4	5	8, 4	6	10, 4	7
2, 6	4	4, 6	5	6, 6	6	8, 6	7	10, 6	8
2, 8	5	4, 8	6	6, 8	7	8, 8	8	10, 8	9
2, 10	6	4, 10	7	6, 10	8	8, 10	9	10, 10	10

$$\text{분산} = \frac{(\text{실제값} - \text{평균})^2}{\text{전체 개수}} = \frac{\sum (\text{표본의 평균} - \text{표본 평균의 평균값})^2}{\text{전체 표본의 개수}}$$

답)

```
u=[2,4,6,8,10]
```

```
res1=0
```

```
res2=0
```

```
res3=0
```

```
for i in u:
```

```
    for j in u:
```

```
        res1=(i+j)/2
```

```
        res2+=res1
```

```
a=res2/25
```

```
for i in u:
```

```
    for j in u:
```

```
        res1=(i+j)/2
```

```
        res3+=(res1-a)**2
```

```
print(res2/25)
```

```
print(res3/25)
```

문제57. 표본평균의 막대 그래프를 그리기 위해 표본 평균의 도수 분포표를 아래와 같이 파이썬으로 구현하시오.

보기)

표본평균	도수
2	1
3	2
4	3
5	4
6	5
7	4
8	3
9	2
10	1

답)

```
u=[2,4,6,8,10]
```

```
res1=0
```

```

res2=0
res3=0
aa=[]
e=[]
f=[]

for i in u:
    for j in u:
        res1=(i+j)//2 #표본의평균
        aa.append(res1)
        res2+=res1
for i in set(aa):
    f.append(aa.count(i))
    e.append(i)
    print(i, aa.count(i))

print(f, e)

```

문제58. 위의 결과를 가지고 막대 그래프를 그리시오.

답)

```

import matplotlib.pyplot as plt

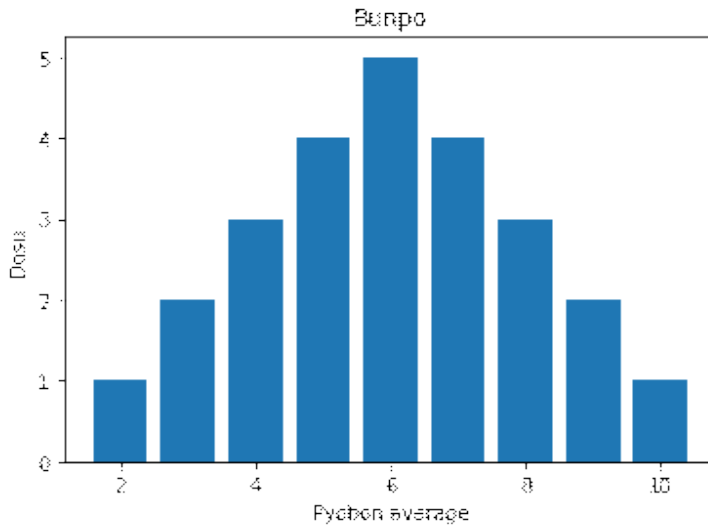
u=[2,4,6,8,10]
res1=0
res2=0
res3=0
aa=[]
e=[]
f=[]

for i in u:
    for j in u:
        res1=(i+j)//2 #표본의평균
        aa.append(res1)
        res2+=res1
for i in set(aa):
    f.append(aa.count(i))
    e.append(i)
    print(i, aa.count(i))

plt.bar(e,f) #x축,y축

plt.xlabel('Pyobon average')
plt.ylabel('Dosu')
plt.title('Bunpo')
plt.show()

```



이항 분포

"한번 이상 반복 실시한 베르누이 시행 결과의 합을 변수값으로 하는 확률변수의 분포를 이항(확률)분포라고 한다."

이항 분포의 평균? $n \cdot p$ (n:시행횟수, p:확률)

이항 분포의 분산? $n \cdot p(1-p)$

• 이항 분포의 정규 근사

$$Z = \frac{x - \text{평균}}{\text{표준편차}} = \frac{x - np}{\sqrt{np(1-p)}}$$

문제58. 한 공장에서 생산되는 제품의 불량품이 5%라고 한다. 어느 날 제품 100개를 표본 추출했을 때,

1) 불량품이 2개 이하일 확률은? $P(x \leq 2)$

$$\begin{aligned} P(x \leq 2) &= P\left(\frac{x - np}{\sqrt{np(1-p)}} \leq \frac{2 - np}{\sqrt{np(1-p)}}\right) = P\left(Z \leq \frac{2 - 5}{\sqrt{(100 \cdot 0.05) \cdot (1 - 0.05)}}\right) \\ &= P\left(Z \leq \frac{-3}{\sqrt{4.75}}\right) = P(Z \leq -1.376) = 0.0845 \end{aligned}$$

2) 3개에서 7개일 확률은?

$$\begin{aligned} P\left(\frac{3 - 5}{\sqrt{4.75}} \leq Z \leq \frac{7 - 5}{\sqrt{4.75}}\right) &= P(-0.918 \leq Z \leq 0.918) \\ &= P(Z \leq 0.918) - P(Z \leq -0.918) \end{aligned}$$

문제59. 문제54번에서 사용한 영업사원 5명의 근무연수 모집단을 다시 생각해보자.

보기)

6, 2, 4, 8, 10

1) 이 모집단에서 표본크기가 2인 모든 가능한 표본을 단순확률 복원추출하여 각각의 표본분산을 구하라. 또, 표본분산 전체의 평균과 분산을 구하고 모집단의 분산을 비교하라.

분산 = (표본-평균)²

표본 $s^2(\text{분산})$

2, 2 $(2-2)^2 + (2-2)^2 = 0$

2, 4 $(2-3)^2 + (4-3)^2 = 2$

2, 6 $(2-4)^2 + (6-4)^2 = 8$
 2, 8 $(2-5)^2 + (8-5)^2 = 18$
 2, 10 $(2-6)^2 + (10-6)^2 = 32$

표본의 분산의 평균값 : $0+2+8+18+32+ \dots / 25 = 8$

표본의 분산의 분산값 : $(0-8)^2+(2-8)^2+(8-8)^2 \dots / 25 = 2)$

문제60. 표본의 분산의 분산값을 파이썬으로 구현하시오.

패키지를 이용하자.....

답)

```
import itertools
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

`l=[6,2,4,8,10]`

`perm=list(itertools.product(l,repeat=2))` #표 생성, 설명은 아래

표 4-3 모집단에서 추출 가능한 $n=2$ 인 모든 표본과 표본평균

표본	\bar{x}	표본	\bar{x}	표본	\bar{x}	표본	\bar{x}	표본	\bar{x}
2, 2	2	4, 2	3	6, 2	4	8, 2	5	10, 2	6
2, 4	3	4, 4	4	6, 4	5	8, 4	6	10, 4	7
2, 6	4	4, 6	5	6, 6	6	8, 6	7	10, 6	8
2, 8	5	4, 8	6	6, 8	7	8, 8	8	10, 8	9
2, 10	6	4, 10	7	6, 10	8	8, 10	9	10, 10	10

```
print(perm)
s=[]
for i in perm:
    s.append((i[0]-np.mean(i))**2+(i[1]-np.mean(i))**2)
```

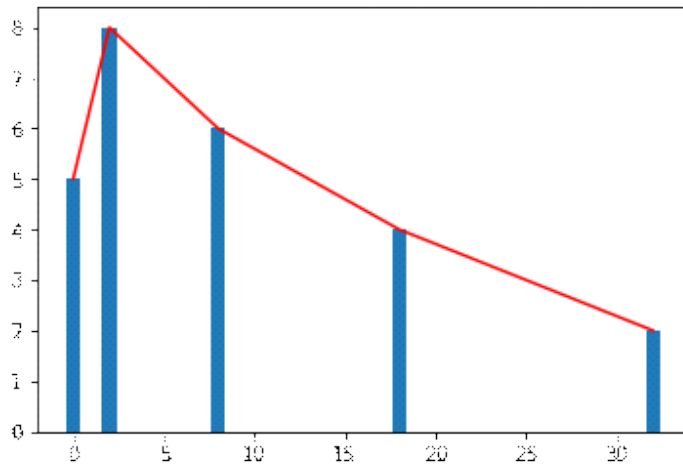
`print(np.mean(s))` #평균

`print(np.var(s))` #분산

####판다스 ★

```
lst=pd.Series(s)
lst=lst.value_counts().sort_index().reset_index()
plt.bar(lst['index'],lst[0],
```

```
tick_label=lst['index'],align='center')
line, = plt.plot(lst['index'],lst[0], color = 'red')
plt.show()
```



설명:

Combinatoric generators:

Iterator	Arguments	Results
<code>product()</code>	<code>p, q, ... [repeat=1]</code>	cartesian product, equivalent to a nested for-loop
<code>permutations()</code>	<code>p[, r]</code>	r-length tuples, all possible orderings, no repeated elements
<code>combinations()</code>	<code>p, r</code>	r-length tuples, in sorted order, no repeated elements
<code>combinations_with_replacement()</code>	<code>p, r</code>	r-length tuples, in sorted order, with repeated elements
<code>product('ABCD', repeat=2)</code>		AA AB AC AD BA BB BC BD CA CB CC CD DA DB DC DD
<code>permutations('ABCD', 2)</code>		AB AC AD BA BC BD CA CB CD DA DB DC
<code>combinations('ABCD', 2)</code>		AB AC AD BC BD CD
<code>combinations_with_replacement('ABCD', 2)</code>		AA AB AC AD BB BC BD CC CD DD

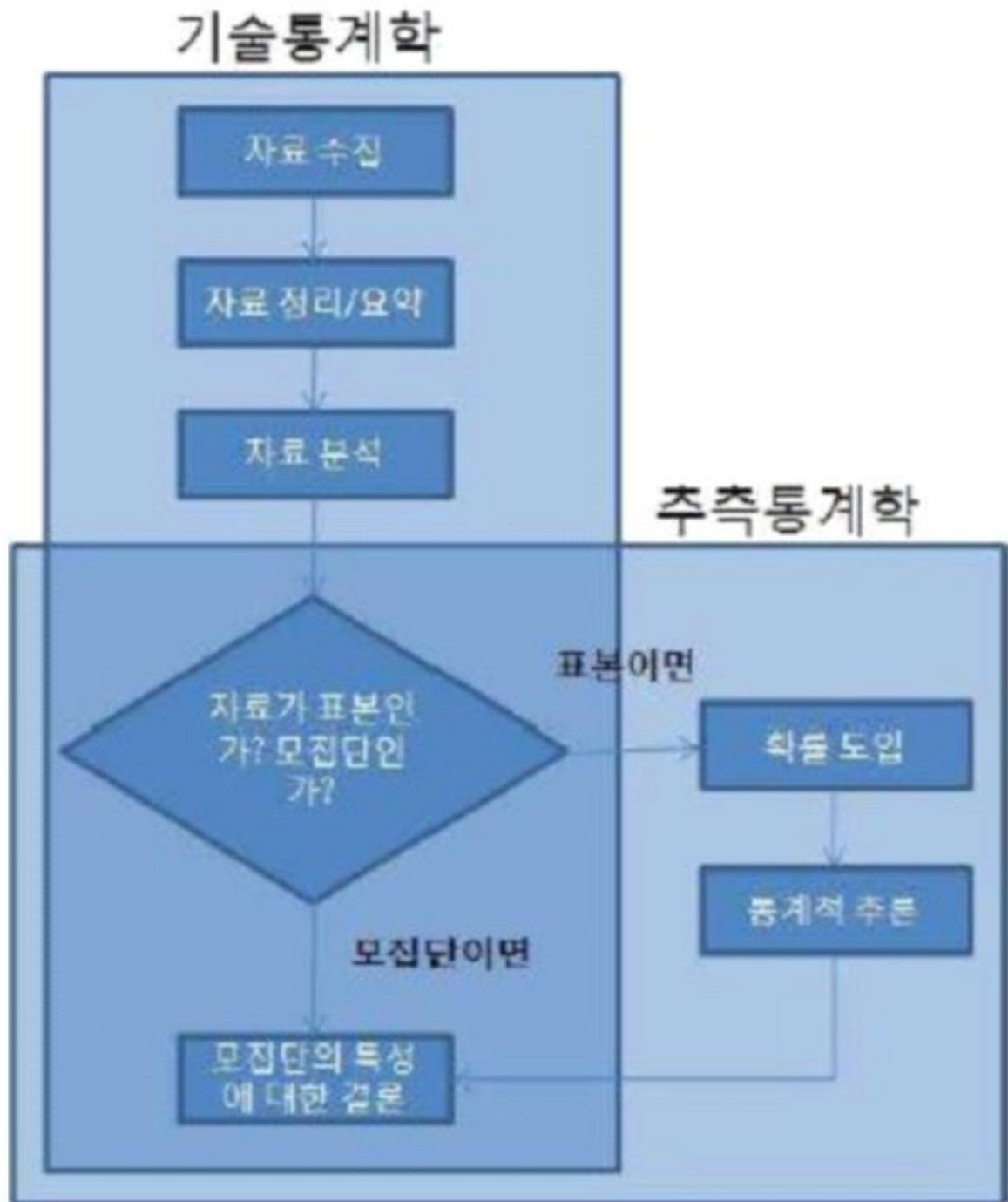
아래와 같은의미

```
def product(*args, **kwargs):
    # product('ABCD', 'xy') --> Ax Ay Bx By Cx Cy Dx Dy
    # product(range(2), repeat=3) --> 000 001 010 011 100 101 110 111
    pools = map(tuple, args) * kwargs.get('repeat', 1)
    result = [[]]
    for pool in pools:
        result = [x+[y] for x in result for y in pool]
    for prod in result:
        yield tuple(prod)
```

5장. 통계적 추정의 이해

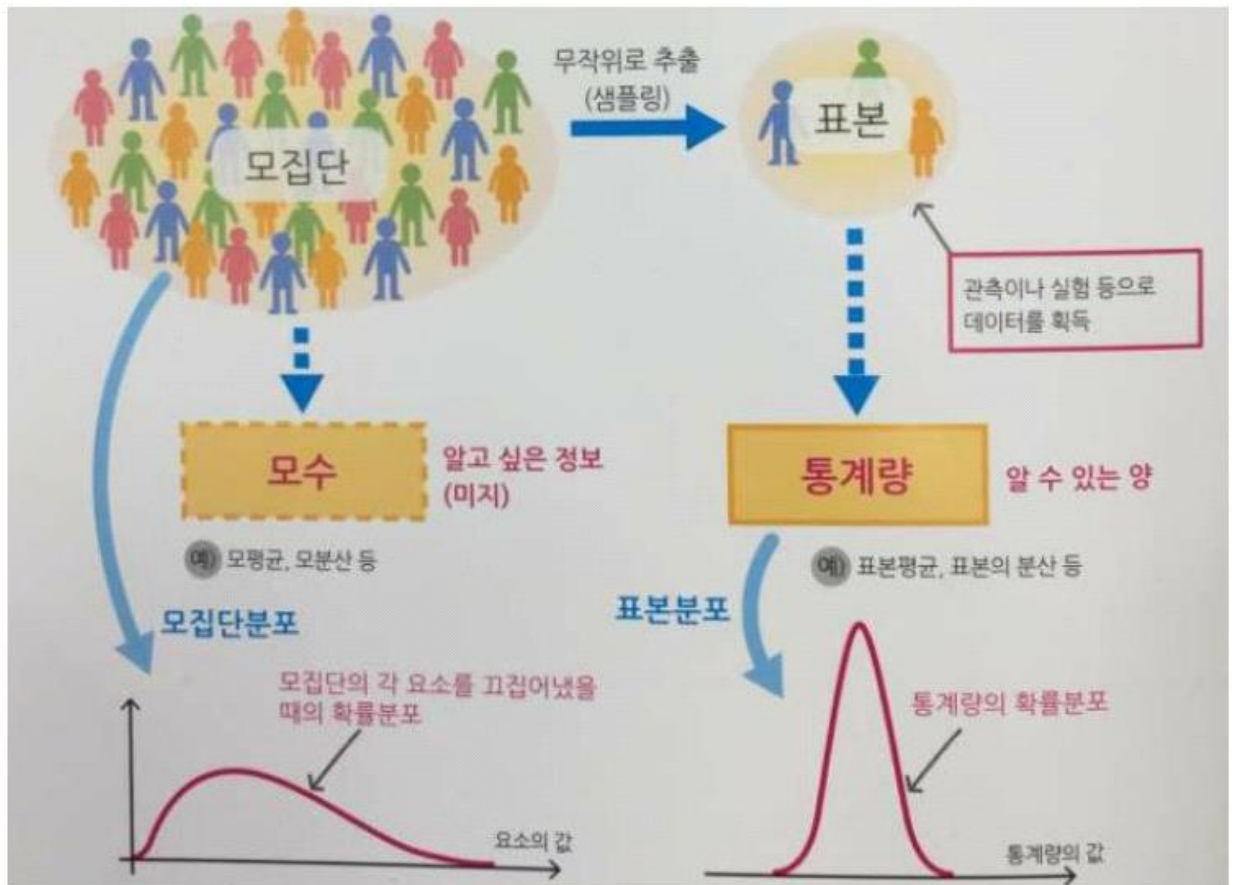
통계

기술통계와 추측통계



기술통계 : 데이터가 수집된 후 이것을 요약, 정리하여 데이터의 특징을 설명하는 분야

추측통계 : 모집단 전체에서 일부를 추출한 표본을 분석함으로써 전체에 관해 추측하고 일반화 시킴



이 추측통계에서는 모집단에서 표본을 추출하여 특정 수치나 수치의 범위로 모수를 추정한다.

추정

" 모집단을 대표할 수 있는 일부를 표본으로 추출하여 분석한 통계량을 이용해서 모집단의 모수를 예측하는 일련의 과정 "

여기에 수식을 입력하십시오.

추정이 필요한 이유?

: 정답을 범위로 제시하면서 분석 자료에 대한 신뢰를 높여주기 위해 필요하다.

점추정과 구간 추정

점추정 : 모수를 특정 수치로 표현하는 것.(ex. 출근시간이 30,40분을 수치로 표현)

10000명의 평균키를 알기위해 100명 표본 학생들의 평균키가 168이면 모평균도 168이다.

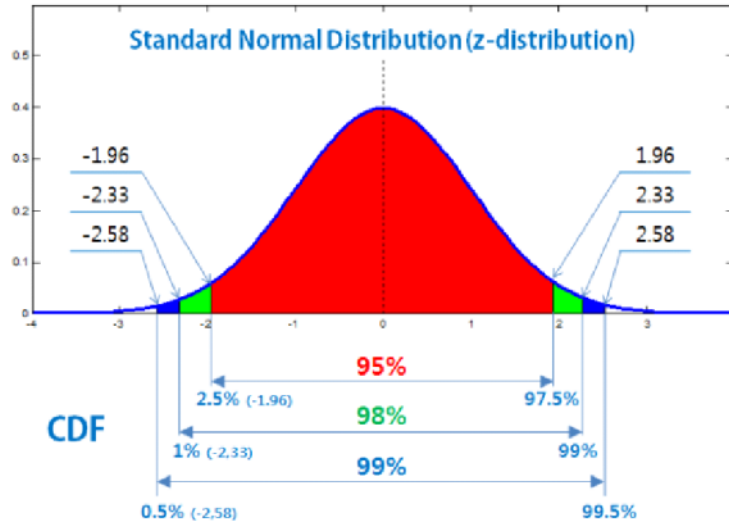
구간 추정 : 모수를 최소값과 최대값을 범위로 추정하는 것(ex. 출근시간 30~40분을 범위로 표현)

모수가 존재할 것으로 예상되는 **신뢰구간**과 그 가능성을 나타내는 **신뢰수준**이 필요하다.

신뢰수준($1-\alpha$) : 표본 통계량을 중심으로 형성된 신뢰구간내에 실제 모수가 존재할 확률

구간(범위)로 추정을 하면 상한과 하한으로 표현되므로 점추정보다 더 정확하게 추정할 수 있습니다

100%의 신뢰도로 구간추정을 하면 조사결과가 모두
맞아야 하므로 추정해야 하는 범위가 너무 넓어집니다

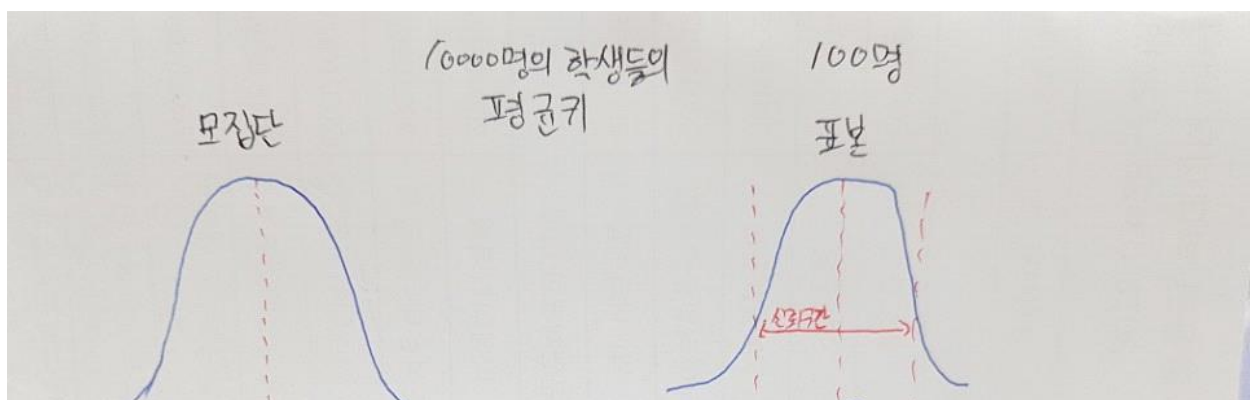


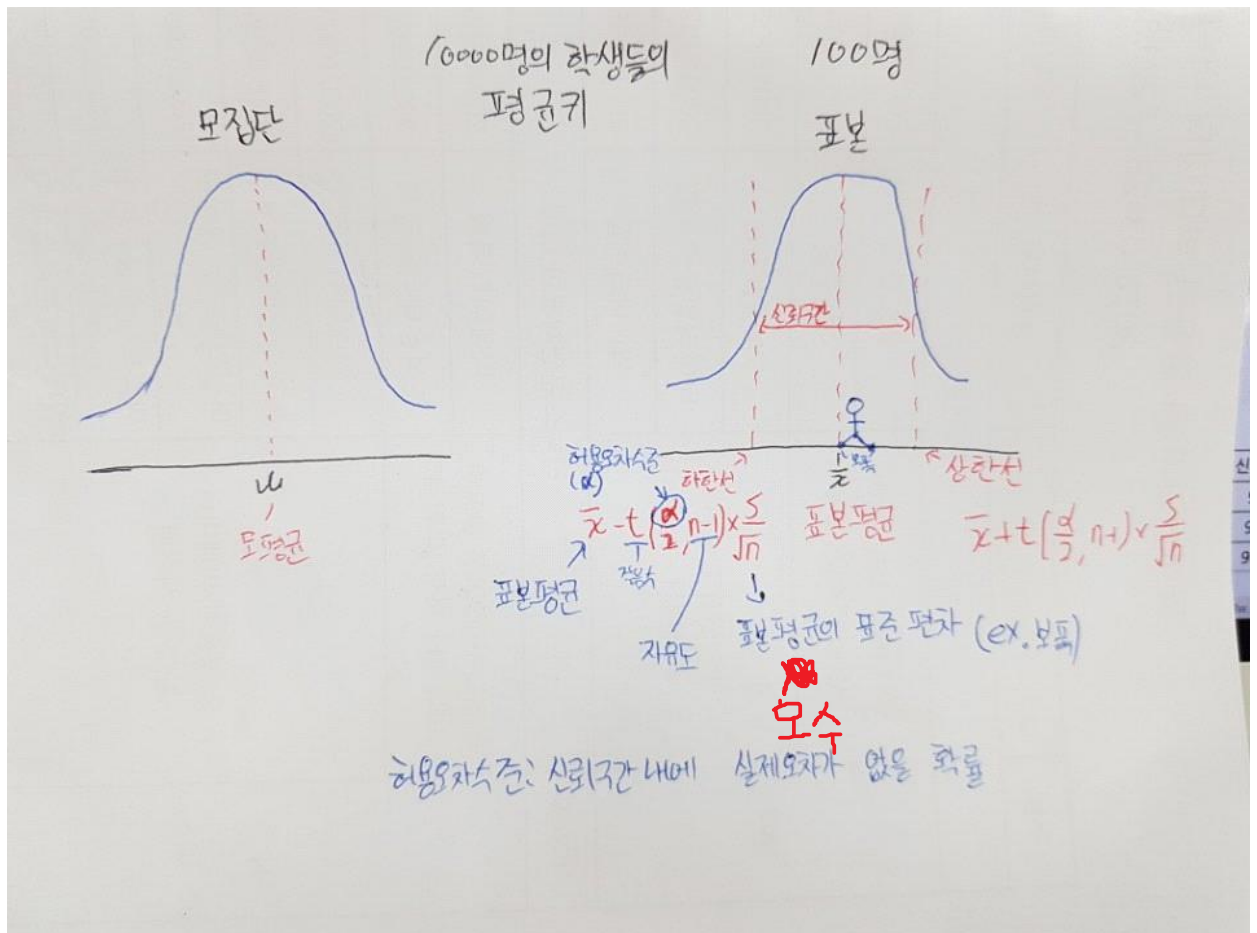
모수가 구간내에 포함될 확률을 나타내는 신뢰수준($1-\alpha$)은 연구나 조사의 목적에 따라 90%(0.9), 95%(0.95), 99%(0.99)로 정할 수 있음.

신뢰도에 따른 구간추정을 보면 **신뢰도가 높을수록**
그 범위가 넓어지는 것을 확인할 수 있습니다

신뢰구간이 좁을수록

조사자가 더 의미있는 결과를 제시할 수 있는것 처럼
보이지만 신뢰수준이 낮아지는것을 감수 해야합니다





문제61. 학생들의 통계학 실력을 살펴보기 위해 전체 학생중 16명을 단순확률 추출하여 시험을 실시했다. 시험결과 표본평균이 175점(200점 만점)이었다. 표준편차가 5점으로 추정되었다, 학생들의 통계학 시험점수가 정규분포를 따른다고 가정했다.

1) 전체 학생의 통계학 점수의 평균을 점추정하여라.

점추정 = 표본평균 이므로 **175**이다.

2)전체 학생의 통계학 점수의 평균에 대한 95%신뢰구간을 구하라.

$$\text{하한값} : \bar{x} - t\left(\frac{\alpha}{2}, n-1\right) \times \frac{S}{\sqrt{n}}$$

$$\text{상한값} : \bar{x} + t\left(\frac{\alpha}{2}, n-1\right) \times \frac{S}{\sqrt{n}}$$

$$\text{하한값} = 175 - (2.13 \times \frac{5}{\sqrt{16}}) = 172.34$$

$$\text{상한값} = 175 + (2.13 \times \frac{5}{\sqrt{16}}) = 177.66$$

3)전체 학생의 통계학 점수의 평균에 대한 99%신뢰구간을 구하라. 이 구간의 너비가95% 신뢰구간과 비교해 어떠한가?

$$\text{신뢰구간의 하한값} = 175 - (2.95 \times \frac{5}{\sqrt{16}}) = 171.31$$

$$\text{신뢰구간의 상한값} = 175 + (2.95 \times \frac{5}{\sqrt{16}}) = 178.69$$

신뢰도가 증가하면 신뢰구간이 너비가 넓어진다.

4) 표본크기가 100명이었을 때 전체 학생의 통계학 점수의 95% 신뢰구간을 구하고, 문제2)의 구간 너비와 비교해 구간 너비가 어떠한가?

$$\text{하한값} = 175 - (2.13 \times \frac{5}{\sqrt{100}}) = 173.94$$

$$\text{상한값} = 175 + (2.13 \times \frac{5}{\sqrt{100}}) = 176.07$$

표본크기가 증가하면 신뢰구간의 너비가 좁아져서 정확한 추정이 된다.

문제62. 대기업의 금면도 대졸자 초임을 알아보기 위하여 15명을 단순 확률 추출하여 성별과 월임금을 조사해보니 다음과 같았다.

보기)

163, 162, 157, 172, 161, 165, 162, 165, 158, 159, 160, 161, 165, 168, 170

1)전체 대졸자 초임 평균에 대한 95%신뢰구간을 구하여라.

$$\text{하한값} : \text{표본평균} - t_{\alpha/2}(2.14) * \frac{S}{\sqrt{n}}$$

$$\text{상한값} : \text{표본평균} + t_{\alpha/2}(2.14) * \frac{S}{\sqrt{n}}$$

답)

```
import numpy as np
x=np.array([163,162,157,172,161,165,162,165,158,159,160,161,165,168,170])
m=np.mean(x)
s=np.std(x)
print('평균 값 : ', m)
print('표준편차 : ', s)
```

```
print('하한값 : ', m-2.14*(s/np.sqrt(15)))
```

```
print('상한값 : ', m+2.14*(s/np.sqrt(15)))
```

```
평균 값 : 163.2
```

```
표준편차 : 4.182503237695499
```

```
하한값 : 160.88897613839904
```

```
상한값 : 165.51102386160093
```

2)남,여 별로 대졸자 초임의 평균에 대한 95% 신뢰구간을 구하여라.

```
x = [['남','남','여','남','남','여','남','여','남','남','여','여','여','남','남'],\
      [163,162,157,172,161,165,162,165,158,159,160,161,165,168,170]]
man=[]
woman=[]
for i in range(len(x[0])):
    if x[0][i]=='남':
        man.append(x[1][i])
    else:
        woman.append(x[1][i])
mm=np.array(man)

m_mean=np.mean(mm)
m_v=np.var(mm)
m_s=np.std(mm)
print('남자 표본평균 : ',np.mean(mm))
print('남자 분산 : ',np.var(mm))
print('남자 표준편차 : ',np.std(mm))
print('남자 신뢰구간의 하한값 : ', (m_mean-2.31*m_s/np.sqrt(len(man)-1)))
print('남자 신뢰구간의 상한값 : ', (m_mean+2.31*m_s/np.sqrt(len(man)-1)),'\n')
```

```

mw=np.array(woman)
w_mean=np.mean(mw)
w_v=np.var(mw)
w_s=np.std(mw)
print('여자 표본평균 :',np.mean(mw))
print('여자 분산 :',np.var(mw))
print('여자 표준편차 :',np.std(mw))
print('여자 신뢰구간의 하한값 :', (w_mean-2.57*w_s/np.sqrt(len(woman)-1)))
print('여자 신뢰구간의 상한값 :', (w_mean+2.57*w_s/np.sqrt(len(woman)-1)))

```

남자 표본평균 : 163.88888888888889
남자 분산 : 21.654320987654323
남자 표준편차 : 4.6534203536382055
남자 신뢰구간의 하한값 : 160.0884017125156
남자 신뢰구간의 상한값 : 167.68937606526217

여자 표본평균 : 162.16666666666666
여자 분산 : 9.472222222222221
여자 표준편차 : 3.0776975521032313
여자 신뢰구간의 하한값 : 158.62934902315328
여자 신뢰구간의 상한값 : 165.70398431018003

설명 : 표본의 분산은 n-1을 해줘야한다.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (X_i - \mu)^2$$

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

문제63. 어느 학교에서 학생12명을 대상으로 1년동안 읽은 책 수를 조사했더니 다음과 같았다.

읽은 평균 책수에 대해 95%의 신뢰구간을 구하여라. (단, 모집단은 정규분포를 따른다고 가정한다.)

보기)

[13, 18, 1, 10, 15, 15, 10, 5, 23, 20, 9, 11]

표본의 평균값? 12.5

표본의 표준 편차? 6.215

표본의 개수? 12

t값 ? 2.201

허용오차 수준? 0.05/2=0.025

자유도(n-1)? 11

```
print(confidence_range(표본평균,표본표준편차,표본개수,t값))
```

하한값 : 8.55, 상한값 : 16.45

답)

```
import numpy as np
def confidence_range(x):
    m=np.mean(x)
    std=np.std(x,ddof=1)
    moe=2.2*std/np.sqrt(x.shape[0])
    return round(m-moe,2), round(m+moe,2)
x=np.array([13,18,1,10,15,15,10,5,23,20,9,11])
print(confidence_range(x))

( 8.55, 16.45)
```

모비율의 추정

비율이란?

: 특정 후보자에 대한 지지율, 어떤 정책이나 안건에 대하여 찬성하는 사람들의 비율, 그리고 특정한 상품을 선택한 사람들이 전체에서 차지하는 비를 말한다.

예: 전체 유권자가 100,000 명 중에서 특정한 A후보를 지지하는 유권자가 35,000이라면
 $35,000/100,000 = 0.35$ 를 구할 수 있다.

이 경우 전체 유권자를 조사하지 않고 이중의 일부를 표본으로 추출하여 그 표본에서 A 후보 지지율을 구한 다음, 그 값으로 전체 유권자의 A후보에 대한 지지율을 추정하는 것을 모집단 **비율 추정**이라 한다.

모집단 비율 --> π

표본 비율 --> p

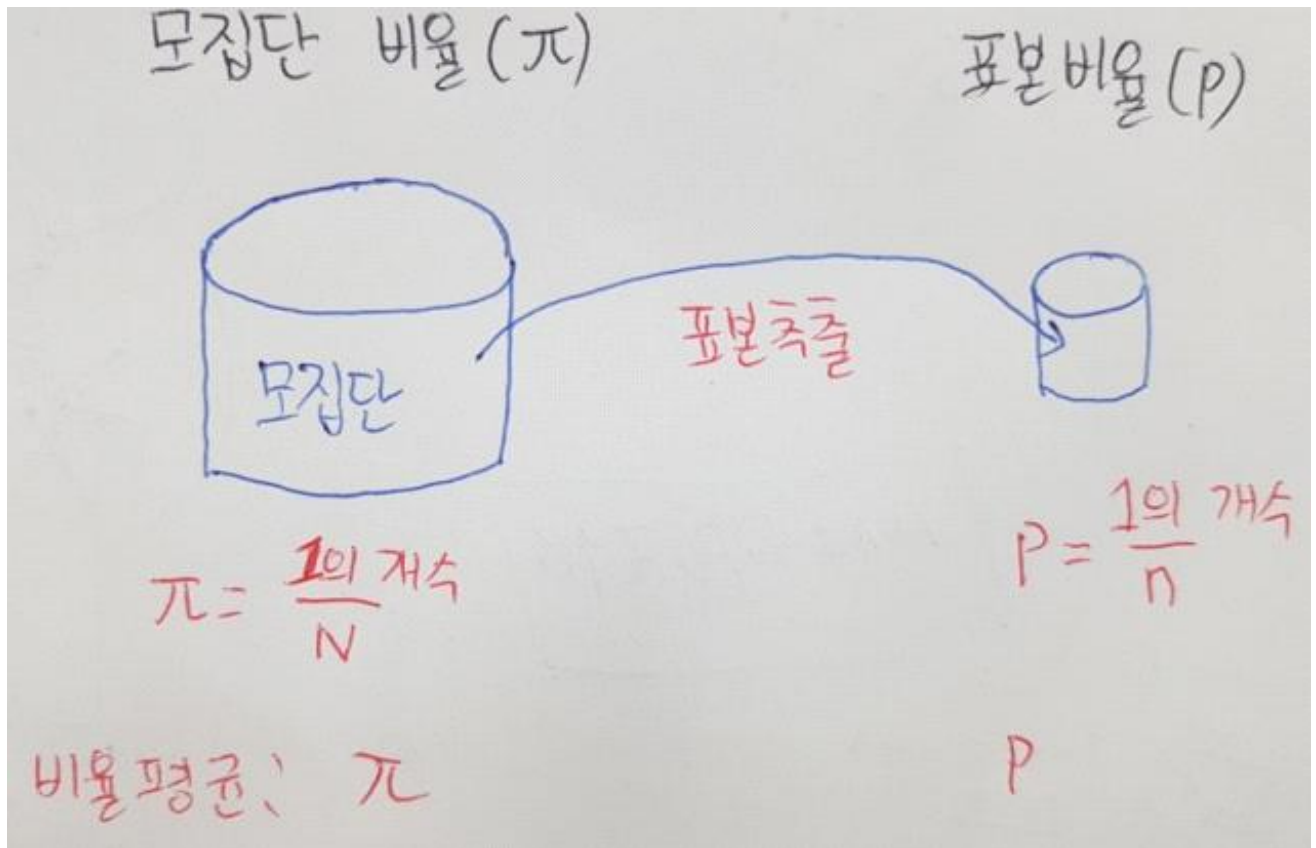
모비율(π)을 표본비율(p)로 추정하는 것이다.

모비율(π)

" 모집단의 구성원 중에 1의 값을 갖는 구성원의 수를 전체 구성원 수로 나눈값, 즉 전체 모집단에서 1의 값을 갖는 구성원의 비율을 의미한다. "

표본비율(p)

" 모집단에서 추출한 표본에서 1의 값을 갖는 구성원의 비율을 의미한다. "



모집단 비율의 평균값이

확률변수(x)	$P(X)$
1	π
0	$1-\pi$

평균 : $\sum x_i \cdot p(x_i) = (1 \cdot \pi) + (0 \cdot (1-\pi)) = \pi$

분산 : $\sum (x_i - \pi)^2 \cdot p(x_i) = \pi(1-\pi)$

표본편차(모집단 비율) : $\sqrt{\pi(1-\pi)}$

표준편차(표본비율) : $\sqrt{p(1-p)}$

① 모집단 평균 추정시 신뢰구간

$$\bar{x} \pm t_{\alpha/2, n-1} \times \frac{\text{표본편차}}{\sqrt{n}}$$

↓

$$\bar{x} \pm t_{\alpha/2, n-1} \times \frac{\text{표준편차}}{\sqrt{n}}$$

평균 모비율 표본비율

π p

② 모집단 비율 추정시 신뢰구간

$$p \pm t_{\alpha/2, n-1} \times \sqrt{\frac{p(1-p)}{n}}$$

표본비율의 평균값

문제64. 새로운 부동산 정책에 찬성하는 사람의 비율을 조사하기 위하여, 우리나라 성인 500명을 임의로 추출하여 조사한 결과, 400명이 찬성한다고 대답했다. 우리나라 성인중 복지정책에 찬성하는 사람의

비율의 추정값을 구하고 95%신뢰구간을 구하라.

표본 비율 : $\frac{400}{500} = 0.8$

표본비율의 평균값 : 0.8

표본비율의 표준편차 : $\sqrt{p(1-p)} =$

표본의 개수 : 500

t 값 : 1.96

허용오차수준 : $0.05 / 2 = 0.025$

자유도(n-1) : 499 #120이후는 결과가 같다

답)

신뢰 하한값 : $0.8 - 1.96 \times \sqrt{0.8(1-0.8)} / \sqrt{500} = 0.765$

신뢰 상한값 : $0.8 + 1.96 \times \sqrt{0.8(1-0.8)} / \sqrt{500} = 0.835$

• 모분산의 추정

언제 이런 분석 작업이 필요한가?

"금년도에 실시된 대입학력고사의 난이도를 평가하기 위해 학력고사 성적의 분산을 알아보고자 한다"

표본으로 모분산값을 추정해야하는데 표본분산은 자유도가 n-1인 카이제곱 분포를 따른다

※카이제곱 분포란?

1900년경 칼 피어슨에 의해 개발되어 모집단 분산에 대한 가설검정이나 교차분석에서 유용하게 사용되는 분포이다.

카이제곱 분포 공식

공식 : $\chi^2_{n-1} = \frac{(n-1) \times S^2}{\sigma^2}$

S^2 → 표본분산
 σ^2 → 모분산
↓
알고 싶은 것에 대해 정리

$\sigma^2 \times \chi^2_{n-1} = (n-1) \times S^2$

$\sigma^2 = \frac{(n-1) \times S^2}{\chi^2_{n-1}}$

모분산
신뢰상한, 하한 값에 쓰인다

문제65. 금년도 대출자 초임을 조사하기 위하여 25명을 단순 확률 복원추출하여 조사 했더니 표본표준 편차가 5만원 이었다.

모분산을 점추정하고 95%신뢰구간을 구하여라!(단, 모집단이 정규분포를 한다고 가정한다.)

보기)

신뢰 하한값, 신뢰 상한값

답)

$$\frac{(25 - 1) * \text{표본분산}(5^2)}{x(24, 0.025)} = 39.364$$

$$\frac{(25 - 1) * \text{표본분산}(5^2)}{x(24, 0.975)} = 12.4$$

$$= (15.242, 48.383)$$

6장. 가설검정의 이해

귀무가설(H_0) : 일반적으로 받아 들여지고 있는 사실

대립가설(H_1) : 우리가 관심이 있고, 결정하고자 하는 가설

제1종 오류와 제2종 오류

		검정의 결과	
		H_0 기각하지 않음	H_0 기각
실제	H_0 참	올바른 판단	제1종 오류
	H_1 참	제2종 오류	올바른 판단

기각 = H_0

채택 = H_1

파이썬

2018년 7월 11일 수요일 오전 11:36

- 파이썬을 배워서 무엇을 하는가?
 - 통계학 : 관심대상에 대하여 관련된 데이터를 수집하고 그 데이터를 요약, 정리하여
이로부터 불확실한 사실에 대한 결론이나 일반적인 규칙성을 이끌어내는
방법을 연구하는 학문
ex) 유동인구 조사
파이썬으로 오브젝트 디텍션 기술을 사용
파이썬으로 웹 스크롤링 해서 데이터 수집
파이썬으로 딥 러닝 신경망 구현을 한다.
정상 폐 사진, 폐 결절 사진을 분류하는 신경망

파이썬을 배워서 해야할 것들

1. 자유롭게 어떤 데이터든 다 긁어올 수 있는 웹스크롤링 능력.
2. 딥러닝 신경망을 파이썬으로 구현하는 것.
3. 데이터 전처리를 위한 함수를 파이썬으로 만드는 것.
4. 게임 프로그램을 인공지능으로 구현하기 위해.

파이썬 149제

2018년 7월 11일 수요일 오후 1:57

1. 대화식 모드로 프로그래밍 하기

1. 대화식 모드

```
>>> a=1
>>> b=1
>>> a+b
2
```

ctrl+d를 누르면 빠져나온다.

```
>>>
[root@wskim ~]#
```

2. 배치모드

```
[root@wskim ~]# vi a.py
```

```
-----
a=1
b=1
print(a+b)
-----
```

저장하고 나와서

```
[root@wskim ~]# python a.py
[root@wskim ~]# python a.py
2
```

2. 텍스트 에디터로 프로그래밍 하기

- 윈도우

1. spyder 프로그램

: 드래그해서 ctrl+enter 나 F9번을 누르면 실행

2. pycharm 프로그램

: 한 라인씩 실행하는 방법 = alt + shift + e

- 리눅스

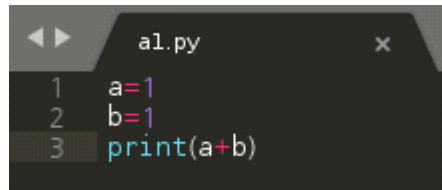
1. sublime text

: <https://www.linuxhelp.com/how-to-install-sublime-text-editor-on-centos-7/>

에 들어가서 설치후에 x윈도우에서 실행해보자

```
Installed:
sublime-text.x86_64 0:3176-1
Complete!
```

```
[oracle@wskim ~]$ su -
암호:
마지막 로그인: 수 7월 1'
시 pts/1
[root@wskim ~]# subl
```



```
a1.py
1 a=1
2 b=1
3 print(a+b)
```

```
[root@wskim ~]# python a1.py
2
-
```

2. vi에서 파이선 문법 자동완성되게 하는 프레임 워크 다운

3. 변수명 만들기

○ 변수 이름 규칙

1. 변수 이름에는 다음 문자만 사용할 수 있다.

소문자(a~z)

대문자(A~Z)

숫자(0~9)

언더 스코어(_)

2. 변수 이름은 숫자로 시작할 수 없다.

3. 예약어를 사용할 수 없다.

ex) False, class, is, return

- 일반 변수

사용예 : 변수 안에서는 데이터가 한개

```
a=100
```

```
print(a)
```

print(type(a)) # a변수의 데이터 유형을 확인하겠다.

```
In [9]: a=100
```

```
In [10]: print(a)
```

```
100
```

```
In [11]: print(type(a))
```

```
<class 'int'>
```

문제1. 문자형 변수를 만들고 type을 확인하시오.

답)

```
a="문자형"
```

```
print(type(a))
```

결과)

```
In [12]: a="문자형"
```

```
In [13]: print(type(a))
<class 'str'>
```

- 리스트 변수

: 변수 안에 여러 개의 데이터가 들어있는 변수

ex)

```
d = [1000,2000,3000,4000]
```

```
print(d)
```

```
print(type(d))
```

```
In [16]: print(type(d))
<class 'list'>
```

문제2. d로 만든 리스트 변수에서 숫자 1000을 출력하시오.

답)

```
print(d[0])
```

결과)

```
In [17]: print(d[0])
1000
```

설명 : 파이썬은 시작을 0부터 시작한다.

문제3. d list변수 안에 있는 요소들을 하나씩 다 출력하시오.

답)

```
for i in d :
```

```
    print(i)
```

결과)

```
In [20]: for i in d :
...:     print(i)
...:
...:
...:
1000
2000
3000
4000
```

설명 : PL/SQL이나 Java, C 등을 사용하던 사람들은 뒤에 ';'을 붙이는데 파이썬은 구문뒤에 아무것도 붙이지 않는다. if문, while loop, for loop, def, class 문의 뒤에
는':'을 붙여야한다.

파이썬이 다른 언어보다 보기 쉽고 소스코드가 간결한 이유는 ':' 을사용해서 들여쓰기를

하도록 만들었기 때문 ':' 뒤에 나오는 명령어를 다음줄에 작성하려면 들여쓰기를 해야 함.

4. 변수에 값 대입하기

설명 : 파이썬에서는 변수의 값에 대입할때 =(assignment) 기호를 사용한다.

ex) a = 7788

파이썬의 코드는 심플함을 철학으로 삼고 있다.

- 파이썬 기본 철학을 확인하는 방법

```
import this
Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated
```

파이썬에서 사용하는 변수 = 객체를 가리키는 것

객체? 우리가 지금까지 보았던 자료형을 포함해 "파이썬에서 사용되는 모든 것"

a = 3 -----> 3이라는 값을 가지는 정수형 자료형이 자동으로 메모리에 생성되고
a는 변수이름이고 3이라는 객체가 저장된 메모리에 위치는 가리킴.

```
>>> a=3
>>> b=3
>>> a is b
In [29]: a is b
Out[29]: True
```

문제4. a와 b는 정말 같은 객체를 가리키는 것인가?

답)

a=3

b=3

import sys

sys.getrefcount(3) # 3이라는 정수형 객체를 참조하는 변수의 개수를 확인

```
In [35]: sys.getrefcount(3)
Out[35]: 6287
```

하나더 늘리기

c=3

sys.getrefcount(3)

```
In [37]: sys.getrefcount(3)
Out[37]: 6288
```

- 특정 객체를 가리키는 변수를 없애는 작업

```

a=3
b=3
c=3
import sys
del c
sys.getrefcount(3)

```

```

>>> sys.getrefcount(3)
55
>>> c=3
>>> sys.getrefcount(3)
56
>>> del(c)
>>> sys.getrefcount(3)
55

```

설명)

사용한 변수를 del 명령어를 이용해서 일일이 삭제할 필요는 없다.

왜냐하면 파이썬이 모든 것을 자동으로 해주기 때문이다.

5. 주석 처리하기(#)

코드를 설명하거나 나중에 어떤 문제를 고치기 위해 표시하는 등 다양한 목적으로 코멘트를 사용할 수 있다.

1. 주석 처리하는 방법

더하는 프로그램이다.

```

a=1
b=1
print(a+b)

```

2. 라인을 유지하기

```
1 + 2 ₩
```

```
+ 3
```

```

>>> 1+2\
... +3
6

```

3. 여러행 주석 처리하는 방법

```
>>> """ asdqwe₩
```

```
... qweasd """
```

```

>>> """ asdqwe\
... qweasd """
' asdqweqweasd '

```

한글 주석하려면 # -*- coding: utf-8 -*- 을 써주면 된다.

6. 자료형 개념 배우기

자료형이란?

: 프로그래밍을 할 때 쓰이는 숫자, 문자열 등 자료 형태로 사용하는 모든 것.

파이썬의 자료형

1. 숫자

2. 문자

3. 리스트 : 데이터의 목록을 다루는 자료형

ex) d = [1,2,3]

4. 튜플 : 데이터의 목록을 다루는 자료형인데 리스트와 다르게 변경이 불가능

ex) f = (1,2,3)

5. 딕셔너리 : key와 value 를 조합해서 사용하는 자료형

ex) m={ }

m['파이썬'] = 'www.python.org'

(인덱스,key)

(값,value)

m

```
>>> m={}
>>> m['python']='www.python.org'
>>> m
{'python': 'www.python.org'}
```

m.keys() #key만 보기

m.values() #value만 보기

6. 집합 자료형

: 리스트 데이터 집합의 교집합, 합집합, 차집합을 출력할 때 유용하게 사용되는
자료형

ex) s1 = set([1,2,3,4,5,6])

s2 = set([4,5,6,7,8,9])

s1&s2 #교집합

```
>>> s1&s2
set([4, 5, 6])
```

s1|s2 #합집합

```
>>> s1|s2
set([1, 2, 3, 4, 5, 6, 7, 8, 9])
```

s1-s2 #차집합

```
>>> s1-s2
set([1, 2, 3])
```

7. 자료형 출력 개념 배우기(print)

: 변수의 값을 출력

ex) d= [1000,2000,3000]

print(d)

```
>>> print(d)
[1000, 2000, 3000]
```

```
>>> for i in d:
...     print(i)
...
1000
2000
3000
```

문제5. emp2.csv 를 파이썬으로 로드해서 print 하시오

```
import csv          #import를 해줘야한다
file = open("/root/emp2.csv",'r') # r : read
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
    print(emp_list)
```

```
['7839', 'KING', 'PRESIDENT', '0', '1981-11-17', '5000', '0', '10']
['7698', 'BLAKE', 'MANAGER', '7839', '1981-05-01', '2850', '0', '30']
['7782', 'CLARK', 'MANAGER', '7839', '1981-05-09', '2450', '0', '10']
['7566', 'JONES', 'MANAGER', '7839', '1981-04-01', '2975', '0', '20']
['7654', 'MARTIN', 'SALESMAN', '7698', '1981-09-10', '1250', '1400', '30']
['7499', 'ALLEN', 'SALESMAN', '7698', '1981-02-11', '1600', '300', '30']
['7844', 'TURNER', 'SALESMAN', '7698', '1981-08-21', '1500', '0', '30']
['7900', 'JAMES', 'CLERK', '7698', '1981-12-11', '950', '0', '30']
['7521', 'WARD', 'SALESMAN', '7698', '1981-02-23', '1250', '500', '30']
['7902', 'FORD', 'ANALYST', '7566', '1981-12-11', '3000', '0', '20']
['7369', 'SMITH', 'CLERK', '7902', '1980-12-09', '800', '0', '20']
['7788', 'SCOTT', 'ANALYST', '7566', '1982-12-22', '3000', '0', '20']
['7876', 'ADAMS', 'CLERK', '7788', '1983-01-15', '1100', '0', '20']
['7934', 'MILLER', 'CLERK', '7782', '1982-01-11', '1300', '0', '10']
```

문제6. 이름과 월급을 출력하시오.

답)

```
[root@wskim ~]# vi a1.py
```

```
import csv
file = open("/root/emp2.csv",'r')
emp_csv = csv.reader(file)
```

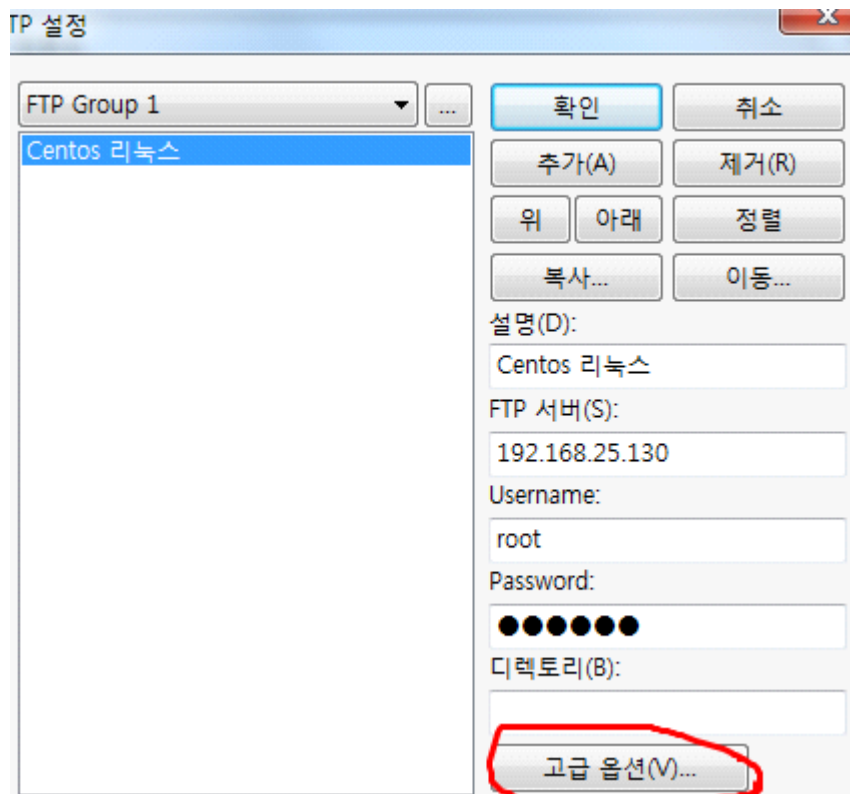
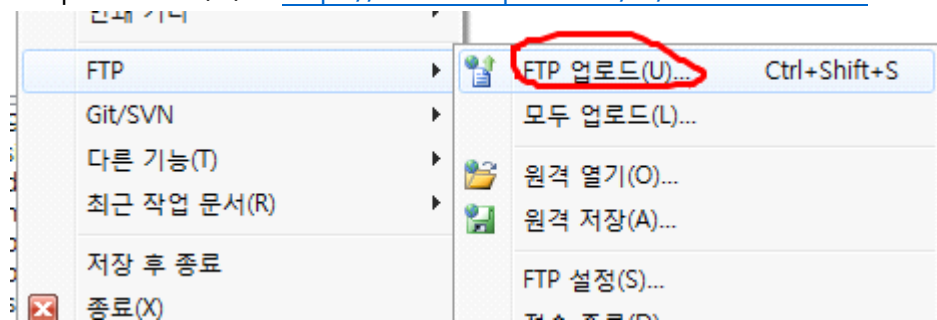
```
for emp_list in emp_csv:
    print(emp_list[1],emp_list[5])
```

```
[root@wskim ~]# python a1.py
```

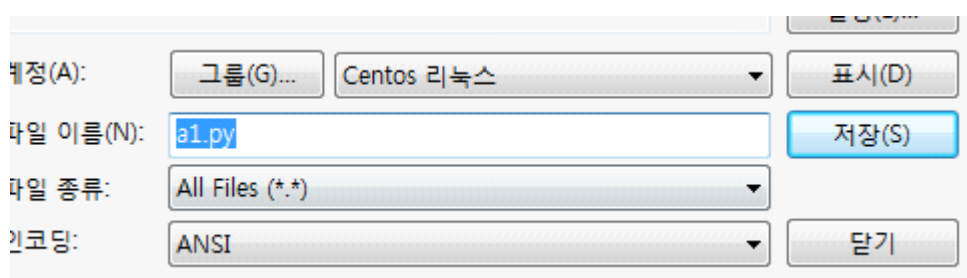
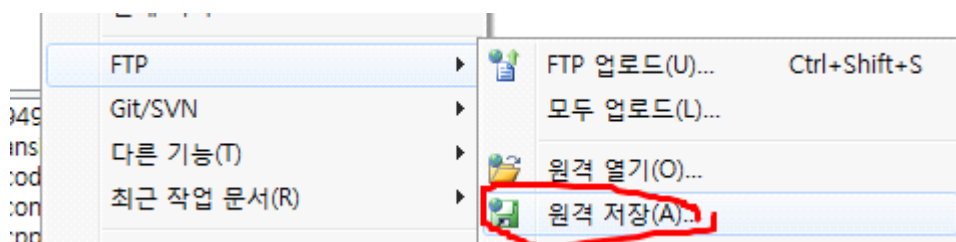
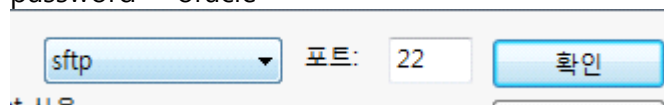
```
[root@wskim ~]# vi a1.py
[root@wskim ~]# python a1.py
('KING', '5000')
('BLAKE', '2850')
('CLARK', '2450')
('JONES', '2975')
('MARTIN', '1250')
('ALLEN', '1600')
('TURNER', '1500')
('JAMES', '950')
('WARD', '1250')
('FORD', '3000')
('SMITH', '800')
('SCOTT', '3000')
('ADAMS', '1100')
('MILLER', '1300')
```

이제부터 editplus에서 하는방법

editplus를 설치하고 <https://www.editplus.com/kr/download.html>



password = oracle



이제 xshell에서 실행하면된다.

```
[root@wskim ~]# python a1.py
```

8. 들여쓰기 개념 배우기

: 파이썬이 다른 언어보다 보기 쉽고 코드가 간결한 이유는 ':' 을 사용해서 들여쓰기를 하도록 만들었기 때문.

파이썬은 블록 구분을 들여쓰기를 통해서 한다.

콜론(:) 뒤에 나오는 명령어를 다음줄에 작성하려면 4칸 들여쓰기를 해야한다.

ex) 월급이 3000인 사원의 이름과 월급을 출력하시오.

```
import csv
file = open("/root/emp2.csv",'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if int(emp_list[5])==3000:
        print(emp_list[1],emp_list[5])
```

```
('FORD', '3000')
('SCOTT', '3000')
```

문제7. 직업이 ANALYST, CLERK인 사원들의 이름과 직업을 출력하시오.

```
for emp_list in emp_csv:
    print(emp_list)
```

```
0 1 2 3 4
['7839', 'KING', 'PRESIDENT', '1', '1981-11-17', '5000', '0', '10']
['7698', 'BLAKE', 'MANAGER', '7839', '1981-05-01', '2850', '0', '30']
['7782', 'CLARK', 'MANAGER', '7839', '1981-05-09', '2450', '0', '10']
['7566', 'JONES', 'MANAGER', '7839', '1981-04-01', '2975', '0', '20']
['7654', 'MARTIN', 'SALESMAN', '7698', '1981-09-10', '1250', '1400', '30']
['7499', 'ALLEN', 'SALESMAN', '7698', '1981-02-11', '1600', '300', '30']
['7844', 'TURNER', 'SALESMAN', '7698', '1981-08-21', '1500', '0', '30']
['7900', 'JAMES', 'CLERK', '7698', '1981-12-11', '950', '0', '30']
['7521', 'WARD', 'SALESMAN', '7698', '1981-02-23', '1250', '500', '30']
```

답)

```
import csv
file = open("/root/emp2.csv",'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
```

```
if (emp_list[2] in ('ANALYST', 'CLERK')):  
    print(emp_list[1],emp_list[2])
```

```
(' JAMES', ' CLERK')  
( 'FORD', ' ANALYST')  
( 'SMITH', ' CLERK')  
( 'SCOTT', ' ANALYST')  
( 'ADAMS', ' CLERK')  
( 'MILLER', ' CLERK')
```

문제8. 부서번호가 10번인 직원들의 이름과 월급을 출력하시오.

답)

```
import csv  
file = open("/root/emp2.csv",'r')  
emp_csv = csv.reader(file)  
  
for emp_list in emp_csv:  
    if (int(emp_list[7]) == 10):  
        print(emp_list[1],emp_list[5])
```

```
(' KING', ' 5000')  
( ' CLARK', ' 2450')  
( ' MILLER', ' 1300')
```

부서번호를 in로 바꿔주거나 10을 문자형으로 바꿔주면 된다.

```
if (emp_list[7] == '10'):
```

파이썬 149제(if, for, while)

2018년 7월 12일 목요일 오전 11:19

9. if문 개념 배우기

ex)

```
# -*- coding: utf-8 -*- <- 한글 적용하기
```

```
x = 1
```

```
y = 2
```

```
if x >= y:
```

```
    print(' x가 y보다 크거나 같습니다')
```

```
else:
```

```
    print(' x가 y보다 작습니다')
```

- 파이썬에서 콜론(:) 을 쓰는 4가지
 1. if
 2. for loop
 3. while loop
 4. def 함수

ex2)

숫자를 물어보게하고 숫자를 입력하면 짝수인지 홀수인지 출력되게 하시오.

```
# -*- coding: utf-8 -*-
```

```
a = int(input('숫자를 입력하세요 ~ : '))
```

```
if a%2 == 0:
```

```
    print('짝수 입니다')
```

```
else:
```

```
    print('홀수 입니다')
```

문제9. 숫자를 두개를 각각 아래 처럼 물어보게하고 아래 처럼 출력되게 하시오.

보기)

첫번째 숫자를 입력하세요 ~ 1

두번째 숫자를 입력하세요 ~ 10

1은 10보다 작습니다.

첫번째 숫자를 입력하세요 ~ 10

두번째 숫자를 입력하세요 ~ 1

10은 1보다 큽니다.

답)

```
num1 = int(input('enter num1~ '))
```

```
num2 = int(input('enter num2~ '))
```

```
if num1 < num2 :
```

```
    print num1,'is smaller than', num2
```

```
elif num1 == num2 :
```

```
    print num1, 'is equal to', num2
```

```
else :
```

```
    print num1, 'is bigger than', num2
```

print() 을 빼면

```
10 is equal to 10
```

이렇게 나오고

붙이면

```
(1, 'is smaller than', 10)
```

이렇게 나온다.

파일을 저장하고 오라클에서 실행한다.

문제10. 이름을 물어보게하고 이름을 입력하면 해당 사원의 월급이 출력되게하시오.

보기)

이름을 입력하세요 ~ SCOTT

답)

```
name=raw_input('enter is name : ') <- raw_ 를 쓰는 이유는 버전 차이!!
```

```
import csv
```

```
file = open("/root/emp2.csv",'r')
```

```
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
```

```
    if emp_list[1] == name:
```

```
        print emp_list[5]
```

파이썬 버전 업그레이드

1. 커뮤니티 레파지토리 추가

```
yum install -y https://centos7.iuscommunity.org/ius-release.rpm
```

2. 설치

우선 파이썬의 버전을 체크하세요

```
python -v
```

그리고 python3.6 버전이 yum을 통해 설치되었는지 체크하세요.

```
yum search python36
```

설치되어 있지 않다면 다음과 같이 설치명령어를 입력하여 설치합니다.

```
yum install -y python36u python36u-libs python36u-devel python36u-pip
```

3. 설치확인

파이썬 3.6 설치완료 확인

```
python3.6 -V
```

설치경로 확인

```
which python3.6
```

4. 심볼릭 링크 걸기

기존 python과 같이 명령어를 쉽게 해주는 링크를 거는 방법입니다.

다음은 입력을 하여 링크된 버전확인.

```
python -v
```

심볼릭 링크는 which로 찾은 경로를 입력하면 됩니다.

```
ln -s /bin/python3.6 /bin/python3
```

```
unlink /bin/python
```

```
ln -s /bin/python3 /bin/python
```

버전이 3.6.5로 바뀌었다

문제11. 문제10번을 다시 해결하는데 scott을 대/소문자 입력 상관없이 출력되게 하시오.

답)

```
name=input('enter is name : ')
```

```
import csv
```

```
file = open("/root/emp2.csv",'r')
```

```
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
```

```
    if emp_list[1] == name.upper():
```

```
        print (emp_list[5])
```

10. if문 개념배우기2

ex) 이름을 물어보게하고 이름을 입력하면 해당 사원의 월급이 3000 이상이면 고소득자

2000 ~ 3000 사이이면 적당합니다.

2000 이하면 월급 인상자 입니다 라는 메시지가 출력되게 하시오.

답)

```
# -*- coding: utf-8 -*-
```

```
name=input('enter is name : ')
```

```
import csv
```

```
file = open("/root/emp2.csv",'r')
```

```
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
```

```
    if emp_list[1] == name.upper():
```

```
        if int(emp_list[5]) >= 3000:
```

```
            print('고소득자 입니다')
```

```
        elif int(emp_list[5]) >= 2000:
```

```
            print('적당합니다')
```

```
        else:
```

```
            print ('월급 인상 대상입니다')
```

문제12. 이름을 물어보게 하고 이름을 입력했을때 해당 사원의 직업이 SALESMAN이면 부서

이동 대상자 입니다. ANALYST면 지역 이동 대상자 입니다. 나머지 직업은 해당사항 없습니다. 라는 메시지가 출력되게 하시오.

답)

```
# -*- coding: utf-8 -*-
```

```
name=input('enter is name : ')
```

```
import csv
```

```
file = open("/root/emp2.csv",'r')
```

```
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
```

```
    if emp_list[1] == name.upper():
```

```
        if emp_list[2] == 'SALESMAN':
```

```
            print('부서 이동 대상자 입니다')
```

```
        elif emp_list[2] == 'ANALYST':
```

```

        print('지역 이동 대상자 입니다')
    else:
        print ('해당 사항이 없습니다')
[root@wskim ~]# python a1.py
enter is name : scott
지역 이동 대상자 입니다
[root@wskim ~]# python a1.py
enter is name : smith
해당 사항이 없습니다

```

11. for문 개념배우기1

파이썬의 for문의 기본적인 구조는 다음과 같다.

for 변수 in 리스트 or 튜플 or 문자열 :

수행할 문장1

수행할 문장2

리스트나 튜플, 문자열의 첫번째 요소부터 마지막 요소까지 차례로 변수에 대입하여 "수행할 문장1", "수행할 문장2" 등이 수행된다.

ex)

```
for i in (1,2,3):
```

```
    print(i)
```

```

[root@wskim ~]# python a1.py
1
2
3

```

ex2)

```
for i in 'I am a boy':
```

```
    print(i)
```

```

[root@wskim ~]# python a1.py
I
a
m
a
b
o
y

```

문제13. for 문을 이용해서 숫자 1부터 10까지 출력하시오.

보기)

```

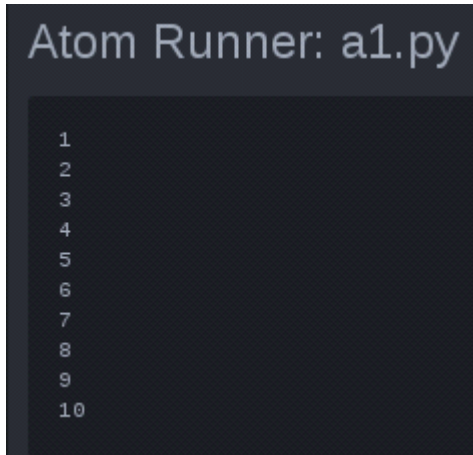
1
2
3

```

```
.  
.   
9  
10
```

답)

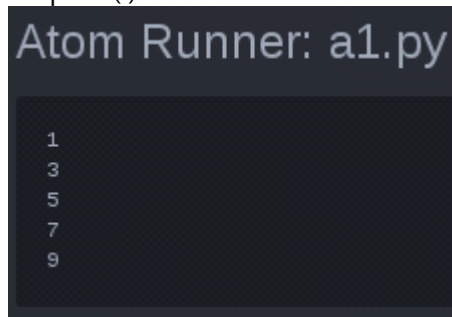
```
for i in range(1,11):  
    print(i)
```



```
Atom Runner: a1.py  
  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

tip : 2씩 증가 하고 싶다면?

```
for i in range(1,11,2):  
    print(i)
```



```
Atom Runner: a1.py  
  
1  
3  
5  
7  
9
```

문제14. 아래와 같이 결과를 출력하시오.

보기)

```
★  
★★  
★★★  
★★★★  
★★★★★  
★★★★★★  
..  
..  
★★★★★★★★★★
```

답)

```
for i in range(1,11):  
    print(i*'★')
```


문제15. 숫자를 물어보게하고 숫자를 입력하면 ★이 숫자만큼 출력되게 하시오.

보기)

숫자를 입력하세요 ~ :10

★

★★

..

..

★★★★★★★★★★

답)

```
# -*- coding: utf-8 -*-
```

```
num=int(input('숫자를 입력하세요 ~ : '))
```

```
for i in range(1,num+1):
```

```
    print(i*'★')
```

문제16. 숫자를 물어보게하고 숫자를 입력하면 ★이 아래와 같이 출력되게 하시오.

보기)

숫자를 입력하세요 ~ : 10

★★★★★★★★★★

★★★★★★★★★★

.

.

★★

★

답)

```
# -*- coding: utf-8 -*-
```

```
num=int(input('숫자를 입력하세요 ~ : '))
```

```
for i in range(num,0,-1):
```

```
    print(i*'★')
```

문제17. 중첩 for loop문을 이용하지 않은 상태에서 ★로 사각형을 만드시오.

보기)

가로의 숫자를 입력하세요 ~ 5

세로의 숫자를 입력하세요 ~ 6

★★★★★

★★★★★

★★★★★

★★★★★

★★★★★

★★★★★

답)

```
# -*- coding: utf-8 -*-
num1=int(input('가로의 숫자를 입력하세요~ : '))
num2=int(input('세로로 숫자를 입력하세요~ : '))

for i in range(num2):
    print('★' * num1)
```

문제18. 아래와 같이 ★을 출력하시오.

보기)

숫자를 입력하세요~ 5

```
★
★★
★★★
★★★★
★★★★★
```

답)

```
# -*- coding: utf-8 -*-
num1=int(input('숫자를 입력하세요~ : '))
b=num1+1
for i in range(1,num1+1):
    print(' '*b, '★' * i)
    b -= 1
```

문제19. 구구단 2단을 출력하시오.

보기)

```
2 x 1 = 2
2 x 2 = 4
.
.
2 x 9 = 18
```

답)

```
for i in range(1,10):
    print('2 x',i,'=', 2*i )
```

```

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18

```

문제20. 중첩 for loop문을 사용해서 2단부터 9단까지 출력하시오.

답)

```

for i in range(2,10):
    for j in range(1,10):
        print(i, 'x', j, '=', i*j)

```

```

2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9

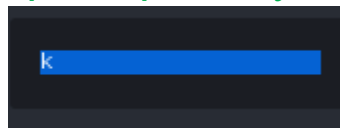
```

```

8 x 6 = 48
8 x 7 = 56
8 x 8 = 64
8 x 9 = 72
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81

```

tip) >>> print('k'.ljust(20))



SQL로 치면 select lpad(문자열,20,' ')
from emp;

전체 20자리이고 k 한글자 제외하고 나머지 19글자는 공백으로 출력된다
result = "

```
for i in range(2,10):
    result += str(i) + ' x '
    print(result)
```

문제21. 구구단 2단을 가로로 출력하시오.

보기)

$$2 \times 1 = 2 \quad 2 \times 2 = 4 \quad 2 \times 3 = 6 \quad \dots \quad 2 \times 9 = 18$$

답)

```
result = ""
for i in range(2,10):
    result += '2' + 'x' + str(i) + '=' + str(2*i).ljust(5)

print(result)
```

설명 : str() --> 문자로 형변환
 + ---> 연결 연산자
 += ---> i+=1 == i = i + 1

문제22. 구구단 2단부터 9단까지 전체를 출력하시오.

보기)

$$\begin{array}{l} 2 \times 1 = 2 \quad 3 \times 1 = 3 \quad 4 \times 1 = 4 \quad \dots \quad 9 \times 1 = 9 \\ 2 \times 2 = 4 \quad 3 \times 2 = 6 \quad 4 \times 2 = 8 \quad \dots \quad 9 \times 2 = 18 \\ \vdots \\ 2 \times 9 = 18 \quad 3 \times 9 = 27 \quad 4 \times 9 = 36 \quad \dots \quad 9 \times 9 = 81 \end{array}$$

답)

```
result = ""
for i in range(1,10):
    for j in range(2,10):
        result += str(j) + 'x' + str(i) + '=' + str(j*i).ljust(5)
    print(result)
result=""
```

문제23. (마지막 문제) 숫자를 물어보게하고 숫자를 입력하면 마름모가 나오게 하시오.

보기)

숫자를 입력하세요~ 5

```
★
★★
★★★
★★★★
★★★★★
```

```

★★★★
★★★
★★
★

```

답)

```
# -*- coding: utf-8 -*-
```

```
a=int(input('숫자를 입력하세요~'))
```

```
for i in range(1,a):
    print(' ' * (a-i), '□' * i)
```

```
for j in range(a,0,-1):
    print(' ' * (a-j), '□' * j)
```

```

`숫자를 입력하세요~5
  □
 □ □
□ □ □
□ □ □ □
□ □ □ □ □
 □ □ □ □
  □ □ □
   □ □
    □

```

14. for문 개념배우기2(for ~ continue ~ break)

1. continue 문

반복문이 실행되는 동안 특정 코드 블록을 실행하지 않고 다른 코드 블록만 실행되게 할 때 사용하는 문법

```

for i in range(10):
    if i % 2 == 1:
        continue
    print(i)
0
2
4
6
8

```

문제24. 숫자 1부터 10까지 출력하는데 중간 5는 출력되지 않게 하시오

답)

```

for i in range(1,11):
    if i == 5:

```

```

        continue
    print(i)
1
2
3
4
6
7
8
9
10

```

문제 25) 아래의 학생 점수 리스트를 이용해서 60점 이상 인 학생들에게는 축하 메시지를 보내고 나머지 학생들에게는 아무런 메시지도 전하지 않는 프로그램을 작성하시오
보기)

```
score=[90, 25, 67, 45, 80]
```

답)

```
score=[90, 25, 67, 45, 80]
```

```
num=0
```

```
for i in score:
```

```
    num+=1
```

```
    if i<=60 :
```

```
        continue
```

```
    print(i, '점입니다.', '%d 번학생 합격 축하' %num)
```

```
90 점입니다. 1 번학생 합격 축하
```

```
67 점입니다. 3 번학생 합격 축하
```

```
80 점입니다. 5 번학생 합격 축하
```

문제 26) emp2.csv를 로드해서 월급이 3000 이상이면 축하메시지를 보내고 작으면 아무런 메

시지도 보내지 않게 하시오.

답)

```
import csv
```

```
file = open("C:/python_data/emp2.csv",'r')
```

```
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
```

```
    if int(emp_list[5])<3000:
```

```
        continue
```

```
    print(emp_list[1], 'ㄸㄸㄸ')
```

```
KING ㄸㄸㄸ
```

```
FORD ㄸㄸㄸ
```

```
SCOTT ㄸㄸㄸ
```

문제 27. 위의 스크립트를 수정해서 1 ~ 100 의 숫자가 출력되는 루프문을 만드는데 숫자를

물어보게 해서 해당 숫자에서 loop 문이 종료되게 하시오.

답)

```
stop_num=int(input())
```

```
for i in range(1, 101):
```

```
    print(i)
```

```
    if i == stop_num:
```

```
        break
```

```
7
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

15 while문 개념 배우기 (while ~ continue ~ break)

기본구조)

```
while <조건문>:
```

```
    <수행할 문장1>
```

```
    <수행할 문장2>
```

```
    <수행할 문장3>
```

ex)

```
x = 0
```

```
while x < 10:
```

```
    x = x+1
```

```
    if x < 3:
```

```
        continue
```

```
    print(x)
```

```
    if x > 7:
```

```
        break
```

문제28. while 루프문을 이용해서 숫자 1 ~ 100 를 출력하는데 숫자를 물어보게 하고 해당 숫

자만큼 숫자가 출력되게 하시오.

답)

```
num = int(input("Enter the number: "))
```

```
i=0
```

```

while i < 100:
    i+=1
    print(i)
    if i==num:
        break
Enter the number: 8
1
2
3
4
5
6
7
8

```

문제29. log 함수를 파이썬으로 구현하시오 (while loop 문으로)

답)

```

meetsu=int(input("밑수 입력 : "))
jinsu=int(input('진수 입력 : '))
a=0
b=0

```

```

while True:
    b+=1
    a=jinsu/meetsu
    if a ==1:
        break
    jinsu=a
print(b)

```

```

밑수 입력 : 2
진수 입력 : 16
4

```

문제30. factorial 을 while loop 문으로 구현하시오

답)

```

num=int(input("숫자 입력 : "))
a=num
res=1

```

```

while a>1:
    res*=num
    num-=1

```



```
a-=1
print(res)
숫자 입력 : 5
120
```

16. None의 개념배우기

: None은 Types.NonType 의 유일한 값으로, 값이 존재하지 않는 변수에 대입하여 이 변수에 아무런값이 없다는 것을 나타내기 위해 주로 활용된다

```
val = None
```

위의 val이라는 변수는 아무런 값도 없는 빈 깡통 변수라고 생각하면 된다

```
ex)
val = None
condition = 1
if condition == 1:
```

```
val = [1, 2, 3]
else:
```

```
val = 'I love Python'
print(val)
```

설명)

조건이 1이면 숫자 값 리스트가 할당되기도 하고

조건이 1이 아니면 문자열이 할당되기도 한다.

그렇기 때문에 변수를 만들때 빈 깡통인 'None' 으로 만들어 주는 것이다.

문제31. 커미션이 null 인 직원들의 이름과 커미션을 출력하시오

```
답)
import csv
file = open("C:\wpython_data\wemp_comm.csv", "r")
emp_csv = csv.reader(file)
for emp in emp_csv:
    if emp[6] == "":
        print(emp[1], emp[6])
```

문제32. 커미션이 null 이 아닌 직원들의 이름과 커미션을 출력하시오

```
답)
import csv
```

```
emp_csv = csv.reader(file)
for emp in emp_csv:
    if emp[6] != "":
        print(emp[1], emp[6])
```

답)

```
def nvl(val1, val2):
    if val1 is "":
        return val2
    return val1
```

```
print(emp[1], nvl(emp[6], 0)) # null이면 0이 출력되라, 아니면 emp[6] 을 출력해라
```

: 자연수 $(1, 2, 3, \dots)$ 와 음수 $(-1, -2, -3, \dots)$ 와 0으로 이루어진 수 체계를 정수라 한다.

파이썬은 메모리가 허용하는 범위에서 지원 가능한 수를 사용할 수 있다.

```
print(a)
```

파이썬1 페이지 90

: 실수는 소수로 나타낼 수 있는 유리수와 소수로 나타낼 수 없는 무리수로 구성된 집합

파이썬은 실수를 지원하기 위해서 부동 소수형을 제공한다.

부동 소수형의 특징

1. 8바이트만 이용해서 수를 표현한다. 즉, 한정된 범위의 수만 표현할 수 있다.
2. 디지털 방식으로 소수를 표현해야 하므로 정밀도의 한계가 있다.

```
ex)
a=3.14
print(type(a))
<class 'float'>
```

문제35. $b = 22/7$ 의 결과는 무한 소수인가 유한 소수인가?

```
답)
b=22/7
print(b)
print(type(b))
3.142857142857143
<class 'float'>
```

설명)

소수점 이하 15자리만 표현하고 끝낼 수 있는 수가 아니다.

8바이트만 이용해서 수를 표현하기 때문에 한정된 범위의 수만 표현한 것이다.

문제36. 소수점 43.2 - 43.1 을 하면 얼마인지 파이썬으로 구현하시오

```
답)
43.2-43.1
Out [58]: 0.100000000000000142
```

설명)

부동 소수형은 정밀도의 한계를 가지고 있다.

이것은 파이썬 만의 문제가 아니라, 부동 소수형이 어떻게 구성되고 동작되는가를 정의하는 표준인 IEEE 754를 따르는 모든 컴퓨터 시스템의 문제이다.

IEEE 754 는 제한된 메모리를 이용해서 (파이썬의 경우 8바이트) 실수를 표현하기 때문에 제한된 정밀도를 갖는다.

문제37. while loop 문을 이용해서 코드를 작성하는데 두 수를 입력 받아서 최대 공약수를 출력하시오.

```
답)
x = int(input("첫번째 수 입력: "))
y = int(input("두번째 수 입력: "))
```

SWAP 하는 방법

```
if y > x:
```

```
    (x, y) = (y, x)
```

```
while y > 0 :
```

```
    remain = x % y
```

```
    x = y
```

```
    y = remain
```

```
print(x)
```

첫번째 수 입력: 2

두번째 수 입력: 3

1

19. 대입 연산자 이해하기

a = 1

왼쪽의 변수에 오른쪽의 값을 대입해라

대입 연산자와 일반 연산자 비교

=	대입연산자
==	같다
in	여러 개 비교

문제38. 아래의 리스트 변수에서 **positive** 라는 단어는 몇 개가 나오는가?

보기)

```
word = ['winter', 'cold', 'positive', 'negative']
```

답1)

```
word = ['winter', 'cold', 'positive', 'negative']
```

```
sum=0
```

```
for i in word:
```

```
    if i == 'positive':
```

```
        sum += 1
```

```
print(sum)
```

답2)

```
word = ['winter', 'cold', 'positive', 'negative']
```

```
print(word.count('positive'))
```

문제39. word 리스트에 positive와 negative를 조회하시오

```
답)
word = ['winter', 'cold', 'positive', 'negative']
sum=0

for i in word:
    if i in ('positive','negative'):
        sum += 1
print(sum)
```

문제40. 겨울왕국 대본에는 긍정적인 단어가 몇 개가 들어있는가? (jupyter)

라이나 생명 데이터 분석 프로젝트

보험 상담원의 상담사가 규칙에 맞는 단어를 사용해서 상담을 했는지 판독하는 프로젝트

상담원의 상담 내용 녹음(음성) ---> 텍스트로 변환

예) 90일이라고 얘기해야 하는데 3달이라고 얘기하면 안된다

이런데 사용 될 수 있다.

```
답)
# 겨울왕국 대본을 가져오는 과정
winter = open("C:\python_data\winter.txt", "r")
winter = winter.read()
# 겨울왕국의 긍정 단어
positive = open("C:\python_data\positive-words.txt", "r")
positive_line = positive.read().split('\n')
sum = 0
for i in winter.split(' '):
    if i.lower() in positive_line:
        sum += 1
print(sum)
330
```

20. 사칙연산 이해하기(+, -, *, /, **)

	오라클	파이썬
산술 연산자	+	+
	-	-
	/	/
	*	*

	mod	%
	power	**
비교 연산자	>	>
	>=	>=
	<	<
	<=	<=
	=	==
	!=	!=
논리 연산자	and	&
	or	
	not	!
기타 비교 연산자	between and ...	<= & >=
	in	in
	is null	==''
	like	^, \$, 정규식 표현

문제41. 월급이 1000 에서 3000 사이 인 직원들의 이름, 월급을 출력하시오
 답)

```
import csv
file=open("C:\python_data\emp2.csv",'r')
emp_csv=csv.reader(file)
```

```
for emp_list in emp_csv:
    if 1000 <= int(emp_list[5]) <= 3000:
        print(emp_list[1],emp_list[5])
BLAKE 2850
CLARK 2450
JONES 2975
MARTIN 1250
ALLEN 1600
TURNER 1500
WARD 1250
FORD 3000
SCOTT 3000
ADAMS 1100
MILLER 1300
```

문제42. 부서번호가 10, 20번이 아닌 직원들의 이름, 부서번호를 출력하시오
 답)

```

import csv
file=open("C:\python_data\emp2.csv",'r')
emp_csv=csv.reader(file)

for emp_list in emp_csv:
    if int(emp_list[7]) not in [10,20]:
        print(emp_list[1],emp_list[7])
BLAKE 30
MARTIN 30
ALLEN 30
TURNER 30
JAMES 30
WARD 30

```

문제43. 이름과 이름의 첫번째 철자만 출력하시오

답)

```

import csv
file=open("C:\python_data\emp2.csv",'r')
emp_csv=csv.reader(file)

```

```

for emp in emp_csv:
    print(emp[1][0])

```

```

K
B
C
J
M
A
T
J
W
F
S
S
A
M

```

문제44. 이름과 이름의 첫번째 철자에서 두번째 철자까지 출력하시오

답)

```

import csv
file=open("C:\python_data\emp2.csv",'r')
emp_csv=csv.reader(file)

```

```

for emp in emp_csv:
    print(emp[1][0:2])

```

```
KI  
BL  
CL  
JO  
MA  
AL  
TU  
JA  
WA  
FO  
SM  
SC  
AD  
MI
```

문제45. 이름의 두번째 철자가 M 인 직원들의 이름을 출력하시오.

답)

```
import csv  
file=open("C:\python_data\emp2.csv",'r')  
emp_csv=csv.reader(file)  
  
for emp_list in emp_csv:  
    if emp_list[1][1]=='M':  
        print(emp_list[1])
```

문제46. dept.csv 를 로드해서 부서번호와 부서위치를 출력하시오

답)

```
file2=open("C:\python_data\dept2.csv",'r')  
dept_csv=csv.reader(file2)  
  
for dept in dept_csv:  
    print(dept[0], dept[2])
```

문제47. 오라클이 rdbms를 만들어서 크게 성공을 했는데 rdbms 의 꽃이 조인이다.

이름과 부서위치를 출력하시오.

답)

```
import csv  
file1=open("C:\python_data\emp2.csv",'r')  
emp_csv=csv.reader(file1)  
file2=open("C:\python_data\dept2.csv",'r')  
dept_csv=csv.reader(file2)  
  
for dept in dept_csv:  
    for emp in emp_csv:
```



```
if emp[7] in dept[0]:  
    print(emp[1], dept[2])  
file1=open("C:\python_data\emp2.csv",'r')  
emp_csv = csv.reader(file1)
```

파이썬 149제(연산자)

2018년 7월 16일 월요일 오전 10:06

21. 연산자 축약 이해하기(+=, -=, *=, /=)

ex)

```
a=0
```

```
a+=1
```

```
print(a)      --> a= 1
```

```
a-=5
```

```
print(a)      --> a= -4
```

```
a*=2
```

```
print(a)      --> a= -8
```

```
a/=4
```

```
print(a)      --> a= -2
```

문제48. 아래의 a의 결과가 무엇이 나오는지 파이썬으로 구현하시오.

보기)

a의 값을 입력하세요~ 10

```
a+=1
```

```
a-=5
```

```
a*=2
```

```
a/=4
```

```
print(a)
```

답)

```
a=int(input('a의 값을 입력하세요~ :'))
```

```
a+=1
```

```
a-=5
```

```
a*=2
```

```
a/=4
```

```
print(a)
```

```
a의 값을 입력하세요~ :10
3.0
```

22. True와 False 이해하기

ex)

```
a = True
```

```

b= False
print(a == 1)
print(b != 0)
    True
    False

```

문제49. True를 사용해서 while loop문의 무한 루프문을 구현해서 숫자 7이 반복해서 출력되게 하시오.

답)

```

while True:
    print(7)

```

23. 관계 연산자 이해하기 (==, !=, <, <=, >, >=)

```

ex)
x = 1
y = 2

str1='abc'
str2='python'

print(x==y)
print(x!=y)
print(str1==str2)
print(str2=='python')
print(str1<str2)      # 아스키 코드값
False
True
False
True
True

```

문제50. 직업이 SALESMAN이 아닌 직원들의 이름과 직업을 출력하시오.

답)

```

import csv

file = open("C:/python_data/emp2.csv", "r")
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if emp_list[2] != 'SALESMAN':
        print(emp_list[1], emp_list[2])

```

결과)

```
KING PRESIDENT
BLAKE MANAGER
CLARK MANAGER
JONES MANAGER
JAMES CLERK
FORD ANALYST
SMITH CLERK
SCOTT ANALYST
ADAMS CLERK
MILLER CLERK
```

문제51. 위의 결과를 pandas를 이용해서 출력하시오.(컬럼이름이 있어야됨)

답)

```
import pandas as pd #as pd : 별칭
```

```
emp = pd.read_csv("C:/python_data/emp.csv")
result = emp[['ename', 'job']][emp['job'] == 'SALESMAN']
print(result)
```

결과)

	ename	job
4	MARTIN	SALESMAN
5	ALLEN	SALESMAN
6	TURNER	SALESMAN
8	WARD	SALESMAN

- 판다스 기본 문법

판다스 데이터 프레임 [열][행]

emp [열 선택] [행 선택]

문제52. 월급이 3000이상인 직원들의 이름과 월급을 출력하시오.

1. pandas를 이용하지 않은 코드
2. pandas를 이용한 코드

답1)

```
import csv
file = open("/root/emp2.csv",'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if int(emp_list[5]) >= 3000:
        print(emp_list[1],emp_list[5])
KING 5000
FORD 3000
SCOTT 3000
```

답2)

```
import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")
result = emp[ ['ename', 'sal'] ] [emp['sal'] >= 3000]
print(result)
```

	ename	sal
0	KING	5000
9	FORD	3000
11	SCOTT	3000

24. 논리 연산자 이해하기(and, or, nor)

문제53. 직업이 SALESMAN이고 월급이 1200이상인 직원들의 이름과 월급과 직업을 출력하시오. (pandas 이용 x, pandas 이용 o)

답1)

```
import csv
file = open("C:/python_data/emp2.csv",'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if (emp_list[2]) == 'SALESMAN' and (int(emp_list[5]) >= 1200):
        print(emp_list[1],emp_list[2],emp_list[5])
MARTIN SALESMAN 1250
ALLEN SALESMAN 1600
TURNER SALESMAN 1500
WARD SALESMAN 1250
```

답2)

```
import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")
result = emp[ ['ename', 'sal','job'] ] [(emp['sal'] >= 1200) & (emp['job']=='SALESMAN')]
print(result)
```

#괄호로 묶어줘야한다

why?

& 연산자가 == !=연산자들 보다 우선순위가 높아서 괄호를 묶어주지 않을 시 &가 먼저 수행됨

• 기타 비교 연산자를 오라클과 파이썬과 판다스와 비교

오라클	vs	파이썬	vs	판다스
between..and		<= & >=		[(emp['sal']>=1000) & (emp['sal'] <= 3000)]

				도 가능하고 .between(,)
in		in		isin
like		%, %, 정규식함수		apply 함수
is null		== ''		isnull()

```
>>> emp[ 'ename' ] [ emp[ 'sal' ] . between( 1200, 3000) ]
```

문제54. 직업이 SALESMAN, ANALYST인 직원들의 이름과 직업을 출력하시오.

답1)

```
import csv
file = open("C:/python_data/emp2.csv",'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if (emp_list[2] in ('SALESMAN', 'ANALYST')):
        print(emp_list[1], emp_list[2])
MARTIN SALESMAN
ALLEN SALESMAN
TURNER SALESMAN
WARD SALESMAN
FORD ANALYST
SCOTT ANALYST
```

답2)

```
import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")
result = emp[ ['ename', 'job'] ] [emp['job'].isin( ['SAELSMAN', 'ANALYST'] ) ]
print(result)
   ename    job
9   FORD  ANALYST
11  SCOTT  ANALYST
```

문제55. 직업이 SALESMAN, ANALYST가 아닌 직원들의 이름과 직업을 출력하시오.

답1)

```
import csv
file = open("C:/python_data/emp2.csv",'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if (emp_list[2] not in ('SALESMAN', 'ANALYST')):
        print(emp_list[1], emp_list[2])
```

	ename	job
4	MARTIN	SALESMAN
5	ALLEN	SALESMAN
6	TURNER	SALESMAN
8	WARD	SALESMAN
9	FORD	ANALYST
11	SCOTT	ANALYST

답2)

```
import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")
result = emp[ ['ename', 'job'] ] [ ~emp['job'].isin( ['SALESMAN', 'ANALYST'] ) ]
print(result)
```

	ename	job
0	KING	PRESIDENT
1	BLAKE	MANAGER
2	CLARK	MANAGER
3	JONES	MANAGER
7	JAMES	CLERK
10	SMITH	CLERK
12	ADAMS	CLERK
13	MILLER	CLERK

문제56. 커미션이 null인 직원들의 이름과 커미션을 출력하시오.

답1)

```
import csv
file = open("C:/python_data/emp.csv",'r')
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
    if emp_list[6]=="":
        print(emp_list[1], emp_list[6])
```

KING
BLAKE
CLARK
JONES
JAMES
FORD
SMITH
SCOTT
ADAMS
MILLER

답2)

```
import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")
result = emp[ ['ename', 'comm'] ] [ emp['comm'].isnull() ]
print(result)
```

	ename	comm
0	KING	NaN
1	BLAKE	NaN
2	CLARK	NaN
3	JONES	NaN
7	JAMES	NaN
9	FORD	NaN
10	SMITH	NaN
11	SCOTT	NaN
12	ADAMS	NaN
13	MILLER	NaN

문제57. 이름의 첫번째 철자가 S로 시작하는 직원들의 이름을 출력하시오.

답1)

```
import csv
file = open("C:/python_data/emp.csv",'r')
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
    if emp_list[1][0]=='S':
        print(emp_list[1])
SMITH
SCOTT
```

답2)

```
import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")
result = emp ['ename'].apply(lambda x:x[0]=='S')
result2 = emp[ ['ename'] ] [result==True]
print(result2)
      ename
10  SMITH
11  SCOTT
```

```
S C O T T
0 1 2 3 4
```

• lambda 표현식

여러줄의 코드를 딱 한줄로 만들어주는 인자

ex)

```
def hap(x,y) :
    return x+y
```

```
print(hap(10,20) )
```

위의 코드를 lambda로

```
print( (lambda x,y : x+y) (10,20) )
```

문제58. 이름의 두번째 철자가 M인 직원들의 이름을 pandas로 출력하시오.


```

답)
import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")
result = emp['ename'].apply(lambda x:x[1]=='M')
result2 = emp[['ename']][result==True]
print(result2)
      ename
10  SMITH

```

25. 스퀀스 자료형 이해하기

```

strdata='Time is money!!' #문자열 변수
listdata=[1,2,[1,2,3] ] #리스트 변수
                # 리스트 안에 리스트가 있는 경우
print(strdata[5])  # 'i'가 출력됨
print(strdata[-2]) # '-i'가 출력됨

print(listdata[0])  # 1이 출력됨
print(listdata[-2]) #[1,2,3]이 출력됨
print(listdata[2][-1]) # 3이 출력됨

```

문제59. 이름의 끝글자가 T로 끝나는 직원들의 이름을 출력하시오.

답1) pandas 사용 x

```

import csv
file = open("C:/python_data/emp2.csv",'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    if (emp_list[1][-1])=='T':
        print(emp_list[1])

SCOTT

```

답2) pandas 사용 o

```

import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")
result = emp ['ename'].apply(lambda x:x[-1]=='T')
result2 = emp[ ['ename'] ] [result==True]
print(result2)

```

	ename
11	SCOTT

파이썬 149제(시퀀스)

2018년 7월 17일 화요일 오전 9:52

26. 시퀀스 자료 슬라이싱 이해하기

```
ex)
strdata = 'Time is money!!'
print(strdata[1:5]) # 'ime'가 출력됨
print(strdata[:7]) # 'Time is'가 출력됨
print(strdata[9:]) # 'oney!!'가 출력됨
print(strdata[:-3]) # 'Time is mone'이 출력됨
print(strdata[-3:]) # 'y!!'이 출력됨
print(strdata[:]) # 'Time is money!!'가 출력됨
print(strdata[::2]) # 'Tm smny!'가 출력됨 (0,2,4, ... 출력)
```

문제60. 이름을 출력하고 이름의 첫번째 철자부터 세번째 철자까지 아래와 같이 출력되게 하시오.

```
보기)
SMITH SMI
ALLEN ALL
FORD FOR
```

```
답1)
import csv
file = open("C:/python_data/emp2.csv",'r')
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
    print(emp_list[1],emp_list[1][0:3])
```

```
답2)
import pandas as pd

emp=pd.read_csv("C:/python_data/emp.csv")
result=emp['ename']
print(emp.ename.apply(lambda x: x[:3]))
```

```

0    KIN
1    BLA
2    CLA
3    JON
4    MAR
5    ALL
6    TUR
7    JAM
8    WAR
9    FOR
10   SMI
11   SCO
12   ADA
13   MIL
Name: ename, dtype: object

```

27. 시퀀스 자료 연결하기 (+)

```

ex)
strdata1 = 'I love '; strdata2 = 'Python'; strdata3 = 'you'
listdata1 = [1, 2, 3]; listdata2 = [4, 5, 6]
print(strdata1 + strdata2)    # 'I love Python'이 출력됨
print(strdata1 + strdata3)    # 'I love you'가 출력됨
print(listdata1 + listdata2)  # [1, 2, 3, 4, 5, 6]이 출력됨

```

문제61. 사원 테이블에 이름을 출력하는데 이름의 첫글자는 대문자로 출력하고 나머지 문자는

소문자로 출력하시오.(대문자 : upper(), 소문자:lower())

```

답)
import csv
file = open("C:/python_data/emp2.csv",'r')
emp_csv = csv.reader(file)

for emp_list in emp_csv:
    print(emp_list[1][0].upper()+emp_list[1][1:].lower())

```

이런 방법도 있지만

```

for emp_list in emp_csv:
    print(emp_list[1].capitalize())

```

28. 시퀀스 자료 반복 이해하기(*)

```

ex)
artist = '빅뱅'
sing = '뱅~'
dispdata = artist + '이 부르는 ' + sing*3

```

```
print(dispdata) # '빅뱅이 부르는 뱅~뱅~뱅~'이 출력됨
```

문제62. 숫자를 물어보게하고 숫자를 입력하면 해당 숫자만큼 ★이 출력되게 하시오.

보기)

숫자를 입력하세요~ 7

답)

```
num=int(input('input is num'))
```

```
for i in range(1,num+1):
```

```
    print ('★'*i)
```

```
input is num7
```

```
★
```

```
★★
```

```
★★★
```

```
★★★★
```

```
★★★★★
```

```
★★★★★★
```

```
★★★★★★★
```

```
★★★★★★★★
```

29. 시퀀스 자료 크기 이해하기(len)

```
strdata1 = 'I love python'
```

```
strdata2 = '나는 파이썬을 사랑합니다'
```

```
listdata = ['a', 'b', 'c', strdata1, strdata2]
```

```
print(len(strdata1)) # 13이 출력됨
```

```
print(len(strdata2)) # 13이 출력됨
```

```
print(len(listdata)) # 5가 출력됨
```

```
print(len(listdata[3])) # 13이 출력됨
```

문제63. 이름과 이름의 철자의 개수를 아래와 같이 출력하시오.

보기)

```
SMITH 5
```

```
SCOTT 5
```

```
ALLEN 5
```

답)

```
import csv
```

```
file = open("C:/python_data/emp2.csv",'r')
```

```
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
```

```
    print(emp_list[1], len(emp_list[1]))
```

KING 4
BLAKE 5
CLARK 5
JONES 5
MARTIN 6
ALLEN 5
TURNER 6
JAMES 5
WARD 4
FORD 4
SMITH 5
SCOTT 5
ADAMS 5
MILLER 6

파이썬 149제(문자열)

2018년 7월 17일 화요일 오전 9:53

30. 문자열의 멤버 체크 이해하기 (in)

문제64. 겨울왕국 대본에 elsa가 몇번 나오는지 확인하기 위해서 겨울왕국 스크립트 한라인을 각각 list 변수에 담으시오.

답)

```
file = open("C:/python_data/winter.txt", 'r')
```

```
for winter_list in file:
```

```
    a=winter_list.split(' ')
```

```
    print(a)
```

```
['', '"Final', 'Shooting', 'Draft\n']  
['by\n']  
['Jennifer', 'Lee\n']  
['9/23/13\n']  
['Property', 'of', 'the', 'Walt', 'Disney', 'Animation', 'Studios  
['OPEN', 'ON:', 'ICE.\n']  
['We' re', 'underwater', 'looking', 'up', 'at', 'it.', 'A', 'saw'  
['heading', 'right', 'for', 'us.\n']  
['EXT.', 'SNOW-CAPPED', 'MOUNTAINS', '?', 'DUSK\n']  
['ICE', 'HARVESTERS', 'dressed', 'in', 'traditional', 'Sami', '<  
['frozen', 'lake.', 'They', 'SING.\n']  
['" The', 'Frozen', 'Heart', '(Ice', 'Worker' s', 'Song)" \n']  
['ICE', 'HARVESTERS\n']  
['BORN', 'OF', 'COLD', 'AND', 'WINTER', 'AIR\n']  
['AND', 'MOUNTAIN', 'RAIN', 'COMBINING,\n']  
['THIS', 'ICY', 'FORCE', 'BOTH', 'FOUL', 'AND', 'FAIR\n']  
['HAS', 'A', 'FROZEN', 'HEART', 'WORTH', 'MINING.\n']  
['The', 'men', 'drag', 'giant', 'ice', 'blocks', 'through', 'char  
['ICE', 'HARVESTERS', '(CONT' D)\n']  
['CUT', 'THROUGH', 'THE', 'HEART,', 'COLD', 'AND', 'CLEAR.\n']  
['STRIKE', 'FOR', 'LOVE', 'AND', 'STRIKE', 'FOR', 'FEAR.\n']  
['SEE', 'THE', 'BEAUTY', 'SHARP', 'AND', 'SHEER.\n']  
['SPLIT', 'THE', 'ICE', 'APART!\n']  
['AND', 'BREAK', 'THE', 'FROZEN', 'HEART.\n']  
['Hup!', 'Ho!', 'Watch', 'your', 'step!', 'Let', 'it', 'go!\n']  
['Hup!', 'Ho!', 'Watch', 'your', 'step!', 'Let', 'it', 'go!', 'Hup!', 'Ho!', 'Watch', 'your', 'step!', 'Let', 'it', 'go!']
```

문제65. 위의 스크립트에 한번더 for loop문을 써서 list변수 안에 있는 단어들을 한 단어씩 출력하시오.

답)

```
file = open("C:/python_data/winter.txt", 'r')
```

```
for winter_list in file:
```

```
    a=winter_list.split(' ')
```

```
for b in a:
    print(b)
```

```
"Final
Shooting
Draft

by

Jennifer
Lee

9/23/13

Property
of
the
Walt
Disney
Animation
Studios

OPEN
ON:
ICE.

We're
underwater
looking
up
at
it.
A
saw
cuts
through,
```

설명: text -----> list 변수 -----> 하나의 단어

	for loop	for loop	
script	['OPEN', 'ON:', 'ICE']		OPEN
			ON:
			ICE

문제66. 위의 단어들이 소문자로 출력되게 하시오.

답)

```
file = open("C:/python_data/winter.txt", 'r')
```

```
for winter_list in file:
    a=winter_list.split(' ')
    for b in a:
        print(b.lower())
```



```
"final  
shooting  
draft  
  
by  
  
jennifer  
lee  
  
9/23/13  
  
property  
of  
the  
walt  
disney  
animation  
studios
```

문제67. 위의 스크립트에 `count` 함수를 사용해서 `elsa`가 나오면 1로 카운트 되게 하시오.

답)

```
file = open("C:/python_data/winter.txt", 'r')
```

```
for winter_list in file:
```

```
    a=winter_list.split(' ')
```

```
    for b in a:
```

```
        print(b.lower(), b.lower().count('elsa'))
```

```
air.  
0  
young 0  
anna 0  
(cont'd)  
0  
catch 0  
me!  
0  
elsa 1  
makes 0  
another 0
```

문제68. 그러면 겨울왕국에는 `elsa`가 몇번 나오는가?

답)

```
file = open("C:/python_data/winter.txt", 'r')
```

```
sum=0
```

```
for winter_list in file:
```

```
    a=winter_list.split(' ')
```

```
    for b in a:
```

```
        sum+=b.lower().count('elsa')
```

```
print(sum)
```

문제69. 겨울왕국에는 anna라는 단어는 몇번 나오는가?

답)

```
file = open("C:/python_data/winter.txt", 'r')
```

```
sum=0
```

```
for winter_list in file:
```

```
    a=winter_list.split(' ')
```

```
    for b in a:
```

```
        sum+=b.lower().count('anna')
```

```
print(sum)
```

```
685
```

문제70. 영어 성경에는 jesus라는 단어가 몇번나오는가?

답)

```
file = open("C:/python_data/NIV.txt", 'r')
```

```
sum=0
```

```
for niv_list in file:
```

```
    a=niv_list.split(' ')
```

```
    for b in a:
```

```
        sum+=b.lower().count('jesus')
```

```
print(sum)
```

```
1275
```

31. 문자열 이해하기

ex)

```
data1='i am python programer'
```

```
data2="you are a programer"
```

```
data3=""" i love
```

```
    i love
```

```
    i love """
```

```
data4="my son's names is john" # 문자열에 싱글을 나타내고 싶으면
```

```
data5='name of my son is "john"' # 문자열에 더블을 나타내고 싶으면
```

문제71. 문자열 안에 싱글과 더블 쿼테이션 마크를 다 넣고 싶을때 어떻게 해야하는가?

답)

```
data6="" my son's names is "john" ""
print(data6)
```

```
dat="my son's name is ₩"John₩""
print(dat)
```

문제72. 아래의 data1~data5 변수에 있는 문자열값을 한번에 출력하시오.

보기)

```
data1='i am python programer'
data2="you are a programer"
data3="" i love
    i love
    i love ""
```

```
data4="my son's names is john" # 문자열에 싱글을 나타내고 싶으면
data5='name of my son is "john"' # 문자열에 더블을 나타내고 싶으면
```

답)

```
print_string = ""
for i in locals():
    if 'data' in i:
        print_string += eval(i) + '\n'

print (print_string)
```

설명 : locals() : 지역변수의 이름과 값을 확인하는 함수
globals() : 전역변수의 이름과 값을 확인하는 함수

답2)

```
data1='i am python programer'
data2="you are a programer"
data3="" i love
    i love
    i love ""
```

```
data4="my son's names is john"
data5='name of my son is "john"'
```

```
print_string = ""
for i in range(1,6):
```

```
dt="data"+str(i)
print(locals()[dt])
```

32. 문자열 포매팅 이해하기

: 변하는 값을 포함하는 문자열을 표현하기 위해 하나의 양식으로 문자열을 만드는것.

ex)

%s : 문자열

%d : 정수

%f : 실수

%% : %기호

ex)

```
txt1 = '자바';txt2='파이썬'
```

```
num1= 5; num2=10
```

```
print('나는 %s보다 %s에 더 익숙합니다.' %(txt1, txt2))
```

```
print('%s은 %s보다 %d배 더 쉽습니다.' %(txt2, txt1, num1))
```

```
print('%d + %d = %d' %(num1, num2, num1+num2))
```

```
print('작년 세계 경제 성장률은 전년에 비해 %d%% 포인트 증가했다.' %num1)
```

문제73. 변하는 값을 포함하는 문자열을 나타내는 아래의 코드를 실행하시오.

답)

```
from time import sleep    #time이라는 모듈안에 sleep함수 임포트
```

```
for i in range(100):
```

```
    msg = 'Wr진행률 %d%%' %(100-i-1)
```

```
    print(msg, end="")
```

```
    sleep(0.1)
```

설명 : Wr 리턴 문자를 추가하여 msg변수를 print 할때 줄바꿈 없이 화면에 출력한다.

문제74. 겨울왕국 대본에서 anna를 찾는 for loop문에 위의 코드를 입혀서 아래와 같이 출력되게 하시오.

보기)

'1' 개 anna를 찾으면 숫자가 변경되게 하시오.

답)

```
from time import sleep
```

```
file = open("C:/python_data/winter.txt", 'r')
```

```
sum=0
```

```
for winter_list in file:
```

```

a=winter_list.split(' ')
for b in a:
    sum+=b.lower().count('anna')
    msg = 'Wr 찾은 개수 : %d' %(sum)
    print(msg, end="")
    sleep(0.00001)

```

찾은 개수 : 685

문제75. 단어를 물어보게하고 단어를 입력하면 해당 단어의 개수를 출력하는 프로그램으로 위의 코드를 수정하시오.

답)

```

from time import sleep
file = open("C:/python_data/winter.txt", 'r')

```

```

word=str( input('찾을 단어 입력 : ') )

```

```

sum=0
for winter_list in file:
    a=winter_list.split(' ')
    for b in a:
        sum+=b.lower().count(word)
        msg = 'Wr 찾은 개수 : %d' %(sum)
        print(msg, end="")
        sleep(0.00001)

```

문제76. (마지막문제) 위의 코드를 수정해서 스크립트로 물어보게 하시오.

보기)

분석할 스크립트를 입력하세요 ~ winter.txt

찾을 단어를 입력하세요~ elsa

385개

답)

```

file_name=str( input('분석할 스크립트를 입력 : ') )

```

```

file = open("C:/python_data/"+file_name, 'r')

```

```

word=str( input('찾을 단어 입력 : ') )

```

```

from time import sleep

```

```

sum=0
for winter_list in file:
    a=winter_list.split(' ')
    for b in a:
        sum+=b.lower().count(word)
        msg = 'Wr 찾은 개수 : %d' %(sum)
        print(msg, end="")
        sleep(0.00001)

```

분석할 스크립트를 입력 : winter.txt
 찾을 단어 입력 : elsa
 찾은 개수 : 329

33. 이스케이프 문자 이해하기

이스케이프 문자

코드	설명
\n	개행 (줄바꿈)
\t	수평 탭
\\	문자 "\"
\'	단일 인용부호(')
\"	이중 인용부호(")
\r	캐리지 리턴
\f	폼 피드
\a	벨 소리
\b	백 스페이스
\000	널문자

ex)

```

print('나는 파이썬을 사랑합니다.\n파이썬은 자바보다 훨씬 쉽습니다.')
print('Name: John Smith\tSex: Male\tAge: 22')
print('이 문장은 화면폭에 비해 너무 길어 보기가 힘듭니다. \\\n')
그래서 \\Enter키를 이용해 문장을 다음줄과 연속되도록 했습니다.')
print('작은따옴표(')와 큰 따옴표(")는 문자열을 정의할 때 사용합니다.')

```

```

나는 파이썬을 사랑합니다.
파이썬은 자바보다 훨씬 쉽습니다.
Name: John Smith      Sex: Male      Age: 22
이 문장은 화면폭에 비해 너무 길어 보기가 힘듭니다. 그래서 \\Enter키를 이용해 문장을 다음줄과 연속되도록 했습니다.
작은따옴표(')와 큰 따옴표(")는 문자열을 정의할 때 사용합니다.

```

문제77. 아래의 문장을 파이썬에서 그래도 출력하시오.

보기)

```
"When I was young my family was pool.  
My mother doesn't like Chinese noodles.  
Ya-hee ~, Ya-hee~, Ya "
```

답)

```
print(" When i was young, my family was pool. When My mother doesn't like Chinese  
noodle. When Ya-hee ~, Ya-hee ~, Ya ")
```

```
"When i was young, my family was pool.  
My mother doesn't like Chinese noodles.  
Ya-hee ~, Ya-hee ~, Ya "
```

34. 리스트 변수란?

: 데이터 목록을 다루는 자료형

대괄호 [] 안에 데이터를 입력해서 관리하는 변수

- 리스트 변수의 유용한 메소드 함수

■ 리스트 함수

함수명	설명	예시
append	#리스트에 요소 추가 append(x)는 리스트의 맨 마지막에 x를 추가시키는 함수	<pre>>>> a = [1, 2, 3] >>> a.append(4) >>> a [1, 2, 3, 4]</pre>
sort	#리스트 정렬 sort 함수는 리스트의 요소를 순서대로 정렬	<pre>>>> a = [1, 4, 3, 2] >>> a.sort() >>> a [1, 2, 3, 4]</pre>
reverse	#리스트 뒤집기 reverse 함수는 리스트를 역순으로 뒤집어 준다. 이때 리스트 요소들을 순서대로 정렬한 다음 다시 역순으로 정렬하는 것이 아니라 그저 현재의 리스트를 그대로 거꾸로 뒤집을 뿐이다.	<pre>>>> a = ['a', 'c', 'b'] >>> a.reverse() >>> a ['b', 'c', 'a']</pre>
index	#위치 반환 index(x) 함수는 리스트에 x라는 값이 있으면 x의 위치값을 리턴. 리스트에 존재하지 않으면 값 오류(ValueError)가 발생한다.	<pre>>>> a = [1,2,3] >>> a.index(3) 2 >>> a.index(1) 0</pre>
insert	#리스트에 요소 삽입 insert(a, b)는 리스트의 a번째 위치에 b를 삽입하는 함수	<pre>>>> a = [1, 2, 3] >>> a.insert(0, 4) [4, 1, 2, 3]</pre>
remove	#리스트 요소 제거 remove(x)는 리스트에서 첫 번째로 나오는 x를 삭제하는 함수	<pre>>>> a = [1, 2, 3, 1, 2, 3] >>> a.remove(3) [1, 2, 1, 2, 3]</pre>
pop	#리스트 요소 끄집어내기 pop()은 리스트의 맨 마지막 요소를 돌려 주고 그 요소는 삭제하는 함수	<pre>>>> a = [1,2,3] >>> a.pop() 3 >>> a [1, 2]</pre>
count	#리스트에 포함된 요소 x의 개수 세기 count(x)는 리스트 내에 x가 몇 개 있는지 조사하여 그 개수를 돌려주는 함수	<pre>>>> a = [1,2,3,1] >>> a.count(1) 2</pre>
extend	#리스트 확장 extend(x)에서 x에는 리스트만 올 수 있으며 원래의 a 리스트에 x 리스트를 더하게 된다.	<pre>>>> a = [1,2,3] >>> a.extend([4,5]) >>> a [1, 2, 3, 4, 5] >>> b = [6, 7] >>> a.extend(b) >>> a [1, 2, 3, 4, 5, 6, 7]</pre>

문제78. emp_list에서 b 요소를 제거하시오.

보기)

```
emp_list=[]
emp_list.append('b')
print(emp_list)
```

답)

```
emp_list=[]
```



```
emp_list.append('b')
emp_list.remove('b')
print(emp_list)
```

문제79. emp.csv의 sal 의 값을 sal_list라는 비어있는 리스트 변수에 담으시오.(for loop문)

답)

```
import csv
file = open("C:/python_data/emp2.csv", 'r')
emp_csv= csv.reader(file)
```

```
sal_list = []
```

```
for emp_list in emp_csv:
    sal_list.append( int(emp_list[5]) )
print(sal_list)
```

```
[5000, 2850, 2450, 2975, 1250, 1600, 1500, 950, 1250, 3000, 800, 3000, 1100, 1300]
```

(웹 스크롤링 뼈대 코드)

문제80. 위의 스크립트에 for loop문을 아래에 추가해서 아래의 리스트의 요소를 하나씩 출력 하시오.

보기)

```
[5000, 2850, 2450, 2975, 1250, 1600, 1500, 950, 1250, 3000, 800, 3000, 1100, 1300]
```

답)

```
for emp_list in emp_csv:
    sal_list.append( int(emp_list[5]) )
```

```
for i in sal_list:
```

```
    print(i)
```

```
5000
2850
2450
2975
1250
1600
1500
950
1250
3000
800
3000
1100
1300
```

답2)

```
while sal_list:  
    print(sal_list.pop())
```

문제81. 위의 결과를 다시 출력하는데 월급이 낮은 직원부터 출력하시오.

답)

```
import csv  
file = open("C:/python_data/emp2.csv", 'r')  
emp_csv= csv.reader(file)
```

```
sal_list = []
```

```
for emp_list in emp_csv:  
    sal_list.append( int(emp_list[5]) )
```

```
sal_list.sort()
```

```
for i in sal_list:  
    print(i)
```

문제82. 위의 결과를 월급이 높은 것부터 출력하시오.

답)

```
import csv  
file = open("C:/python_data/emp2.csv", 'r')  
emp_csv= csv.reader(file)
```

```
sal_list = []
```

```
for emp_list in emp_csv:  
    sal_list.append( int(emp_list[5]) )
```

```
sal_list.sort()
```

```
sal_list.reverse()
```

```
for i in sal_list:  
    print(i)
```

문제83. 이름과 월급을 출력하는데 월급이 높은 직원부터 출력하시오.

답1) pandas 이용 안했을때

```
def colcheck(data):
```

```

    return int(data[5])

import csv
file = open("C:/python_data/emp2.csv", 'r')
emp_csv= csv.reader(file)

emp_list = []

for i in emp_csv:
    emp_list.append(i)

emp_sort_list=sorted(emp_list, reverse=True, key=colcheck)
#함수를 실행할때 함수이름(매개변수)를 써줘야하는데 여기서는 key=함수이름 이렇게
사용가능하다

for i in emp_sort_list:
    print(i[1],i[5])

KING 5000
FORD 3000
SCOTT 3000
JONES 2975
BLAKE 2850
CLARK 2450
ALLEN 1600
TURNER 1500
MILLER 1300
MARTIN 1250
WARD 1250
ADAMS 1100
JAMES 950
SMITH 800

```

답2) pandas 이용 했을때

```

import pandas as pd
file=open("C:/python_data/emp2.csv")
file=pd.read_csv(file, header=None)
emp=file.sort_values(5, ascending=False)
print(emp[ [1,5] ] )   # 컬럼명이 숫자여서 이렇게 써줘야한다.

```

	1	5
0	KING	5000
9	FORD	3000
11	SCOTT	3000
3	JONES	2975
1	BLAKE	2850
2	CLARK	2450
5	ALLEN	1600
6	TURNER	1500
13	MILLER	1300
4	MARTIN	1250
8	WARD	1250
12	ADAMS	1100
7	JAMES	950
10	SMITH	800

문제84. 이름과 입사일을 출력하는데 최근에 입사한 사원부터 출력하시오.

답1)

```
def colcheck(data):
```

```
    return data[4]
```

```
import csv
```

```
file = open("C:/python_data/emp2.csv", 'r')
```

```
emp_csv= csv.reader(file)
```

```
emp_list = []
```

```
for i in emp_csv:
```

```
    emp_list.append(i)
```

```
emp_sort_list=sorted(emp_list, reverse=True, key=colcheck)
```

```
for i in emp_sort_list:
```

```
    print(i[1],i[4])
```

```
ADAMS 1983-01-15
```

```
SCOTT 1982-12-22
```

```
MILLER 1982-01-11
```

```
JAMES 1981-12-11
```

```
FORD 1981-12-11
```

```
KING 1981-11-17
```

```
MARTIN 1981-09-10
```

```
TURNER 1981-08-21
```

```
CLARK 1981-05-09
```

```
BLAKE 1981-05-01
```

```
JONES 1981-04-01
```

```
WARD 1981-02-23
```

```
ALLEN 1981-02-11
```

```
SMITH 1980-12-09
```

답2)

```
import pandas as pd
file=open("C:/python_data/emp2.csv")
file=pd.read_csv(file, header=None)
emp=file.sort_values(4, ascending=False)
print(emp[ [1,4] ] )
```

```
      1      4
12  ADAMS 1983-01-15
11  SCOTT 1982-12-22
13  MILLER 1982-01-11
7   JAMES 1981-12-11
9   FORD 1981-12-11
0   KING 1981-11-17
4   MARTIN 1981-09-10
6   TURNER 1981-08-21
2   CLARK 1981-05-09
1   BLAKE 1981-05-01
3   JONES 1981-04-01
8   WARD 1981-02-23
5   ALLEN 1981-02-11
10  SMITH 1980-12-09
```

35. 파이썬과 오라클 그룹함수 비교

오라클	vs	파이썬
max		max
min		min
count		count
sum		sum
avg		사용자 정의 함수

문제85. 사원 테이블에서 최대 월급을 출력하시오.

답1) 판다스x

```
import csv
file = open("C:/python_data/emp2.csv", 'r')
emp_csv= csv.reader(file)

sal_list = []

for emp_list in emp_csv:
    sal_list.append( int(emp_list[5]) )
print(max(sal_list) )
5000
```

답2) 판다스o

```
import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")

result=max(emp['sal'])
print(result)
5000
```

문제86. 사원 테이블에서 토달월급을 출력하시오.

답1) 판다스x

```
import csv
file = open("C:/python_data/emp2.csv", 'r')
emp_csv= csv.reader(file)
```

```
sal_list = []
```

```
for emp_list in emp_csv:
    sal_list.append( int(emp_list[5]) )
print(sum(sal_list) )
```

답2) 판다스o

```
import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")

result=(sum(emp['sal']) )
print(result)
```

문제87. 평균월급을 출력하시오.

답1) 판다스x

```
def avg(data):
    return(sum(data)/len(data))
```

```
import csv
file = open("C:/python_data/emp2.csv", 'r')
emp_csv = csv.reader(file)
sal_list = []
```

```
for emp_list in emp_csv:
    sal_list.append(int(emp_list[5]))
```

```
print(avg(sal_list))
```

답2) 판다스o

```
import pandas as pd  
emp=pd.read_csv("C:/python_data/emp.csv")
```

```
result = sum(emp['sal'])/len(emp['sal'])  
print("%0.2f" %result)
```

판다스2)

```
import pandas as pd  
emp=pd.read_csv("C:/python_data/emp.csv")
```

```
result = emp['sal'].mean()  
print("%0.2f" %result)
```

문제88. 직업이 SALESMAN인 직원들의 최대월급을 출력하시오.

답1) 판다스 x

```
def avg(data):  
    return(sum(data)/len(data))  
  
import csv  
file = open("C:/python_data/emp2.csv", 'r')  
emp_csv = csv.reader(file)  
sal_list = []  
  
for emp_list in emp_csv:  
    if emp_list[2] == 'SALESMAN':  
        sal_list.append(int(emp_list[5]))  
  
print(max(sal_list))
```

답2) 판다스 o

```
import pandas as pd  
emp=pd.read_csv("C:/python_data/emp.csv")  
  
result =[emp['sal'][emp['job']=='SALESMAN'].max()]  
print(result)
```

문제89. (점심시간) 직업, 직업별 최대월급을 출력하시오.

보기)

```
SQL> select job, max(sal)
      from emp
      group by job;
```

답1) 판다스 o

```
import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")
result=emp.groupby('job')['sal'].max()
print(result)
```

job	
ANALYST	3000
CLERK	1300
MANAGER	2975
PRESIDENT	5000
SALESMAN	1600

답2) 판다스 x



```
import csv
file = open("C:/python_data/emp2.csv", 'r')
emp=list(csv.reader(file))
job=[]

for i in range(len(emp)):
    job.append((emp[i][2])) # 직업만 입력한다.
job = list(set(job)) # 중복을 제거
for i in job:
    sal=[]
    for j in range(len(emp)):
        if emp[j][2]== i:
            sal.append(int(emp[j][5]))
    print(i,max(sal))
```

문제90. 부서번호, 부서번호별 토탈월급을 출력하시오.

답1) 판다스o

```
import pandas as pd
emp=pd.read_csv("C:/python_data/emp.csv")
result=emp.groupby('deptno')['sal'].sum()
print(result)
```

답2) 판다스x

```
import csv
```



```

file = open("C:/python_data/emp2.csv", 'r')
emp=list(csv.reader(file))
job=[]

for i in range(len(emp)):
    job.append((emp[i][7])) # 부서번호만 입력한다.

job = list(set(job)) # 중복을 제거

for i in job:
    sal=[]
    for j in range(len(emp)):
        if emp[j][7]== i:
            sal.append(int(emp[j][5]))
    print(i,sum(sal))

```

문제91. 위의 결과를 다시출력하는데 토탈월급이 높은 것부터 출력하시오.(생각할 문제)

답1) 판다스o

답2) 판다스x

36. 튜플 이해하기

: 튜플(tuple) 변수란? 리스트 변수와는 다르게 요소 데이터의 변경이 불가능한 자료형
 리스트는 대괄호 [] 로 요소들을 둘러는데 튜플은 소괄호 () 로 요소를 감싼다.
 " 변경이 안되므로 튜플로 만든 데이터에 대한 신뢰도가 높아진다. "

```

ex)
tuple1 = (1, 2, 3, 4, 5)
tuple2 = ('a', 'b', 'c')
tuple3 = (1, 'a', 'abc', [1, 2, 3, 4, 5], ['a', 'b', 'c'])
#tuple1[0] = 6 에러뜸!!

```

```

def myfunc():
    print('안녕하세요')

```

```

tuple4 = (1, 2, myfunc)
tuple4[2]()      # '안녕하세요' 가 출력됨

```

37. 사전 이해하기

딕셔너리 자료형이란?

딕셔너리 자료형은 key와 value를 조합해서 사용하는 자료형

ex)

```
dic = {}
```

```
dic['파이썬'] = 'www.python.org'
```

```
print(dic)
```

```
print(dic.keys())
```

```
print(dic.values())
```

```
{'파이썬': 'www.python.org'}
```

```
dict_keys(['파이썬'])
```

```
dict_values(['www.python.org'])
```

문제92. dic자료형 변수에 키를 애플로하고 값을 www.apple.com 으로 해서 추가하시오.

답)

```
dic = {}
```

```
dic['파이썬'] = 'www.python.org'
```

```
print(dic)
```

```
print(dic.keys())
```

```
print(dic.values())
```

```
print(dic)
```

```
dic['애플']='www.apple.com'
```

```
print(dic)
```

```
{'파이썬': 'www.python.org'}
```

```
dict_keys(['파이썬'])
```

```
dict_values(['www.python.org'])
```

```
{'파이썬': 'www.python.org'}
```

```
{'파이썬': 'www.python.org', '애플': 'www.apple.com'}
```

문제93. dic 자료형 변수에 키를 마이크로 소프트, 값을 www.microsoft.com으로 추가하시오.

답)

```
print(dic)
```

```
dic['마이크로 소프트']='www.microsoft.com'
```

```
print(dic)
```

```
{'파이썬': 'www.python.org', '애플': 'www.apple.com'}
```

```
{'파이썬': 'www.python.org', '애플': 'www.apple.com', '마이크로 소프트': 'www.microsoft.com'}
```

문제94. dic자료형 변수의 요소중에 애플을 삭제하시오.

```
답)
dic.pop('애플')
print(dic)
{'파이썬': 'www.python.org', '마이크로 소프트': 'www.microsoft.com'}
```

문제95. dic 자료형 변수의 모든 요소를 삭제하시오.

```
답)
dic.clear()
print(dic)
{}
```

문제96. 딕셔너리 자료형을 이용해서 주어가 0, 동사가 1, 명사가 2 로 해서 한글과 영문을 저장하시오.

```
답)
dic={}
dic['나는']='(I',0)
dic['소년']='(boy',2)
dic['이다']='(am',1)
dic['피자']='(pizza',2)
dic['먹는다']='(eat',1)
print(dic)
```

문제97. 아래와 같이 번역할 한글을 물어보게하고 한글을 입력하면 영어로 번역하는 프로그램을 파이썬으로 작성하시오.

```
보기)
번역할 한글을 입력하세요~ 나는 소년 이다.
결과)
I am boy
```

```
답)
dic={}
dic['나는']='(I',0)
dic['소년']='(boy',2)
dic['이다']='(am',1)
dic['피자']='(pizza',2)
dic['먹는다']='(eat',1)
```

```
res=""
input_kor = input('번역할 한글을 입력 : ')
input_list=input_kor.split(' ')
```

```
print(input_list)

for i in range(len(input_list)):
    for j in input_list:
        if dic[j][1]==i:
            res=res+dic[j][0]+' '
print(res)
번역할 한글을 입력 : 나는 소년 이다
['나는', '소년', '이다']
I am boy
```

파이썬 149제(함수)

2018년 7월 18일 수요일 오후 4:11

38. 함수 이해하기(def)

:오라클의 내장함수와 파이썬의 내장함수 비교

1. 문자함수

오라클	vs	파이썬
upper		upper()
lower		lower()
initcap		사용자함수
substr		사용자함수
replace		replace()
length		length()
rtrim		rstrip()
ltrim		lstrip()
trim		strip()
rpadd		사용자함수
lpadd		사용자함수
instr		사용자함수

2. 숫자함수

round	vs	round()
trunc		trunc()
mod		%
power		power()

3. 날짜함수

months_between	vs	사용자함수
add_months		relativedelta
next_day		사용자함수
last_day		monthrange

4. 변환함수

to_char	vs	str()
to_number		int(), float()
to_date		datetime.strptime()

5. 일반함수

nvl	vs	사용자함수
decode		사용자함수
case		if문

ex)

```
def add_number(n1,n2):  
    ret=n1+n2  
    return ret
```

```
print(add_number(2,3) )
```

5

문제98. 오라클의 initcap함수를 만들어서 아래와 같이 실행되게 하시오.

보기)

```
print( initcap('smith') )  
Smith
```

답)

```
def initcap(data):  
    return data[0].upper() + data[1:].lower()  
print(initcap('smith'))  
Smith
```

문제99. substr 함수를 파이썬으로 생성하시오.

보기)

```
print (substr('smith', 1,3) )  
smi
```

답)

```
def substr(val,a,b):  
    return val[a-1:a-1+b]
```

```
print(substr('smith',1,3))  
print(substr('smith',2,2))
```

smi
mi

문제100. instr 함수를 파이썬으로 구현하시오.

보기)

```
SQL> select instr('smith', 'm') from dual;  
2  
>>> print( instr('smith','m') )  
2
```

답)

```
def instr(val,a):  
    cnt=0  
    for i in range(len(val)):  
        if val[i] !=a:  
            cnt+=1  
        elif val[i] == a:  
            return(cnt+1)
```

```
print(instr('smith','m'))
```

문제101. 오라클 lpad함수를 파이썬으로 만드시오.

보기)

```
SQL> select lpad('smith', 10, '*')  
        from dual;  
*****smith
```

```
print( lpad('smith',10,'*') )  
*****smith
```

답)

```
def lpad(val,a,b):  
    return (a-len(val))*b + val
```

```
print(lpad('smith',10,'*'))  
*****smith
```

문제102. 오라클 rpad 함수를 파이썬으로 만드시오.

보기)

```
print(rapd('smith',10,'*') )
```

답)

```
def lpad(val,a,b):  
    return val+(a-len(val))*b
```

```
print(lpad('smith',10,'*'))
```

문제103. 오늘 날짜를 출력하시오.

답)

```
from datetime import date
print( date.today() )
2018-07-17
```

문제104. 파이썬으로 add_months함수를 생성하시오.

(오늘부터 3달뒤에 돌아오는 날짜)

보기)

```
from datetime import date
from dateutil.relativedelta import relativedelta

result = date.today() + relativedelta(months=+3)
print(result)
2018-10-17
```

답)

```
def add_months(dat,a):
    import datetime
    from datetime import date
    from dateutil.relativedelta import relativedelta

    dat1=datetime.datetime.strptime(dat, '%Y-%m-%d') #날짜로 형변환 해줘야함!!
    result= dat1+ relativedelta( months=+int(a) )
    return result

print(add_months('2018-07-17', 3))
2018-10-17 00:00:00
```

문제105. 올해 2월달의 마지막 날짜를 출력하시오.

답)

```
from calendar import monthrange
print(monthrange(2018,2))
(3, 28)
```

설명: 3의 의미는 그달의 1일 요일

월 :0, 화:1, 수:2, 목:3, 금:4, 토:5, 일:6

문제106. 위의 결과에서 28만 출력하시오.

```
답)
from calendar import monthrange
a=monthrange(2018,2)
print(a[1])
28
```

문제107. 아래의 날짜에서 년도만 출력하시오.

```
보기)
'2018-07-18'

답)
from datetime import datetime as date
a='2018-07-18'
year=date.strptime(a,'%Y-%m-%d').year
print(year)
2018
```

문제108. 아래의 날짜에서 달을 출력하시오.

```
보기)
'2018-07-18'

답)
from datetime import datetime as date
a='2018-07-18'
month=date.strptime(a,'%Y-%m-%d').month
print(month)
7
```

문제109. 위의 함수들을 사용해서 오라클 last_day함수를 아래와 같이 생성하시오.

```
보기)
print( last_day('2018-07-18') )
2018-07-31

답)
def last_day(date1):
    from calendar import monthrange
    import datetime

    year=datetime.datetime.strptime(date1,'%Y-%m-%d').year
    month=datetime.datetime.strptime(date1,'%Y-%m-%d').month
```

```
return date1[:8] + str(monthrange(year,month)[1])
```

```
print(last_day('2018-07-18'))  
2018-07-31
```

next_day 함수 생성하기

```
SQL> select next_day(sysdate, '금요일') from dual;
```

문제110. 오늘 날짜를 출력하시오.

```
답)  
from datetime import datetime as date  
print(date.today())  
2018-07-18 11:43:54.351315
```

문제111) 오늘 날짜에서 하루 더한 날짜가 어떻게 되는가?

```
답1)  
import datetime as date  
  
print( date.datetime.today()+ datetime.timedelta(days=1) )
```

```
답2)  
from datetime import datetime as date  
import datetime  
  
print( date.today() + datetime.timedelta(days=1) )  
2018-07-19 11:58:13.977483
```

문제112. 아래의 함수를 생성하시오.

```
보기)  
print( next_day('2018-07-18', 2) )  
2018-07-20
```

```
답)  
def next_day(date1,day):  
    import datetime as date  
  
    date2 = datetime.datetime.strptime(date1, '%Y-%m-%d')  
    date3 = date2 + datetime.timedelta(days=int(day))  
    date3 = date3.date()  
    return date3
```

```
print( next_day('2018-07-18', 2) )
```

2018-07-20

문제113. 오늘이 무슨 요일인지 출력하시오.

```
답)
from datetime import datetime as date
#datetime 패키지에 datetime모듈을 사용하겠다
#datetime 을 date로 코드에서 사용하겠다
print(date.today().weekday() )
2
월요일: 0
화요일: 1
...
일요일: 6
```

문제114. 위의 코드를 이용해서 한글로 수요일이라고 출력하시오.

```
답)
from datetime import datetime as date

day_list=['월요일','화요일','수요일','목요일','금요일','토요일','일요일']

a=(date.today().weekday() )
print(day_list[a])
수요일
```

문제115. 위의 코드들을 활용해서 아래의 next_day함수를 구현하시오.

```
답1)
import datetime as date
def next_day(a,b):
    dic={'월요일':0,'화요일':1,'수요일':2,'목요일':3,'금요일':4,'토요일':5,'일요일':6}
    day=date.datetime.strptime( a,'%Y-%m-%d')
    dys=dic[b]-day.weekday()
    if dys <=0:
        dys=dys+7
    return day + date.timedelta(days=dys).

print(next_day('2018-07-18','월요일'))
```

답2)

```
def next_day(val1, val2):
    import datetime as date
    dy={'월요일':0, '화요일':1, '수요일':2, '목요일':3, '금요일':4, '토요일':5, '일요일':6}
    n=dy[val2]
    d = date.datetime.strptime(val1, '%Y-%m-%d').weekday()
    res=(7+(n-d))%7
    return date.datetime.strptime(val1, '%Y-%m-%d').date()+date.timedelta(days=res)

print(next_day('2018-07-18', '월요일'))
```

- **months_between** 함수를 파이썬으로 만들기

ex)

```
import datetime
d1= '2018-07-18'
print(type(d1)) #문자형
a=datetime.datetime.strptime(d1, '%Y-%m-%d')
print(a)
print(type(a)) #날짜형
<class 'str'>
2018-07-18 00:00:00
<class 'datetime.datetime'>

print(a.year)
print(a.month)
2018
7
```

문제116. months_between 함수를 파이썬으로 구현하시오.

보기)

```
print(months_between('2017-01-01', '2018-07-01'))
```

힌트)

```
a=datetime.datetime.strptime(d1, '%Y-%m-%d')
print(a.year)
print(a.month)
```

답)

```
def months_between(a,b):
    aa=datetime.datetime.strptime(a, '%Y-%m-%d')
    bb=datetime.datetime.strptime(b, '%Y-%m-%d')
    return ((bb.year-aa.year)*12)+bb.month-aa.month
```

```
print(months_between('2017-01-01', '2018-07-01'))  
18
```

- **오라클의 일반함수**

1. nvl
2. decode

문제117. 오라클의 nvl함수를 파이썬으로 구현하시오.

보기)

```
SQL> select ename, nvl(comm,0)  
       from emp;  
  
def nvl(val1, val2):  
    if val1 is "":  
        return val2  
    return val1
```

```
file = open("C:/python_data/emp_comm.csv", "r")  
emp_csv = csv.reader(file)  
for emp in emp_csv:  
    print(emp[1], nvl(emp[6], 0))
```

문제118. 오라클의 decode 함수를 파이썬으로 구현하시오. 이름을 출력하고 부서번호가 10번 이면 3000을 출력하고 10번이 아니면 0을 출력하도록 decode함수를 만드시오.

보기)

```
import csv  
file = open("C:/python_data/emp2.csv", "r")  
emp_csv = csv.reader(file)  
for emp_list in emp_csv:  
    print(emp_list[1], decode(emp_list[7], 10, 3000, 0))
```

답)

```
def decode(a,b,c,d):  
    if type(b)==int:  
        a = int(a)  
    if a==b:  
        return c  
    return d
```

```
import csv  
file = open("C:/python_data/emp2.csv", "r")
```

```
emp_csv = csv.reader(file)
```

```
for emp_list in emp_csv:
```

```
    print(emp_list[1],decode(emp_list[7],10,3000,0))
```

```
KING 3000  
BLAKE 0  
CLARK 3000  
JONES 0  
MARTIN 0  
ALLEN 0  
TURNER 0  
JAMES 0  
WARD 0  
FORD 0  
SMITH 0  
SCOTT 0  
ADAMS 0  
MILLER 3000
```

```
[print(emp_list[1],decode(emp_list[7],10,3000,0)) for emp_list in emp_csv]
```

이렇게 포문을 한줄로 사용 할 수도 있다.

39. 함수 인자(매개변수) 이해하기

ex)

```
def add_txt(t1, t2='파이썬'):
```

```
    print(t1+' : ' +t2)
```

```
add_txt('베스트')           # '베스트 : 파이썬'이 출력됨
```

```
add_txt(t2='대한민국', t1='1등')  # '1등 : 대한민국'이 출력됨
```

```
def func1(*args):
```

```
    print(args)
```

```
def func2(width, height, **kwargs):
```

```
    print(kwargs)
```

```
func1()           # 빈 튜플 ( ) 이 출력됨
```

```
func1(3, 5, 1, 5)      # (3, 5, 1, 5)가 출력됨
```

```
func2(10, 20)         # 빈 사전 { }이 출력됨
```

```
func2(10, 20, depth=50, color='blue')  #{'depth':50, 'color':'blue'} 이 출력됨
```

가변 매개변수?

문자열.format() 함수처럼 매개변수의 수가 유동적인 함수를 만들고 싶을때 사용하는 변수.

함수를 실행할때 매개변수를 10개, 20개를 입력해도 제대로 동작한다.

문제119. 가변 매개변수를 이용해서 아래의 함수를 실행하시오.

보기)

```
print( merge_string('아버지가', '방에', '들어가실려다가', '부엌으로', '가셔서',₩  
                '짜장면이', '싫다고', '하셨어요') )
```

답)

```
def merge_string(*args):
```

```
    res=""
```

```
    for i in args:
```

```
        res=res+i+' '
```

```
    return res
```

```
print( merge_string('아버지가', '방에', '들어가실려다가', '부엌으로', '가셔서',₩  
                '짜장면이', '싫다고', '하셨어요') )
```

아버지가 방에 들어가실려다가 부엌으로 가셔서 짜장면이 싫다고 하셨습니다

문제120. 두수를 입력하고 함수를 실행하면 최대 공약수가 출력되게 하시오.

보기)

```
maxyaksu(16,24)
```

답)

```
def maxyaksu(a,b):
```

```
    if a > b:
```

```
        (a,b)=(b,a)
```

```
    while b>0:
```

```
        remain=a%b
```

```
        a=b
```

```
        b=remain
```

```
    return a
```

```
print(maxyaksu(16,24))
```

문제121. 가변 매개변수를 사용해서 아래와 같이 실행해도 최대 공약수가 나오게 하시오.

보기)

```
maxyaksu(16,24,8)
```

힌트)

```
def maxyaksu(*num)
```

```
    num=list(num)
```

답)

```
def maxyaksu(*num):
```

```
    num=list(num)
```

```

num.sort()
num.reverse()

for i in range(len(num)-1):
    while True:
        remain=num[i]%num[i+1]
        if remain==0:
            break
        num[i], num[i+1] = num[i+1], remain

return num[-1]

print(maxyaksu(16,24,8,4))
4

```

문제122. 0부터 1사이의 숫자를 random으로 생성해보시오.

답)

```

import random
x=random.uniform(0,1)
print(x)

```

0.7099771601081608

문제123. 위의 코드를 for loop문을 이용해서 10000번 반복시키시오.

답)

```

import random
for i in range(10000):
    x=random.uniform(0,1)
    y=random.uniform(0,1)
    print(x)
    print(y)

```



```
0.3456498998499846
0.8616311457616956
0.4030402321475961
0.37873401520165995
0.7034348367254555
0.3680878654825869
0.5962154061865973
0.10843728394127594
0.17254257000574136
0.04634697452957082
0.016830223986386783
0.946190024039383
0.8405662198765177
0.09136311495798777
0.6283413944992429
0.672786095551822
0.4772670731340275
0.44879230851685836
0.5025833075236954
0.22745482742326406
```

문제124. $x^2 + y^2 < 1$ 이면, 부채꼴 안에 들어가는 점이므로 이공식을 이용해서 pi를 구하시오.

답)

```
import random
cnt=0
for i in range(10000):
    x=random.uniform(0,1)
    y=random.uniform(0,1)
    if (x**2 + y**2) < 1 :
        cnt+=1
print(cnt/10000*4)
```

설명)

"몬테 카를로 알고리즘"

수많은 노가다를 통해서 답을 알아내는 것

1. 강화학습 : 컴퓨터가 수많은 게임을 하면서 게임의 방법을 스스로 알아내는 방법.
2. 데이터 분석 : 6기 조은성 학생이 지금 강남 삼성생명에서 데이터 분석 프로젝트를 하는데 몬테카를로 알고리즘으로 수행한다.

삼성생명에서 금융사고가 났을 때를 대비해서 평상시에 가지고 있어야 할 자금이 어떻게 되는지 컴퓨터로 수많은 금융 사고를 랜덤으로 일으켜서 대비해야할 금액을 알아냄.

40.지역변수와 전역변수 이해하기

41. 함수 리턴값 이해하기(return)

: 모든 함수는 이름을 갖고 있다. 이 이름을 불러주면 파이썬은 그 이름 아래 정의되어 있는 코드를 실행한다.

이때 **함수를 부르는 행위를 호출**이라고 하고 **함수를 부르는 코드를 호출자**라고 한다.

함수가 호출자에게 결과를 돌려주는 것을 반환(return)이라고 한다.

```
ex)
def initcap(val):
    return val[0].upper()+val[1:].lower()
print(initcap('smith'))
Smith
```

문제125. my_abs라는 함수를 생성하시오.(무조건 양수로 출력하는 함수)

```
보기)
print(my_abs(9))
9
print(my_abs(-9))
9
```

```
답)
def my_abs(a):
    if a>=0:
        return a
    elif a<0:
        return a*(-1)

print(my_abs(-9))
```

42. 파이썬 모듈 이해하기

: 파이썬에서는 각각의 소스 파일을 일컬어 모듈이라고 한다.

함수를 모아놓은 파이썬 스크립트

이미 만들어져 있고 안정성이 검증된 함수들을 성격에 맞게 하나의 파일로 묶어놓은것을 모듈이라고 한다.

외부에 모듈에 있는 함수를 사용하려면 이 모듈을 먼저 우리 코드로 가져와서 자유롭게 사용할 수 있도록 해야하는데 이런일을 파이썬에서 모듈을 import한다 라고 한다.

```
ex)
import time
```

```

print('5초간 프로그램을 정지합니다')
time.sleep(5)
print('5초가 지났습니다')
5초간 프로그램을 정지합니다
5초가 지났습니다

```

문제126. 아래의 4개 함수를 가지고 cal_test3.py라는 모듈을 만드시오.

주피터에서 하기

1. 주피터 실행되는 위치에 cal_test3.py로 새로 만든다.

```

def plus(a,b):
    return a+b

```

```

def minus(a,b):
    return a-b

```

```

def mul(a,b):
    return a*b

```

```

def divide(a,b):
    return a/b

```

```

import cal_test3
print(cal_test3.plus(10.5))

```

문제127. 우리가 파이썬으로 만든 오라클 함수들을 모아서 oracle_func.py라는 모듈을 생성하고 호출해서 수행하시오.

보기)

```
$ vi oracle_func.py
```

```

-----
initcap
substr
lpad
rpad
instr
months_between
add_months
next_day
last_day
nvl
decode
-----

```

```
vi test7.py
```

```
import oracle_func
print(oracle_func.next_day('2018-07-19', '금요일' ) )
```

답)

```
def decode(a,b,c,d):
    if int(a) == int(b):
        return c
    else :
        return d
```

```
def nvl(val1, val2):
    if val1 is '':
        return val2
    return val1
```

```
def months_between(v1,v2):
    import datetime
    a=datetime.datetime.strptime(v1,'%Y-%m-%d')
    b=datetime.datetime.strptime(v2,'%Y-%m-%d')
    return ((b.year-a.year)*12)+b.month-a.month
```

```
def next_day(val1,val2):
    import datetime as date
    dy={'월요일':0,'화요일':1,'수요일':2,'목요일':3,'금요일':4,'토요일':5,'일요일':6}
    n=dy[val2]
    d = date.datetime.strptime(val1,'%Y-%m-%d').weekday()
    res=(7+(n-d))%7
    return date.datetime.strptime(val1, '%Y-%m-%d').date()+date.timedelta(days=res)
```

```
def last_day(date):
    from calendar import monthrange
    from datetime import datetime
    year = datetime.strptime(date, '%Y-%m-%d').year
    month = datetime.strptime(date, '%Y-%m-%d').month
    return date[:8] + str(monthrange(year, month)[1])
```

```
def add_months(val, x):
    import datetime
    from datetime import date
    from dateutil.relativedelta import relativedelta
```

```

    change_date = datetime.datetime.strptime(val, '%Y-%m-%d')
    result = change_date + relativedelta(months =+ x)
    return result

def rpad(val, x, y):
    return val + y*(x-len(val))

def lpad(val, x, y):
    return y*(x-len(val)) + val

def instr(val, x):
    cnt=0
    for i in range(len(val)):
        val[i]
        cnt+=1
        if val[i]==x:
            break
    return cnt

def substr(val, x, y):
    return val[x-1:x-1+y]

def initcap(string):
    return(string[0].upper() + string[1:].lower())

```

43. 파이썬 패키지 이해하기

: 우리가 음악 파일을 저장할때도 장르별로 폴더를 만들어서 별도로 저장하듯이 파이썬 모듈도 음악처럼 개수가 많아지면 폴더별로 관리를 해야 관리가 편해지는데 이 폴더가 패키지이다.

ex)
C:\Users\Wstu\PycharmProjects\untitled\cal_test3.py

평범한 폴더가 패키지로 인정받으려면 반드시 갖고 있어야하는 파일
`__init__.py`

쥬피터 폴더를 잘몰라서 아톰으로 ㄱㄱ
[root@wskim ~]# mkdir my_loc
[root@wskim ~]# vi __init__.py

문제128. `__init__.py`는 대개 바꿔두는 것이 보통인데 이 파일을 손대는 경우는 언제인가?

보기)
__init__.py안에
__all__ = ['cal_test3', 'oracle_func'] 을 적어준다

답)
__all__ 변수를 조정할 때 손을 댈다.
__init__.py를 열어서 __all__ = ['cal_test3', 'oracle_func']을 설정해주면 패키지의 모듈을 불러올때 from 패키지 import * 명령어를 사용할 수 있다.

문제129. from my_loc import * 를 테스트하시오

답)
from my_loc import *
print(oracle_func.next_day('2018-07-19', '금요일'))
2018-07-20

문제130. mean이라는 함수를 생성하시오.(가변 매개변수를 이용해서 숫자를 몇 개 넣고 실행하면 상관없이 평균값이 구해지게 하시오.)

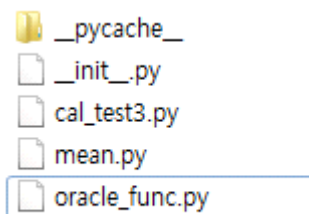
보기)
print(mean(7,6,5,3,1,6,8))

답)
def mean(*args):
 return sum(args)/len(args)

print(mean(7,6,5,3,1,6,8))

문제131. mean 함수를 가지고 mean.py 모듈을 생성하시오. (mean.py모듈을 my_loc폴더에 넣으시오.

파일 만들어주고



mean.py안에
def mean(*args):
 return sum(args)/len(args)

__init__.py안에
__all__ = ['cal_test3', 'oracle_func','mean']

```
from my_loc import *
print(mean(4,4,2,4,6))
```

44. 파이썬 모듈 임포트 이해하기1(import)

설명 : import문을 만나면 파이썬 모듈을 찾는 순서

1. 파이썬 내장 모듈
2. sys.path에 정의 되어있는 디렉토리

문제132. 파이썬에 내장 모듈이 무엇이 있는지 출력하시오.

답)

```
import sys
print(sys.builtin_module_names)
```

```
('ast', '_bisect', '_blake2', '_codecs', '_codecs_cn', '_codecs_hk', '_codecs_iso
2022', '_codecs_jp', '_codecs_kr', '_codecs_tw', '_collections', '_csv', '_datetim
e', '_functools', '_heapq', '_imp', '_io', '_json', '_locale', '_lsprof', '_md5',
'_multibytecodec', '_opcode', '_operator', '_pickle', '_random', '_sha1', '_sha25
6', '_sha3', '_sha512', '_signal', '_sre', '_stat', '_string', '_struct', '_symtab
le', '_thread', '_tracemalloc', '_warnings', '_weakref', '_winapi', 'array', 'atex
it', 'audioop', 'binascii', 'builtins', 'cmath', 'errno', 'faulthandler', 'gc', 'i
tertools', 'marshal', 'math', 'mmap', 'msvcrt', 'nt', 'parser', 'sys', 'time', 'wi
nreg', 'xxsubtype', 'zipimport', 'zlib')
```

문제133. sys.path의 내용을 프린트 하시오.

답)

```
import sys
for path in sys.path:
    print(path)
```

```
C:\Users\wit\Anaconda3\python36.zip
C:\Users\wit\Anaconda3\DLLs
C:\Users\wit\Anaconda3\lib
C:\Users\wit\Anaconda3
C:\Users\wit\Anaconda3\lib\site-packages
C:\Users\wit\Anaconda3\lib\site-packages\win32
C:\Users\wit\Anaconda3\lib\site-packages\win32\lib
C:\Users\wit\Anaconda3\lib\site-packages\Pythonwin
C:\Users\wit\Anaconda3\lib\site-packages\Python\extensions
C:\Users\wit\python
```

- **site-pacakges가 무엇인가?**

site-pacakges란 파이썬의 기본 라이브러리 패키지외에 추가적인 패키지를 설치하는 디렉토리.

site_packages 디렉토리에 여러가지 소프트웨어가 사용할 공통 모듈을 넣어주면 물리적인 장소에 구애 받지 않고 모듈에 접근하여 반입 할 수 있다.

문제134. oracle_func.py 를 site-packages 디렉토리 밑에 넣어두고 C:\Users\Wit\Anaconda3\lib\Waaa\test7.py 를 실행해보시오.

답)

```
import oracle_func
print(oracle_func.next_day('2018-07-01','금요일'))
2018-07-06
```

45. 파이썬 모듈 임포트 이해하기2(from~import)

C:\Users\Wit\Anaconda3\my_loc(패키지)

```
|
oracle_func.py(모듈)
|
add_months(함수)
next_day(함수)
last_day(함수)
months_between(함수)
```

my_loc 패키지 내에 oracle_func모듈안에 next_day함수를 사용하고 싶으면?

답)

```
import my_loc.oracle_func
print(oracle_func.next_day('2018-07-01','금요일'))
```

좀더 편하게 코딩하는 방법

```
import my_loc.oracle_func as of
import my_loc.oracle_func as of
```

```
print(of.next_day('2018-07-01','금요일'))
print(of.add_months('2018-07-01',100))
```

여기서!! 더 편하게 코딩하는 방법

```
from my_loc.oracle_func import next_day
from my_loc.oracle_func import add_months
from my_loc.oracle_func import last_day
```

```
print(next_day('2018-07-01','금요일'))
print(add_months('2018-07-01',100))
print(last_day('2018-07-01'))
```

여기서!! 좀더 편하게 코딩

```
from my_loc.oracle_func import *
```



```
print(next_day('2018-07-01','금요일'))
print(add_months('2018-07-01',100))
print(last_day('2018-07-01'))
```

이걸 사용하려면



init파일안에

```
__all__ = ['cal_test3', 'oracle_func', 'mean']
```

을 넣어줘야한다.

46. 파이썬 모듈 임포트 이해하기3(import ~ as)

문제135. 아래와 같이 코딩해서 수행되게 하시오.

보기)

```
from my_loc.oracle_func import *
```

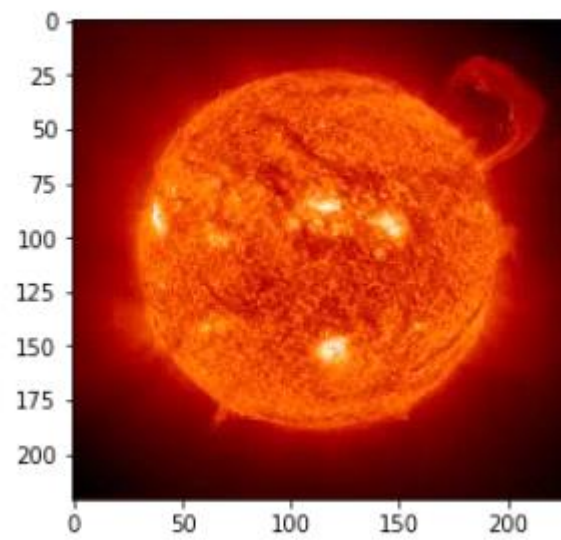
```
print(next_day('2018-07-01','금요일'))
print(add_months('2018-07-01',100))
print(last_day('2018-07-01'))
2018-07-06
2026-11-01 00:00:00
2018-07-31
```

47. 파일 열고 닫기(open, close)

ex)

```
import PIL.Image as pilimg
import numpy as np
import matplotlib.pyplot as plt
im = pilimg.open('C:\python_data\sun.jpg')
pix = np.array(im)
plt.imshow(pix)
```

Out [31]: <matplotlib.image.AxesImage at 0x7e436d8>



파이썬 149제(클래스)

2018년 7월 19일 목요일 오후 4:38

48. 클래스 이해하기(class)

클래스는 자료형이고 아직 객체가 되지 않았다.

자료형 : 붕어빵 틀 / 차 설계도

객체 : 붕어빵 / 차

ex) 총 클래스를 생성하고 객체화 시키시오.

```
class gun():
```

```
    def __init__(self): #클래스가 실제화 될때 바로 작동하는 메소드
        self.bullet=0
```

```
    def charge(self,num): #장전하는 기능
        self.bullet=num
```

```
    def shoot(self,num): #총 쏘는 기능
        for i in range(num):
            if self.bullet>0:
                print('탕!')
                self.bullet -=1
            elif self.bullet==0:
                print('총알부족')
                break
```

```
    def print(self): #출력하는 기능
        print(' {}발 남았습니다.'.format(self.bullet) )
```

#총 실제화

```
gun=gun()
```

#총 충전

```
gun.charge(10)
```

#총알 쏘기

```
gun.shoot(3)
```

#몇발 남음?

```
gun.print()
```

```
탕!  
탕!  
탕!  
7발 남았습니다.
```

```
gun.shoot(10)
```

```
gun.shoot(1)
```

```
탕!  
탕!  
탕!  
탕!  
탕!  
탕!  
탕!  
탕!  
탕!  
탕!  
탕!  
0발 남았습니다.  
총알부족
```

문제136. (마지막)고릴라 클래스를 구현해서 고릴라 객체를 만드시오.

기능)

1. 소리지르기 : shout

와~~~

소리를 지를 수 없습니다(바나나 없으면)

2. 걷는 기능 : walk

앞으로 걸어간다.

앞으로 걸어갈 수 없음(바나나 없으면)

3. 바나나 먹기 : eat

4. 바나나가 배 속에 몇 개 남았는지 프린트 하는 기능

바나나를 10개 먹으면 소리를 10번 지를수 있다.

바나나를 10개 먹었으면 앞으로 10번 갈 수 있다.

답)

```
class gorila():
```

```
    def __init__(self): #클래스가 실제화 될때 바로 작동하는 메소드  
        self.banana=0
```

```
    def eat(self,num): #바나나 먹기  
        self.banana=num
```

```
def shout(self,num): #소리지르기
    for i in range(num):
        if self.banana>0:
            print('쑤아리 질러~~~')
            self.banana-=1
        elif self.banana==0:
            print('배가고파 소리 못질러')
            break
```

```
def walk(self,num): #앞으로
    for i in range(num):
        if self.banana>0:
            print('앞으로~앞으로~')
            self.banana -=1
        elif self.banana==0:
            print('배고파 힘이없어')
            break
```

```
def print(self): #출력하는 기능
    print('{}개 남음'.format(self.banana) )
```

```
gorila=gorila()
```

```
gorila.eat(10)
```

```
gorila.shout(3)
```

```
gorila.walk(7)
```

```
gorila.print()
```

```
gorila.walk(1)
```

```
쑤아리 질러~~~
```

```
쑤아리 질러~~~
```

```
쑤아리 질러~~~
```

```
앞으로~앞으로~
```

```
앞으로~앞으로~
```

```
앞으로~앞으로~
```

```
앞으로~앞으로~
```

```
앞으로~앞으로~
```

```
앞으로~앞으로~
```

```
앞으로~앞으로~
```

```
0개 남음
```

```
배고파 힘이없어
```

문제137. 위의 고릴라 클래스를 gorila.py 모듈로 생성하고 test20.py에 아래의 스크립트를 작성해서 잘 수행되는지 확인하시오.

```
보기)
import gorila
gorila=gorila()
```

```
gorila.eat(10)
gorila.shout(3)
gorila.walk(7)
```

```
gorila.print()
gorila.walk(1)
```

답)
경로를 anaconda3에 gorila.py를 만들어준다.

49. 클래스 멤버와 인스턴스 멤버 이해하기

- 클래스 용어 설명
- 1. 인스턴스와 클래스의 뜻

인스턴스 변수?

: 사람의 이름과 같이 각각의 인스턴스 마다 가지고 있는 고유한 데이터

클래스 변수?

: 단체의 단체명과 같이 클래스로 만들어진 모든 인스턴스가 공유하는 데이터

```
인스턴스 = 클래스()
고릴라_1 = 고릴라설계도()
고릴라_2 = 고릴라설계도()
```

2. self가 무엇?

: self는 클래스에서 사용하는 최초의 매개변수인데 자기 자신을 가리킨다.

```
def __init__(self):
    self.banana=num
```

ex)

클래스명을 코딩할때는 앞의 철자를 대문자로 작성(약속)

```
class Employees(object):
    raise_amount=1.1 #클래스 변수
```

#클래스 변수를 선언하는 이유는? 다른 함수에서도 사용하려고

```
def __init__(self, first, last, pay): #클래스가 객체화되면 바로실행되는 함수
    self.first=first
    self.last=last
    self.pay=pay
    self.email=first.lower() + ',' + last.lower() + '@gmail.com'

    def full_name(self):    #전체 이름 출력하는 함수
        return '{} {}'.format(self.first, self.last)

    def apply_raise(self):    #월급을 인상하는 함수
        self.pay = int(self.pay*self.raise_amount)
```

#사원 2명을 객체화(인스턴스화) 시킨다

```
emp_1 = Employees('ingo','kim', 5000)
emp_2 = Employees('sungsu', 'lee', 5000)
```

```
print(emp_1.pay)    #기존 연봉
emp_1.apply_raise() #월급 인상
print(emp_1.pay)
5000
5500
```

클래스 변수로 **raise_amount**를 사용하는 이유?

여러 함수에서 사용되고있는데 나중에 인상률이 변경이된다면 클래스 변수만 바꿔주면

다른 함수에서도 변경된게 적용이 된다. (하드코딩 할 필요 없음)

- 인스턴스 변수의 사용 예

```
emp_1 = Employees('heachan','eun', 5000)
emp_2 = Employees('hyunsoo', 'shin', 5000)
```

현수형이랑 틀리게 해찬이형에게 더 높은 연봉을 주고 싶다
10%가 아니라 20%를 인상해주고 싶다면?

"인스턴스 변수를 사용해서 특별 인상 적용"

```
emp_1.raise_amount = 1.2 #인스턴스 변수 ( 클래스 밖에서 선언해줘야한다)
```

```
print(emp_1.pay) #기존연봉
emp_1.apply_raise() # 인상을 적용
```

```
print(emp_1.pay)
```

```
print(emp_2.pay) #기존연봉
emp_2.apply_raise() # 인상률 적용
print(emp_2.pay)
```

```
해찬쓰
5000
6000
현수형
5000
5500
```

문제139. 어제 만들었던 고릴라 클래스를 가지고 두명의 고릴라를 만드시오.

(배속에 바나나가 몇 개 남았는지 프린트 하시오.)

보기)

```
heachan_gorilla=gorila()
hyunsoo_gorilla=gorila()
```

```
heachan_gorilla.print()
hyunsoo_gorilla.print()
```

```
0개 남음
0개 남음
```

문제140. 해찬 고릴라와 현수 고릴라가 태어날때 바나나를 배 속에 넣고 태어날 수 있게 하시오.

답)

```
def __init__(self,num):
    self.banana=num
```

```
..
..
```

```
heachan_gorilla=gorila(10)
hyunsoo_gorilla=gorila(10)
```

```
heachan_gorilla.print()
hyunsoo_gorilla.print()
```

```
10개 남음
10개 남음
```

만약 바나나를 더먹이고 싶을 경우

```
def __init__(self,num):
    self.banana=num
```



```

def eat(self,num): #바나나 먹기
    self.banana=num    #이분이 =이라 더먹으려면 +=로 바꿔준다

def eat(self,num): #바나나 먹기
    self.banana+=num

heachan_gorilla=gorilla(5)
hyunsoo_gorilla=gorilla(10)

heachan_gorilla.eat(10)

heachan_gorilla.print()
hyunsoo_gorilla.print()
15개 남음
10개 남음

```

문제141. 인스턴스 변수를 설정해서 현수 고릴라와는 다르게 해찬 고릴라는 바나나를 몇 개를

먹든 먹은것의 두배로 배 속에 채워지게 하시오.

보기)

```

heachan_gorilla=gorilla(10) #바나나 10개
heachan_gorilla.eat(10)
heachan_gorilla.print() # 바나나 30개

```

```

hyunsoo_gorilla=gorilla(10)
hyunsoo_gorilla.eat(10)
hyunsoo_gorilla.print() #바나나 20개

```

답)

```

class gorilla():
    eat_multi =1
    ..
    ..
    def eat(self,num): #바나나 먹기
        self.banana+=(num*self.eat_multi)

```

```

heachan_gorilla=gorilla(10)

```

```

heachan_gorilla.eat_multi=2

```

```

heachan_gorilla.eat(20)

```

```
heachan_gorilla.print()
```

```
hyunsoo_gorilla=gorilla(10)
```

```
hyunsoo_gorilla.eat(10)
```

```
hyunsoo_gorilla.print()
```

```
50개 남음
```

```
20개 남음
```

설명)

: 클래스 내의 코드에는 해찬 고릴라와 현수 고릴라를 다룰 수 있는 코드가 없다.

그래서 해찬 고릴라와 현수 고릴라를 다룰 수 있도록 self를 사용해서 해찬과 현수 객체를 담는다.

```
heachan_gorilla.eat(10)
    ↓           ↓
def eat(self, num):
    self.banana=num
```

문제142. (점심시간) 고릴라클래스에 기능을 하나더 추가하시오. 가슴을 쿵광쿵광 두드리기.

라는 기능을 추가하는데 두드리면 바나나 1개씩 소진되게 하시오.

답)

```
def hit(self,num):
    for i in range(num):
        if self.banana>0:
            print('쿵광!쿵광!')
            self.banana -=1
        elif self.banana ==0:
            print('배고파 힘이없어')
            break
```

```
gorilla=gorilla(10)
```

```
gorilla.hit(3)
```

```
gorilla.print()
```

```
쿵광!쿵광!
```

```
쿵광!쿵광!
```

```
쿵광!쿵광!
```

```
7개 남음
```

static method

: self를 매개변수로 받지않은 메소드

여러 인스턴스가 공유해서 사용하는 메소드

ex)

총 설계도를 가져와서 gun_1 총과 gun_2총을 생성하시오.

```
class gun():
    def __init__(self): #클래스가 실제화 될때 바로 작동하는 메소드
        self.bullet=0

    def charge(self,num): #장전하는 기능
        self.bullet=num

    def shoot(self,num): #총 쏘는 기능
        for i in range(num):
            if self.bullet>0:
                print('탕!')
                self.bullet -=1
            elif self.bullet==0:
                print('총알부족')
                break

    def print(self): #출력하는 기능
        print(' {}발 남았습니다.'.format(self.bullet) )

gun_1=gun()
gun_2=gun()

gun_1.charge(10)
gun_1.shoot(3)
gun_1.print()

gun_2.charge(10)
gun_2.shoot(6)
gun_2.print()
```

```

탕!
탕!
탕!
7발 남았습니다.
탕!
탕!
탕!
탕!
탕!
탕!
4발 남았습니다.

```

설명 : gun_1총과 gun_2총은 서로 별개의 총이다.

문제143. gun_1과 gun_2가 서로 별개의 총이 맞는지 확인하시오.

```

답)
print(gun_1)
print(gun_2)
<__main__.gun object at 0x0000000005150AC8>
<__main__.gun object at 0x0000000005169E80>

```

문제144. 총은 별개의 총이지만 설계도는 같은 설계도를 사용했는지 확인하시오.

```

답)
print(gun_1.__class__)
print(gun_2.__class__)
<class '__main__.gun'>
<class '__main__.gun'>

```

문제145. 총 클래스의 충전 메소드, 쏘는 메소드, 프린트하는 메소드를 클래스 메소드가 아니라

static 메소드로 생성하고 다시 총2개를 만들어서 아까처럼 쏘보시오.

```

답)
class Gun():
    bullet=0

    @staticmethod #데코레이터
    def charge(num): #장전하는 기능
        Gun.bullet=num

    @staticmethod
    def shoot(num): #총 쏘는 기능
        for i in range(num):
            if Gun.bullet>0:
                print('탕!')
                Gun.bullet -=1

```

```

elif Gun.bullet==0:
    print('총알부족')
    break

```

`@staticmethod`

```

def print(): #출력하는 기능
    print(' {}발 남았습니다.'.format(Gun.bullet) )

```

```

gun_1=Gun()
gun_2=Gun()

```

```

gun_1.charge(10)
gun_1.shoot(3)
gun_1.print()

```

```

gun_2.shoot(6)
gun_2.print()
탄!
탄!
탄!
7발 남았습니다.
탄!
탄!
탄!
탄!
탄!
탄!
1발 남았습니다.

```

설명)

static함수는 **self**를 못써줘서 **bullet**을 선언해주고 **self**.을 **Gun**.으로 바꿔준다.

50. 클래스 메소드 이해하기

```

class MyClass: #낙타 등 표기법(CamelCase)
    def sayHello(self):
        print('안녕하세요')

    def sayBye(self, name):
        print('%s! 다음에 보자!' %name)

obj = MyClass()
obj.sayHello() # '안녕하세요'가 출력됨
obj.sayBye('철수') # '철수! 다음에 보자!'가 출력됨

```

- 함수를 모아놓은 **모듈과 클래스의 차이**는 무엇일까?

1. `__init__`
2. 인스턴스 변수
3. 모듈의 함수들은 독립적인데, 클래스의 함수(메소드)들은 서로 공유하여 사용할 수 있다.

ex)

```
gun = Gun()
객체=클래스()
```

```
gun.shoot(3)
객체.메소드
```

[1,2,3,4,5] -> 리스트 객체

```
[1,2,3,4,5].count(2)
리스트객체.메소드
```

- **리스트 변수(객체)의 유용한 메소드**

1. `count()`
2. `append()`
3. `insert()`
4. `remove()`
5. `pop()`
6. `index()`
7. `sort()`
8. `reverse()`

```
a=[1,2,3,4,5]
변수=리스트 객체
```

변수 a는 리스트 객체의 메모리 주소를 가지고 있어서 `a.count(2)`라고 하면 a를 통해서 [1,2,3,4,5] 객체의 `count` 메소드를 실행한 것이다.

51. 클래스 생성자와 소멸자 이해하기

: 객체가 만들어 질때 호출되는 함수를 생성자(`__init__`)라고 한다.

객체가 사라질때 호출되는 함수를 소멸자(`__del__`)라고 한다.

생성자는 객체를 초기화 할때 자주 사용한다.

생성자 ex)

```
class MyClass:
    def __init__(self):
```

```

self.var = '안녕하세요!'
print('MyClass 인스턴스 객체가 생성되었습니다')

obj = MyClass()  # 'MyClass 인스턴스 객체가 생성되었습니다'가 출력됨
print(obj.var)   # '안녕하세요'가 출력됨

```

문제146. 고릴라 객체를 아래와 같이 생성하면 바로 고릴라가 생성되었습니다. 라는 메시지가

출력되게 하시오.

```

보기)
hyunsu_gorilla=gorila(10)

```

바나나 10개를 배 속에 둔 hyunsu_gorilla가 생성되었습니다.

답)

```

def __init__(self,num): #클래스가 실제화 될때 바로 작동하는 메소드
    self.banana=num
    print('바나나 %d 개를 배 속에 둔 고릴라가 생성되었습니다.' %num)

```

바나나 10 개를 배 속에 둔 고릴라가 생성되었습니다.

소멸자 ex)

```

class MyClass:
    def __init__(self):
        self.var = '안녕하세요!'
        print('MyClass 인스턴스 객체가 생성되었습니다')

    def __del__(self):
        print('MyClass 인스턴스 객체가 메모리에서 제거됩니다')

```

```

obj = MyClass()  # 'MyClass 인스턴스 객체가 생성되었습니다'가 출력됨
del obj

```

```

MyClass 인스턴스 객체가 생성되었습니다
MyClass 인스턴스 객체가 메모리에서 제거됩니다
MyClass 인스턴스 객체가 메모리에서 제거됩니다

```

문제147. Card 클래스를 생성해서 Card 발급을 하시오.

보기)

```

class Card:
    def __init__(self,num):

```

```

self.money=num
print("%d원 카드가 생성되었습니다" %num)

def charge(self,num):
    self.money+=num
    print('%d 원이 충전되었습니다' %num)
    print('현재 잔액 : %d' %self.money)

def consume(self,num):
    if self.money>0:
        self.money-=num
        print('%d 원을 사용했습니다' %num)
        print('%d 원이 남았습니다' %self.money)
    elif self.consume==0:
        print('잔액이 없습니다')

card1=Card(10000)
card1.charge(10000)
card1.consume(10000)
card1.consume(10000)
10000원 카드가 생성되었습니다
10000 원이 충전되었습니다
현재 잔액 : 20000
10000 원을 사용했습니다
10000 원이 남았습니다
10000 원을 사용했습니다
0 원이 남았습니다

```

52. 클래스 상속 이해하기

상속이란? 클래스끼리 기능을 물려주는 것

부모 클래스 -----> 자식 클래스

기능

상속을 받게 되면 부모의 기능을 굳이 자식 클래스에 코딩을 하지 않아도 된다.

ex1)

```

class Father:
    def __init__(self):
        print("hello ~~~~")
class Child(father):
    pass

```

father=Father()

father.base_method()


```
child=Child()
child.base_method()
hello ~~~~
hello ~~~~
```

ex2)

그러던 어느날 father에게 속성(변수)가 하나 생겼다.

```
class Father:
    def __init__(self):
        print("hello ~~~~")
        self.message="Good Morning"
```

```
class Child(Father):
    def __init__(self):
        print ("hello ~~~ I am a child")
```

```
father=Father()
print(father.message)
hello ~~~~
Good Morning
```

아래를 쓰면 에러가난다.

```
child=Child()
print(child.message)
```

설명: 파이썬은 암묵적인 것을 싫어하고 명시적인 것을 좋아한다.

이를 해결 하는 방법은

```
class Father:
    def __init__(self):
        print("hello ~~~~")
        self.message="Good Morning"

class Child(Father):
    def __init__(self):
        Father.__init__(self) #이부분 추가
        print ("hello ~~~ I am a child")
```

```
father=Father()
print(father.message)
```

```
child=Child()
print(child.message)
```

설명: 위와 같이 `Father.__init__(self)` 라고 해도 되고 또는 파이썬 내장 함수중에 `super()` 를 사용해서 아래와 같이 작성해도 된다.

```
class Father:
    def __init__(self):
        print("hello ~~~~")
        self.message="Good Morning"

class Child(Father):
    def __init__(self):
        super().__init__()
        print ("hello ~~~ I am a child")
```

```
father=Father()
print(father.message)
```

```
child=Child()
print(child.message)
```

가장 큰 문제점을 위해 위의 두가지 방법중에 `super()`를 쓰는게 더 좋다
가장 큰 문제점 이란?
"다중 상속시 나타나는 죽음의 다이아몬드 "

다중 상속

다중 상속이란 두개 이상의 클래스를 상속 받는 것을 말한다.
이 경우에는 두 클래스의 모든 속성을 물려받게 된다.
이는 하나의 자식 클래스가 두개 이상의 부모 클래스를 가지는 것이라 할 수 있다.



```
ex1)
class Father1:
    def func(self):
        print("지식")
```

```

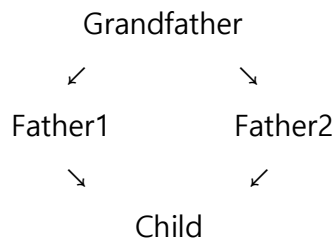
class Father2:
    def funct(self):
        print("지혜")

class Child(Father1, Father2):
    def child_func(self):
        Father1.func(self)
        Father2.funct(self)

childobject = Child()
childobject.child_func()
지식
지혜

```

다중 상속시 주의 할 점!!!!!! "죽음의 다이아몬드 상속"



```

class Grandfather:
    def __init__(self):
        print('튼튼한 두팔')
class Father1(Grandfather):
    def __init__(self):
        Grandfather.__init__(self)
        print("지식")
class Father2(Grandfather):
    def __init__(self):
        Grandfather.__init__(self)
        print("지혜")
class Child(Father1, Father2):
    def __init__(self):
        Father1.__init__(self)
        Father2.__init__(self)
        print('자기 만족도가 높은 삶')

```

```
child = Child()
튼튼한 두 팔
지식
튼튼한 두 팔
지혜
자기 만족도가 높은 삶
```

팔이 4개가 되버린다...

문제148. 다이아몬드 상속을 받았더니 그만 팔이 4개가 되버렸다. 다시 팔이 2개가 되게 하시오.

(앞에서 배웠던 내장 함수)

답)

```
class Grandfather:
```

```
    def __init__(self):
        print('튼튼한 두 팔')
```

```
class Father1(Grandfather):
```

```
    def __init__(self):
        super().__init__()
        print("지식")
```

```
class Father2(Grandfather):
```

```
    def __init__(self):
        super().__init__()
        print("지혜")
```

```
class Child(Father1, Father2):
```

```
    def __init__(self):
        super().__init__()
#        Father1.__init__(self)
#        Father2.__init__(self)
        print('자기 만족도가 높은 삶')
```

```
child = Child()
```

```
튼튼한 두 팔
지혜
지식
자기 만족도가 높은 삶
```

문제149. 위에서 만든 card 클래스를 상속받는 클래스를 생성하는데 영화할인 카드 클래스를 생

성하시오.

card

↓

영화할인 카드 주유할인 카드

1. movie_card 만들때 charge-->x, consume -->x

영화를 보면 영화 관람료의 10% 할인 해주는 메소드를 추가해서 만드시오.
(point_charge메소드)

```
movie_card=Movie_card()
영화카드가 발급 되었습니다.
```

```
movie_card.charge(50000)
50000원이 충전되었습니다.
```

```
movie_card.consume(20000)
20000원 사용했습니다. 30000원 남았습니다.
```

```
movie_card.consume(12000)
10800원 사용했습니다. 19200원이 남았습니다.
```

답)

```
class Card:
```

```
    def __init__(self,num):
        self.money=num
        print("%d원 카드가 생성되었습니다" %num)
```

```
    def charge(self,num):
        self.money+=num
        print('%d 원이 충전되었습니다' %num)
        print('현재 잔액 : %d' %self.money)
```

```
    def consume(self,num):
        if self.money>0:
            self.money-=num
            print('%d 원을 사용했습니다' %num)
            print('%d 원이 남았습니다' %self.money)
        elif self.consume==0:
            print('잔액이 없습니다')
```

```
class Movie_card(Card):
```

```

def movie_consume(self,num):
    discount=0.9
    if self.money>0:
        self.money-=int(num*discount)
        print('%d 원을 사용했습니다' %self.money)
        print('%d 원이 남았습니다' %self.money)
    elif self.consume==0:
        print('잔액이 없습니다')

card1=Card(10000)
movie_card1=Movie_card(50000)
movie_card1.consume(20000)
movie_card1.movie_consume(12000)

10000원 카드가 생성되었습니다
50000원 카드가 생성되었습니다
20000 원을 사용했습니다
30000 원이 남았습니다
19200 원을 사용했습니다
19200 원이 남았습니다

```

문제150. 평균, 분산, 표준편차 함수를 가지고 stats라는 클래스를 생성하시오.

참고)

```

x = [0.062 , 0.071, 0.067, 0.068 , 0.066,
     0.062 , 0.068, 0.067, 0.060, 0.065 ]

```

```

def mean(args):
    return sum(args)/ len(args)

```

```

def variance(args):
    total = 0
    m = mean(args)

    for i in args:
        total += (i-m)**2

    return total/len(args)

```

```

def stddev(args):
    v = variance(args)
    return math.sqrt(v)

```

답)

```
class My_stats():
    import math
    def __init__(self):
        self.total=0
        print('통계 객체가 생성')

    def mean(self,args):
        return sum(args)/ len(args)

    def variance(self,args):
        total = 0
        m = self.mean(args)

        for i in args:
            total += (i-m)**2

        return total/len(args)

    def stddev(self,args):
        v = self.variance(args)
        return math.sqrt(v)

stats=My_stats()
x = [0.062 , 0.071, 0.067, 0.068 , 0.066, 0.062 , 0.068, 0.067, 0.060, 0.065 ]

print(stats.mean(x))
print(stats.variance(x))
print(stats.stddev(x))
통계 객체가 생성
0.06559999999999999
1.0240000000000004e-05
0.0032000000000000006
```

문제151. 중앙값을 구하는 함수를 생성하고 그 함수를 My_stats클래스에 추가 하시오.

답)

```
class My_stats():
    import math
    import numpy as np
```

```

def __init__(self):
    self.total=0
    print('통계 객체가 생성')

def mean(self,args):
    return sum(args)/ len(args)

def variance(self,args):
    total = 0
    m = self.mean(args)

    for i in args:
        total += (i-m)**2

    return total/len(args)

def stddev(self,args):
    v = self.variance(args)
    return math.sqrt(v)

def median(self, args):
    a=np.array(args)
    b=a.median
    return b

stats=My_stats()
x = [0.062 , 0.071, 0.067, 0.068 , 0.066, 0.062 , 0.068, 0.067, 0.060, 0.065 ]

print(stats.mean(x))
print(stats.variance(x))
print(stats.stddev(x))
print(stats.np.median(x))

```

문제152. 최빈값 구하는 함수를 만들고 My_stats 클래스에 추가하시오.

답)

```

class My_stats():
    import math
    import numpy as np
    from collections import Counter

```



```

def __init__(self):
    self.total=0
    print('통계 객체가 생성')

def mean(self,args):    #평균
    return sum(args)/ len(args)

def variance(self,args):    #분산
    total = 0
    m = self.mean(args)

    for i in args:
        total += (i-m)**2

    return total/len(args)

def stddev(self,args):    #표준편차
    v = self.variance(args)
    return math.sqrt(v)

def median(self, args): #중앙값
    a=np.array(args)
    b=a.median
    return b

def mode(self,args): #최빈
    c = Counter(args)
    order = c.most_common()
    maximum = order[0][1]
    modes = []
    for num in order:
        if num[1] == maximum:
            modes.append(num[0])
    return modes

stats2=My_stats()
x = [0.062 , 0.071, 0.067, 0.068 , 0.066, 0.062 , 0.068, 0.067, 0.060, 0.065 ]

print(stats2.mean(x))
print(stats2.variance(x))

```

```

print(stats2.stddev(x))
print(stats2.np.median(x))
print(stats2.mode(x))

```

문제153. 범위를 구하는 함수를 생성하고 My_stats클래스에 추가하시오.

```

답)
..
..
def my_range(self, args):
    return max(args)-min(args)

print(stats2.my_range(x))

```

문제154. 사분위수 범위를 구하는 함수를 생성하고 My_stats클래스에 추가하시오.

```

답)
def data_range(self,args): #사분위수
    import numpy as np
    a=np.percentile(args,75) - np.percentile(args,25)
    return a

```

가끔 함수밖 클래스 안에 임포트해주면 못볼러올 때가 있다!!!

문제155. 통계 클래스를 모듈로 만들어서 아래와 같이 실행되게 하시오.

```

보기)
import my_stats
stats2=my_stats.My_stats()

x = [0.062 , 0.071, 0.067, 0.068 , 0.066, 0.062 , 0.068, 0.067, 0.060, 0.065 ]
print(stats2.median(x))

```

답)

파이썬 149제(예외 처리)

2018년 7월 24일 화요일 오전 11:07

53. 예외처리 이해하기(try~except~)

예외

:프로그램에서 에러가 발생했을때, 에러를 핸들링하는 기능

ex)

```
def my_divide():
```

```
    x = input('분자의 숫자 입력: ')
```

```
    y = input('분모의 숫자 입력: ')
```

```
    return int(x) / int(y)
```

```
print(my_divide())
```

```
분자의 숫자 입력: 10
```

```
분모의 숫자 입력: 0
```

```
-----  
ZeroDivisionError                                T  
<ipython-input-97-3a93a772b37d> in <module>  
      4     return int(x) / int(y)  
      5  
----> 6 print(my_divide())  
  
<ipython-input-97-3a93a772b37d> in my_divic  
      2     x = input('분자의 숫자 입력: ')  
      3     y = input('분모의 숫자 입력: ')  
----> 4     return int(x) / int(y)  
      5  
      6 print(my_divide())
```

```
ZeroDivisionError: division by zero
```

문제156. 위의 코드에 예외처리 코드를 입혀서 분모를 0으로 입력해서 위와 같은 에러가 나오면

나누기를 할 수 없습니다. 라는 메시지가 출력되게 하시오.

답)

```
def my_divide():
```

```
    try:
```

```
        x = input('분자의 숫자 입력: ')
```

```
        y = input('분모의 숫자 입력: ')
```

```
        return int(x) / int(y)
```

```
    except:
```

```
print('나누기를 할 수 없습니다')
```

```
print(my_divide())
```

```
문자의 숫자 입력: 10
```

```
분모의 숫자 입력: 0
```

```
나누기를 할 수 없습니다
```

```
None
```

문제157. 이름을 물어보게하고 이름을 입력하면 해당 사원의 월급이 출력되는 함수를 생성 하시오.

보기)

```
print(find_sal())
```

월급을 알고 싶은 사원명을 입력하세요~ SCOTT

답)

```
def find_sal():
```

```
    import pandas as pd
```

```
    emp=pd.DataFrame.from_csv("C:\python_data\emp.csv")
```

```
    name=input('월급을 알고 싶은 사원명을 입력하세요~ ')
```

```
    sal=emp[['sal']][emp['ename']==name.upper()].values[0]
```

```
    return sal
```

```
print(find_sal())
```

문제158. 위의 코드를 수정해서 없는 사원명을 입력하면 해당 사원은 없습니다. 라는 메시지가

출력되게 하시오.

답)

```
def find_sal():
```

```
    import pandas as pd
```

```
    try:
```

```
        emp=pd.DataFrame.from_csv("C:\python_data\emp.csv")
```

```
        name=input('월급을 알고 싶은 사원명을 입력하세요~ ')
```

```
        sal=emp[['sal']][emp['ename']==name.upper()].values[0]
```

```
        return sal
```

```
    except:
```

```
        return '그런 사원 없습니다.'
```

```
print(find_sal())
```

월급을 알고 싶은 사원명을 입력하세요~ ㅁㄴㅇ

그런 사원 없습니다.

문제159. while loop문을 이용해서 이름을 물어봤을때 아무것도 안넣고 그냥 엔터치면 계

속

물어보게 하시오.

답)

```
def find_sal():
    import pandas as pd
    try:
        emp=pd.DataFrame.from_csv("C:\python_data\emp.csv")
        name=""
        while name!="":
            name=input('월급을 알고 싶은 사원명을 입력하세요~ ')
            sal=emp[['sal']][emp['ename']==name.upper()].values[0]
            return sal
    except:
        return '그런 사원 없습니다.'
print(find_sal())
```

```
월급을 알고 싶은 사원명을 입력하세요~
월급을 알고 싶은 사원명을 입력하세요~
월급을 알고 싶은 사원명을 입력하세요~
월급을 알고 싶은 사원명을 입력하세요~
월급을 알고 싶은 사원명을 입력하세요~
월급을 알고 싶은 사원명을 입력하세요~
월급을 알고 싶은 사원명을 입력하세요~
월급을 알고 싶은 사원명을 입력하세요~
월급을 알고 싶은 사원명을 입력하세요~
월급을 알고 싶은 사원명을 입력하세요~ ee
그런 사원 없습니다.
```

54. 예외처리 이해하기2(try~except~else)

try:

실행할 코드 블록

except:

예외 처리 코드 블록

else:

except 절을 만나지 않았을경우 실행하는 코드 블록

ex)

이름을 물어보게하고 해당 사원의 직업이 출력되게 하는데 이름이 없으면 해당 사원은 없습니다. 라는 메시지가 출력되게하고 만약 이름이 있어서 데이터 추출에 성공했다면 직업 추출에 성공했습니다. 라는 메시지가 같이 출력되게 하시오.

```
import pandas as pd
def find_job():
    try:
```

```

emp=pd.DataFrame.from_csv("C:\wwpython_data\wwemp.csv")
name=""
while name!="":
    name=input('직업을 알고 싶은 직원명을 입력하세요~ ')

    job=emp[['job']][emp['ename']==name.upper()].values[0]
except:
    print('해당 사원은 없습니다')
else:
    print('직업 추출에 성공했습니다')
    return job
find_job()

```

문제160. 두수를 물어봐서 나눈 값을 출력하는 156번 코드를 가져와서 정상적으로 잘 나뉘지 않으면

나눈값을 잘 추출했습니다. 라는 메시지가 출력되면서 나눈 값이 출력되게 하시오.

```

답)
def my_divide():
    try:
        x = input('분자의 숫자 입력: ')
        y = input('분모의 숫자 입력: ')
        a=int(x)/int(y)
    except:
        return '나누기를 할 수 없습니다'
    else:
        print('나눈값을 잘 추출했습니다')
        return a

print(my_divide())

```

문제161. (점심시간) 숫자를 여러 개 물어보게하고 여러 개 입력하면 해당 숫자의 표준편차가

출력되는 함수를 생성하시오. 숫자대신 문자를 입력하면 출력할 수 없습니다. 라는 메시지가 출력되게 하시오.

```

보기)
print(my_sttdev())
숫자를 입력하세요 ~ 24,21,22,32,21,43,21
정상적으로 출력 되었습니다
312

```

```
print(my_stddev())
```

숫자를 입력하세요 ~ 24,21,22,32,21,43,21,a
출력할 수 없습니다

답)

55. 예외처리하기3 (복수개의 except절 사용하기)

"예외처리를 여러 개를 나열할 수 있다"

ex) 아래의 두 수를 나누는 코드를 작성하는데 아래의 두 가지가 다 실행되게 하시오.

분자 숫자를 입력하세요~ 10

분모 숫자를 입력하세요~ 0

0으로 나눌 수 없습니다.

분자 숫자를 입력하세요~ 10

분모 숫자를 입력하세요~ aaa

나눌 수 없습니다.

답)

```
def my_divide():
    try:
        x = input('분자의 숫자 입력: ')
        y = input('분모의 숫자 입력: ')
        z=int(x)/int(y)
        return z
    except ZeroDivisionError:
        return '0으로 나눌 수 없습니다.'
    except:
        return '나누기를 할 수 없습니다'
    else:
        print('나눈값을 잘 추출했습니다')
        return a

print(my_divide())
```

문제162. 위의 코드에 KeyboardInterrupt 예외를 추가해서 ctrl+c 를 누르면 사용자에게 의해서

프로그램이 중단됩니다. 라는 메시지가 출력되게하시오.

답)

```
def my_divide():
    try:
```

```

x = input('분자의 숫자 입력: ')
y = input('분모의 숫자 입력: ')
z=int(x)/int(y)
return z
except KeyboardInterrupt:
    return '사용자에 의해서 프로그램이 중단 됩니다.'
except ZeroDivisionError:
    return '0으로 나눌 수 없습니다.'
except:
    return '나누기를 할 수 없습니다'
else:
    print('나눈값을 잘 추출했습니다')
    return a

print(my_divide())

분자의 숫자 입력: 10
분모의 숫자 입력:
사용자에 의해서 프로그램이 중단 됩니다.

```

설명: 주피터는 위에 중단 버튼을 누르고 input 창에서 엔터를 친다

56. 예외처리하기4 (try~except~finally)

```

try:
    실행할 코드 블록
except:
    예외처리 코드 블록
finally:
    실행할 코드가 성공했든실패했든 무조건 실행되는 블록

```

ex) 이름을 물어보게하고 이름을 입력하면 월급이 출력되는 코드

```

import pandas as pd
def find_job():
    try:
        emp=pd.DataFrame.from_csv("C:\wwpython_data\wwemp.csv")
        name=""
        while name=="":
            name=input('직업을 알고 싶은 직원명을 입력하세요~ ')
        return job
        job=emp[['job']][emp['ename']==name.upper()].values[0]
    except:

```



```

        return '해당 사원은 없습니다'
    finally:
        print('저는 무조건 실행됩니다')

print(find_job())
직업을 알고 싶은 사원명을 입력하세요 ~ scott
저는 무조건 실행됩니다
['ANALYST']

직업을 알고 싶은 사원명을 입력하세요 ~ aa
저는 무조건 실행됩니다
해당 사원은 없습니다

```

문제163. 두 숫자를 물어보게해서 두수를 나누는 함수를 실행할때 성공하던 실패하던 아래의

메시지가 무조건 실행되게 하시오.

보기)

```

분자의 숫자 입력: 10
분모의 숫자 입력: 2
신현수가 만든 프로그램 입니다.
5.0

```

답)

```

def my_divide():
    try:
        x = input('분자의 숫자 입력: ')
        y = input('분모의 숫자 입력: ')
        z=int(x)/int(y)
        return z
    except KeyboardInterrupt:
        return '사용자에 의해서 프로그램이 중단 됩니다.'
    except ZeroDivisionError:
        return '0으로 나눌 수 없습니다.'
    except:
        return '나누기를 할 수 없습니다'
    finally:
        print('신현수가 만든 프로그램 입니다.')

print(my_divide())

```

57. 예외처리 이해하기5(try~except Exception as e)

```
try:
    print(param)
except Exception as e:
    print(e)    #name 'param' is not defined가 출력됨
```

설명: 실패한 이유를 보여준다.

58. 사용자 정의 예외 처리

"파이썬 입장에서 봤을때는 오류가 아닌데 프로그래머가 이건 오류이다. 라고 raise문을 써서 예외처리 하는 경우"

```
def 함수명:
    실행코드
    if 조건:
        raise Exception("예외가 발생")
    else:
        return 변수명
```

예외 클래스	의미 또는 예외 발생 원인
BaseException	모든 예외의 최상위 예외
SystemExit	프로그램을 종료하는 명령이 실행되었을 때
KeyboardInterrupt	Control-C 키가 입력되었을 때
Exception	(시스템 종료를 제외한) 대부분의 예외의 상위 예외
ArithmeticError	수의 연산과 관련된 문제
ZeroDivisionError	수를 0으로 나누려 할 때
AssertionError	assert 문
AttributeError	(모듈, 클래스, 인스턴스에서) 잘못된 속성을 가리킬 때
EOFError	(파일, 스트림 등에서) 읽어들이 데이터가 더이상 없을 때
ImportError	모듈을 임포트할 수 없을 때
ModuleNotFoundError	임포트할 모듈을 찾을 수 없을 때
LookupError	(시퀀스, 매핑에서) 잘못된 인덱스, 키를 인덱싱할 때
IndexError	(시퀀스에서) 잘못된 인덱스를 인덱싱할 때
KeyError	(매핑에서) 잘못된 키를 인덱싱할 때
NameError	잘못된 변수(이름)를 가리킬 때
OSError	컴퓨터 시스템(운영 체제)의 동작과 관련된 다양한 문제
ChildProcessError	하위 프로세스(프로그램이 실행한 별도의 프로그램)에서 오류

OSError	컴퓨터 시스템(운영 체제)의 동작과 관련된 다양한 문제
ChildProcessError	하위 프로세스(프로그램이 실행한 별도의 프로그램)에서 오류 발생
FileExistsError	이미 존재하는 파일/디렉토리를 새로 생성하려 할 때
FileNotFoundError	존재하지 않는 파일/디렉토리에 접근하려 할 때
IsADirectoryError	파일을 위한 명령을 디렉토리에 실행할 때
NotADirectoryError	디렉토리를 위한 명령을 파일에 실행할 때
PermissionError	명령을 실행할 권한이 없을 때
TimeoutError	명령이 수행되는 시간이 시스템이 허용한 기준을 초과했을 때
RuntimeError	다른 분류에 속하지 않는 실행시간 오류
NotImplementedError	내용 없는 메서드가 호출되었을 때
RecursionError	함수의 재귀 호출 단계가 허용한 깊이를 초과했을 때
SyntaxError	구문 오류
IndentationError	들여쓰기가 잘못되었을 때
TabError	들여쓰기에 탭과 스페이스를 번갈아가며 사용했을 때
TypeError	연산/함수가 계산해야 할 데이터가 잘못된 유형일 때
ValueError	연산/함수가 계산해야 할 데이터가 유형은 올바르나 값이 부적절할 때
UnicodeError	유니코드와 관련된 오류
Warning	심각한 오류는 아니나 주의가 필요한 사항에 관한 경고

문제164. 이름을 물어보게하고 이름을 입력하면 해당 사원의 월급이 출력되게 하는 파이썬 코드

를 작성하는데 월급이 3000이상인 직원들은 아래의 메시지가 출력되게 하시오.

보기)

"해당 사원의 월급은 볼 수 없습니다"

답)

```
import pandas as pd
```

```
def find_job():
```

```
    emp=pd.DataFrame.from_csv("C:\wwwpython_data\wwwemp.csv")
```

```
    name=""
```

```
    while name!="":
```

```
        name=input('월급을 알고 싶은 직원명을 입력하세요~ ')

```

.....

```
name=input('월급을 알고 싶은 사원명을 입력하세요~ ')
```

```
sal=emp[['sal']][emp['ename']==name.upper()].values[0]
```

```
if sal>=3000:
```

```
    raise Exception("해당 사원의 월급은 볼 수 없습니다")
```

```
else:
```

```
    return sal
```

```
print(find_job())
```

```
월급을 알고 싶은 사원명을 입력하세요~ smith  
[800]
```

```
월급을 알고 싶은 사원명을 입력하세요~ king
```

```
-----  
Exception                                     Tra  
<ipython-input-27-f267b9cf86e3> in <module>(  
    13         return sal  
    14  
--> 15 print(find_job())  
  
<ipython-input-27-f267b9cf86e3> in find_job(  
    9  
   10     if sal>=3000:  
--> 11         raise Exception("해당 사원의  
   12     else:  
   13         return sal
```

```
Exception: 해당 사원의 월급은 볼 수 없습니다
```

파이썬 149제(input,type)

2018년 7월 26일 목요일 오후 6:56

59. 사용자 입력받기(input)

```
ex)
k=input('값을 입력하세요:')
print('당신이 입력한 값은 <' + k + '> 입니다.')

값을 입력하세요:123
당신이 입력한 값은 <123> 입니다.
```

60. 자료형 확인하기(type)

```
ex)
numdata = 57
strdata = '파이썬'
listdata = [1, 2, 3]
dictdata = {'a':1, 'b':2}

def func():
    print('안녕하세요.')

print(type(numdata))
print(type(strdata))
print(type(listdata))
print(type(dictdata))
print(type(func))
<class 'int'>
<class 'str'>
<class 'list'>
<class 'dict'>
<class 'function'>
```

파이썬 149제 (%divmod)

2018년 7월 27일 금요일 오후 6:07

61. 나눗셈에서 나머지만 구하기(%)

```
ex)
a = 11113
b = 23
ret = a%b
print('<%d>를 <%d>로 나누면 <%d>가 나머지로 남습니다.' %(a, b, ret))
```

<11113>를 <23>로 나누면 <4>가 나머지로 남습니다.

62. 몫과 나머지 구하기(divmod)

```
ex)
a = 11113
b = 23
ret1, ret2 = divmod(a, b)
print('<%d/%d>는 몫이 <%d>, 나머지가 <%d>입니다.' %(a, b, ret1, ret2))
```

<11113/23>는 몫이 <483>, 나머지가 <4>입니다.

파이썬 149제(pandas)

2018년 7월 26일 목요일 오후 7:03

63. Pandas를 이용한 데이터 검색

Pandas란?

1. 판다스는 데이터 분석을 위한 파이썬 패키지
2. 엑셀의 스프레드시트와 관계형 데이터 베이스 데이터 처리능력이 뛰어나다.
3. 판다스는 DataFrame이라는 기본 자료구조를 사용한다.

문제165. Pandas 모듈을 이용해서 사원 테이블 전체를 출력하시오.

답)

```
import pandas as pd
```

```
emp=pd.read_csv("C:\python_data\emp.csv")
```

```
print(emp)
```

	empno	ename	job	mgr	hiredate	sal	comm	deptno
0	7839	KING	PRESIDENT	NaN	1981-11-17	5000	NaN	10
1	7698	BLAKE	MANAGER	7839.0	1981-05-01	2850	NaN	30
2	7782	CLARK	MANAGER	7839.0	1981-05-09	2450	NaN	10
3	7566	JONES	MANAGER	7839.0	1981-04-01	2975	NaN	20
4	7654	MARTIN	SALESMAN	7698.0	1981-09-10	1250	1400.0	30
5	7499	ALLEN	SALESMAN	7698.0	1981-02-11	1600	300.0	30
6	7844	TURNER	SALESMAN	7698.0	1981-08-21	1500	0.0	30
7	7900	JAMES	CLERK	7698.0	1981-12-11	950	NaN	30
8	7521	WARD	SALESMAN	7698.0	1981-02-23	1250	500.0	30
9	7902	FORD	ANALYST	7566.0	1981-12-11	3000	NaN	20
10	7369	SMITH	CLERK	7902.0	1980-12-09	800	NaN	20
11	7788	SCOTT	ANALYST	7566.0	1982-12-22	3000	NaN	20
12	7876	ADAMS	CLERK	7788.0	1983-01-15	1100	NaN	20
13	7934	MILLER	CLERK	7782.0	1982-01-11	1300	NaN	10

문제166. 이름과 월급을 출력하시오.

답)

```
import pandas as pd
```

```
emp=pd.read_csv("C:\python_data\emp.csv")
```

```
print(emp[ ['ename','sal'] ])
```

	ename	sal
0	KING	5000
1	BLAKE	2850
2	CLARK	2450
3	JONES	2975
4	MARTIN	1250
5	ALLEN	1600
6	TURNER	1500
7	JAMES	950
8	WARD	1250
9	FORD	3000
10	SMITH	800
11	SCOTT	3000
12	ADAMS	1100
13	MILLER	1300

64. Pandas를 이용해서 조건절 사용 데이터 검색 방법

문제167. 직업이 SALESMAN인 직원들의 이름과 직업을 출력하시오.

답)

```
import pandas as pd
emp=pd.read_csv("C:\python_data\emp.csv")
print(emp[ ['ename','sal'] ] [ emp['job']=='SALESMAN'] )
```

	ename	sal
4	MARTIN	1250
5	ALLEN	1600
6	TURNER	1500
8	WARD	1250

문제168. 이름이 신현수 학생의 이름과 나이를 출력하시오.

답)

```
import pandas as pd
emp3=pd.DataFrame.from_csv("C:\python_data\emp3.csv")
print(emp3[ ['ename','age'] ] [ emp3['ename']=='신현수' ])
```

	ename	age
empno		
22	신현수	27

65. Pandas를 이용해서 조건절 사용 데이터 검색방법

- 기타 비교 연산자를 오라클과 파이썬과 판다스와 비교

오라클	vs	파이썬	vs	판다스
between..and		<= & >=		[(emp['sal']>=1000) & (emp['sal'] <= 3000)] 도 가능하고 .between(,)

in		in		.isin[]
like		\$, \$, 정규식함수		apply 함수,lambda
is null		== ''		isnull()

문제169. 나이가 25에서 28사이인 학생들의 이름과 나이를 출력하시오.

힌트: emp['sal'].between[1000,3000]

답)

```
import pandas as pd
emp3=pd.DataFrame.from_csv("C:\python_data\emp3.csv")
print(emp3[ ['ename','age'] ] [ emp3['age'].between(25,28) ])
   ename  age
empno
1   윤진민   27
2   송윤호   27
4   김영토   28
6   김지우   27
7   정인중   27
8   이유진   25
9   백광흠   26
10  김원섭   26
11  김광록   27
13  지윤철   28
14  이상민   26
15  윤동환   25
16  이한새   25
17  방승준   26
18  신영근   25
20  유혜린   26
21  김건태   28
22  신현수   27
23  이근호   28
24  김대경   27
25  김동윤   28
26   도웅   28
27  이찬중   28
28  차호성   28
30  이광훈   25
```

문제170. 나이가 25에서 28사이가 아닌 학생들의 이름과 나이를 출력하시오.

답)

```
import pandas as pd
emp3=pd.DataFrame.from_csv("C:\python_data\emp3.csv")
print(emp3[ ['ename','age'] ] [ ~emp3['age'].between(25,28) ])
   ename  age
```

	ename	age
empno		
3	은혜 찬	29
5	정호 진	32
12	장은희	24
19	김근아	24
29	한지윤	24

문제171. 환경공학, 신학 인 학생들의 이름과 전공을 출력하시오.

답)

```
import pandas as pd
emp3=pd.DataFrame.from_csv("C:\python_data\emp3.csv")
print(emp3[ ['ename','major'] ] [ emp3['major'].isin(['환경공학','신학']) ])
#괄호를 잘 써줘야함
```

	ename	major
empno		
21	김건태	환경공 학
27	이찬중	신학

문제172. 성이 김씨인 학생들의 이름과 나이를 출력하시오.

답)

```
import pandas as pd
emp3=pd.DataFrame.from_csv("C:\python_data\emp3.csv")
print(emp3[ ['ename','age'] ] [ emp3['ename'].apply(lambda x:x[0]=='김' ) ] )
```

	ename	age
empno		
4	김영 토	28
6	김지 우	27
10	김원 섭	26
11	김광 록	27
19	김근 아	24
21	김건 태	28
24	김대 경	27
25	김동 윤	28

66. pandas를 이용한 조인

문제173. pandas를 이용해서 이름과 부서위치를 출력하시오.

답)

```
import pandas as pd
emp=pd.read_csv("C:\python_data\emp.csv")
dept=pd.read_csv("C:\python_data\dept.csv")
result=pd.merge(emp,dept,on='deptno')
print(result[['ename','loc']])
```

	ename	loc
0	KING	NEW YORK
1	CLARK	NEW YORK
2	MILLER	NEW YORK
3	BLAKE	CHICAGO
4	MARTIN	CHICAGO
5	ALLEN	CHICAGO
6	TURNER	CHICAGO
7	JAMES	CHICAGO
8	WARD	CHICAGO
9	JONES	DALLAS
10	FORD	DALLAS
11	SMITH	DALLAS
12	SCOTT	DALLAS
13	ADAMS	DALLAS

문제174. 부서위치가 DALLAS인 직원들의 이름과 부서위치를 출력하시오.

답)

```
import pandas as pd
emp=pd.read_csv("C:\python_data\emp.csv")
dept=pd.read_csv("C:\python_data\dept.csv")
result=pd.merge(emp,dept,on='deptno')
print(result[['ename','loc']][result['loc']=='DALLAS'])
```

	ename	loc
9	JONES	DALLAS
10	FORD	DALLAS
11	SMITH	DALLAS
12	SCOTT	DALLAS
13	ADAMS	DALLAS

문제175. 이름과 부서위치를 출력하는데 SQL의 outer join의 결과가 출력되게 하시오.

보기)

```
SQL> select e.ename, d.loc
      from emp e, dept d
      where e.deptno(+) = d.deptno;
```

답)

```
import pandas as pd
emp=pd.read_csv("C:\python_data\emp.csv")
dept=pd.read_csv("C:\python_data\dept.csv")
result=pd.merge(emp,dept,on='deptno', how='right') #left
print(result[['ename','loc']])
```

	ename	loc
0	KING	NEW YORK
1	CLARK	NEW YORK
2	MILLER	NEW YORK
3	BLAKE	CHICAGO
4	MARTIN	CHICAGO
5	ALLEN	CHICAGO
6	TURNER	CHICAGO
7	JAMES	CHICAGO
8	WARD	CHICAGO
9	JONES	DALLAS
10	FORD	DALLAS
11	SMITH	DALLAS
12	SCOTT	DALLAS
13	ADAMS	DALLAS
14	NaN	BOSTON

문제176. 아래의 full outer join을 pandas로 구현하시오.

보기)

```
SQL> select e.ename, d.loc
      from emp e full outer join dept d
      on( e.deptno=d.deptno);
```

답)

```
import pandas as pd
emp=pd.read_csv("C:\python_data\emp.csv")
dept=pd.read_csv("C:\python_data\dept.csv")
result=pd.merge(emp,dept,on='deptno', how='outer')
print(result[['ename','loc']])
```

	ename	loc
0	KING	NEW YORK
1	CLARK	NEW YORK
2	MILLER	NEW YORK
3	BLAKE	CHICAGO
4	MARTIN	CHICAGO
5	ALLEN	CHICAGO
6	TURNER	CHICAGO
7	JAMES	CHICAGO
8	WARD	CHICAGO
9	JONES	DALLAS
10	FORD	DALLAS
11	SMITH	DALLAS
12	SCOTT	DALLAS
13	ADAMS	DALLAS
14	NaN	BOSTON

67. pandas를 이용한 서브쿼리

문제177. JONES 보다 더 많은 월급을 받는 직원들의 이름과 월급을 출력하시오.

보기)

```
SQL> select ename,sal
      from emp
      where sal > (select sal
                  from emp
                  where ename='JONES');
```

답)

```
import pandas as pd
emp=pd.read_csv("C:\python_data\emp.csv")
```

```
js=emp[ ['sal'] ][emp['ename']=='JONES'].values[0][0] #[0]하나만쓰면 리스트
print(js)
```

```
print(emp[ ['ename','sal'] ][emp['sal']>=js] )
2975
```

	ename	sal
0	KING	5000
3	JONES	2975
9	FORD	3000
11	SCOTT	3000

문제178. SCOTT의 직속상사의 이름을 출력하시오.

보기)

```
SQL select ename
      from emp
      where empno=(select mgr
                  from emp
                  where ename='SCOTT');
```

답)

```
import pandas as pd
emp=pd.read_csv("C:\python_data\emp.csv")
```

```
smgr=emp[ ['mgr'] ][emp['ename']=='SCOTT'].values[0][0]
```

```
res=emp[ ['ename'] ][emp['empno']==smgr ]
print(res)
```

	ename
3	JONES

문제179. 관리자인 직원들의 이름을 출력하시오.

보기)

```
SQL> select ename
```

```

from emp
where empno in ( select mgr
                  from emp );

```

답)

```

import pandas as pd
emp=pd.read_csv("C:\python_data\emp.csv")

res=emp[ ['ename'] ] [emp['empno'].isin(emp['mgr'])]
print(res)

```

	ename
0	KING
1	BLAKE
2	CLARK
3	JONES
9	FORD
11	SCOTT

문제180. 관리자가 아닌 직원들의 이름을 출력하시오.

답)

```

import pandas as pd
emp=pd.read_csv("C:\python_data\emp.csv")

res=emp[ ['ename'] ] [~emp['empno'].isin(emp['mgr'])]
print(res)

```

	ename
4	MARTIN
5	ALLEN
6	TURNER
7	JAMES
8	WARD
10	SMITH
12	ADAMS
13	MILLER

문제181. 최대값을 받는 직원의 이름과 월급을 출력하시오.

보기)

```

SQL> select ename, sal
      from emp
      where sal=(select max(sal)
                  from emp);

```

답)

```

import pandas as pd
emp=pd.read_csv("C:\python_data\emp.csv")

res=emp[ ['ename','sal'] ] [emp['sal']==max(emp['sal'])]

```

```
print(res)
```

```
      ename  sal  
0  KING  5000
```

문제182. 가장 마지막으로 멤버쉽에 가입한 고객의 고객번호, 성별, 거주지역, 가입년월을 출력

하시오(마지막)

참고)

```
import pandas as pd
```

```
cust=pd.read_csv("C:\python_data\1cust.csv", engine='python')
```

```
memb=pd.read_csv("C:\python_data\5memb.csv", engine='python')
```

답)

```
import pandas as pd
```

```
cust=pd.read_csv("C:\python_data\1cust.txt")
```

```
memb=pd.read_csv("C:\python_data\5memb.txt")
```

```
cmjoin=pd.merge(cust,memb,on='고객번호')
```

```
res=cmjoin[['고객번호','성별','거주지역','가입년월']] [cmjoin['가입년월']  
월']=max(cmjoin['가입년월']) ]
```

```
print(res)
```

	고객번호	성별	거주지역	가입년월
0	11	1	100.0	201512
39	204	2	610.0	201512
54	279	2	100.0	201512
98	491	2	60.0	201512
125	573	2	13.0	201512
133	614	2	60.0	201512
151	697	1	49.0	201512
158	726	2	55.0	201512
177	802	2	55.0	201512
272	1244	2	24.0	201512
---	----	-	----	-----