



Python

🕒 작성일시	@2023년 3월 4일 오전 3:00
📄 강의 번호	더조은 컴퓨터학원
📄 유형	강의
📎 자료	
☑️ 복습	<input type="checkbox"/>

• Python 환경 설정

=> <https://www.python.org/> -> python 3.1.1 설치 (최초 체크박스 All 체크)

★★ 단축키 ★★

- 저장 : Ctrl + S
- 실행 : F5
- 문장정렬 : Ctrl + Alt = i

• if

=> 파이썬에서는 else if --> elif로 사용

=> 파이썬에서는 조건 대신 숫자 사용 가능 (0 : false , 0 이외의 숫자 true)

• 사용방법

if 조건식:
true 값 실행결과 (tab key 반드시 사용)
false 값 실행 결과 * if 조건식에 관계없이 무조건 실행된다

• 사용예시

```
if num > 0 :  
    print("양의 정수입니다.")  
  
1. if문 , if-else문 예제  
1-1.  
a=5  
if a>7:  
    print("풍당")  
    print("철썩")  
  
1-2.  
a=5  
if a<0:  
    print("풍당1")  
else:  
    print("풍당2")  
    print("철썩")  
  
1-3. 홀수 / 짝수  
x=int(input("입력 : "))  
if x%2==0:  
    print("짝수입니다")  
else:
```

```
print("출수입니다")
```

1-4. 성적구하기

```
x=int(input("성적 : "))
if x>=90:
    print("우수")
elif x>=80:
    print("보통")
elif x>=70:
    print("미달")
else:
    print("노력")
print("끝")
```

1-5. 큰수, 작은수

```
x=int(input("입력1 : "))
y=int(input("입력2 : "))
if x>y:
    print(x,"이 큼니다")
    print("%d이 큼니다"%x)
    print("{}이 큼니다".format(x))
    print(str(x)+"이 큼니다")
elif x<y:
    print(y,"가 큼니다")
else:
    print("x와 y는 같습니다")
    print("끝")
```

※ 파이썬에서는 증감연산자를 사용하지 않음

• 메인선언

```
def main():
    내용
main()
```

• 파이썬에서의 삼항연산자

=> print("True 일 때 실행") if a > 5 else print("False 일 때 실행") ※ 조건이 가운데 사용

• 파이썬에서의 for문

```
for i in [0, 1, 2] --> 리스트 , Java의 배열 개념과 비슷
    sum = sum + i
    print(sum)

for i in range(1, 11): --> range(시작점 , 끝지점 +1):
    ※ 단순 반복이 목적이라면 시작점을 0으로 표기하되, 시작점 0은 생략가능
    ※ 파이썬에서는 do ~ while 문으로 무한루프 사용 (for , while x)
```

• 파이썬 for문 중첩 예시

```
for i in range(3):
    print("-----")
    for j in range(3):
        print("[ " , i , " , " , " ]" , end = " ")
    print("")
```

• def : 파이썬에서 함수를 정의할 때 사용

=> def 함수명():

• input 기능

=> num = eval(input("정수 입력 : "))

=> int(), float() 함수 : 문자열로 입력받은 숫자값이 정수 또는 실수 변경이 가능할 때 사용

• 클래스

```
class MyClass:
    # 클래스 이름
    def __init__(self):
        # 호출할 내용이 없으면 생략가능
        print("생성자함수")
        a="안녕하세요"
    # 자바와 달리 __init__함수가 생성자함수로 고정되어 사용
    # 정적변수 : 클래스 바로 아래에 선언된 변수 (인스턴스 생성시 복사 불가)
    def talk(self):
        # 멤버함수
        self.a="하이"
    print(self.a)
    # self 로 사용시 인스턴스에서 자유롭게 사용할 수 있는 변수가 된다 (클래스변수)
    # 인스턴스.변수명으로 호출하면 self로 선언된 변수가 호출된다

ins=MyClass()
# 인스턴스 생성 과정 (생성자함수 호출)
# MyClass 호출 시 __init__ 함수가 실행된다
# MyClass 의 복사본을 ins에 저장

print(ins.a)
ins.talk()
ins.a="떠나립니다"
print(MyClass.a)
print(ins.a)
```

1. 클래스 예시

#1

```
class Calculator:
    def __init__(self,a,b):
        # a,b : 지역변수
        self.x=a
        self.y=b
    def add(self):
        print(self.x+self.y)
    def minus(self,one,two):
        return one+two

#2
import cla
temp=cla.Calculator(2,5)
temp.add()
result=temp.minus(1,1)
print(temp.x,temp.y)
print(result)
```

```
--> Class 까지 import
from cla import Calculator
temp=Calculator(2,5)
temp.add()
result=temp.minus(1,1)
print(temp.x,temp.y)
print(result)
```

1-1. Class 호출 (홀수, 짝수)

```
class Choice:
    def __init__(self):
        print("출발합니다")
    def oddeven(self,a):
        if a % 2 ==0:
            print("짝수입니다")
        else:
            print("홀수입니다")
```

```
ch = int(input("입력 : "))
temp=Choice()
temp.oddeven(ch)
```

1-2. Class 호출 (구구단)

```
class Good:
    def gugudan(self,x):
        for i in range(1, 10):
            print("%d * %d = %d" % (x, i, x * i))
            print("{} * {} = {}".format(x, i, x * i))
```

```
gu = int(input("구구단 입력 : "))
temp=Good()
```

```

temp.gugudan(gu)

1-3. Class 호출 + return 사용
class Cal:
    def plus(self,x,y):
        return x+y

    def minus(self,x,y):
        return x-y

a = int(input("입력1 : "))
b = int(input("입력2 : "))
r=Cal()
res1=r.plus(a,b)
res2=r.minus(a,b)
print("더한 값 %d   뺀 값 %d"%(res1, res2))

2. 함수 추가 호출
#1
class Calculator:
    def __init__(self,a,b):
        self.x=a
        self.y=b
    def add(self):
        print(self.x+self.y)
    def minus(self,one,two):
        return one+two

def echo():
    print("Hello")

#2
import cla
temp=cla.Calculator(2,5)
temp.add()
result=temp.minus(1,1)
print(temp.x,temp.y)
cla.echo()
print(result)

3. 패키지 호출
#1
def echo():
    print("Hi")

def echo2():
    print("HiHi")

#2
import mount.san
x=mount.san.echo()
y=mount.san.echo2()

--> 간소화
from mount import san
san.echo()
san.echo2()

※ 메인함수에 종속되어 import 되어 실행되는 내용 중 실행하지 않아야 할 파일이 있을 경우
메인함수가 실행되는 경우에만 실행
if __name__=="__main__":
    print("실행되면 안됩니다")

4. 모듈 호출 (구구단)
#1
class Good:
    def gugudan(self,x):
        for i in range(1, 10):
            print("%d * %d = %d" % (x, i, x * i))
            print("{} * {} = {}".format(x, i, x * i))

#2
from test3 import Good
gu = int(input("구구단 입력 : "))
temp=Good.Good()
temp.gugudan(gu)

```

• 지역변수

=> 자기자신의 함수안에서만 사용가능한 함수

```
class Cal:
    def __init__(self):
        print("Ping")
    def add(x,y):
        return x+y
a=Cal()
res=a.add(5,6)
```

```
class Cal:
    def __init__(self):
        self.a=9
    def echo(self):
        print(a)
    def echo2(self):
        a=5
        self.a=8
win=Cal()
win.echo()
win.echo2()
print(win.a)
```

※ 문제 풀이 및 참고 설명

1. 출력문

```
print("hello")
print("hello2")

print("hello",end="_")
print("hello",end="")
print("hello3")
print("hello4")

a=5
print("fail","back")
print("fail"+"back")
print("fail"+"back",sep="-")      # 결합한 하나의 문자열이기에 불가
print("fail","back",sep="-")
print("정수",a,"이다")
print("정수"+a+"이다")           # 결합은 하나의 문자열이 됨을 의미 , 정수형은 결합 불가
print("정수"+str(a)+"이다")

#print("연산시작")
a=5
b=7
print(a,"+",b,"=",a+b)
print("%d"%a)                   # 1개 대입은 그냥 써도 되나 아래처럼 2개 이상 대입시는 튜플화 시켜준다.
#print("%d + %d = %d"%a,b,a+b)  # 이렇게 사용하면 안된다.
print("%d + %d = %d"%(a,b,a+b))

a=3.141592000
b=2
print(a+b)
print("%f"%a)
print("%.2f + %d = %10.3f"%(a,b,a+b))  # 3번째 자리 반올림
                                         # %10.3f : 열칸만큼 간격을 두고 오른쪽에 붙여서 출력
                                         # -가 붙을 경우 왼쪽에 붙여서 출력

apple=2
banana="three"
print("i have {0} apple and {1} banana".format(3,4))
# {0}, {1} 자리에 3,4 대입 *{}로 지정해도 되지만 번지순서대로 하기위해 값을 순차 지정
print("i have {0} apple and {1} banana".format(apple,banana))
# 형식 상관없이 format 함수를 이용하여 출력 가능
print("i have {apple} apple and {banana} banana".format(apple=3,banana="four"))
# 지정한 속성에 값을 넣어 출력(변수 선언)

print("i have {0:<10} apple and {1} banana".format(3,4))  # <10 : %-10d 와 동일한 기법
print("i have {0:>10.2f} apple".format(3.145592))
# >10.2f : 10칸만큼 오른쪽에 붙여 출력 , 소수점 2자리까지 , {0:.2f}
# %.d로 표현 가능
```

```

inch=5
print("%d 인치는 %.1f cm입니다"%(5,2.5*5))
print("{0} 인치는 {1:.1f} cm입니다".format(5,2.5*5))

2. 입력문
val1 = input("what :")      # input : 실행과 동시에 키보드 깜빡임 -> 키보드로 입력 받은 값 저장
print(type(val1), val1)    # input 함수로 입력받은 모든 값은 문자열로 취급

val2 = int(input("what :"))  # x=int val2
                             # print(val2)
print(type(val2), val2)     # 정수값 출력

val3 = int(val1)+val2
print("result",val3)

2-1. 입력문 예제
val1 = int(input("입력1 :"))
val2 = int(input("입력2 :"))

print("%d + %d = %d"%(val1,val2,val1+val2))
print("{} + {} = {}".format(val1,val2,val1+val2))

inch=int(input("inch : "))
print("%.2f cm입니다"%(inch*2.5))
print("{0:.2f} cm입니다".format(inch*2.5))

3. 몫과 나머지
val1 = int(input("입력 :"))
mok = int(val1/2)
nam = int(val1%2)
print("몫은 %d 나머지는 %d 입니다"%(mok,nam))
print("몫은 {} 나머지는 {} 입니다".format(mok,nam))

4. 사칙연산
val1 = int(input("입력1 :"))
val2 = int(input("입력2 :"))
sum = int(val1+val2)
mius = int(val1-val2)
p = int(val1*val2)
y = int(val1/val2)
print("%d + %d = %d 입니다"%(val1,val2,sum))
print("%d - %d = %d 입니다"%(val1,val2,mius))
print("{0} * {1} = {2} 입니다".format(val1,val2,p))
print("{0} / {1} = {2} 입니다".format(val1,val2,y))

5. 리스트형 인덱싱
a =[1,2,3,4,5,6,7]      # []: 리스트 (인접하여 연속적으로 데이터 저장 목적)
print(a)                # C, JAVA에서의 배열과 동일, a[0] : 인덱스, 첨자
print(a[0],a[-1])       # 역순으로 출력시 -1부터 시작
print(a[0:2])           # 요소 앞자리라는 것을 기억  // : ~에서 ~까지 , 0~1요소까지 출력
print(a[0:-2])

--> a[]
    a.append() 의 형태로 많이 사용
ex) a =int(input("입력 : "))
    a.append(a)

b=[[1,2],[3,4],[5,6]]
print(b[0][1],b[-1][0])
c=[[1,2,3],4,[5,6,7]]
print(c[1])
print(c[0][-2])

6. append (실시간 값 추가)
a=[]                    # 리스트 = 배열
x=int(input("입력1 :"))
a.append(x)
print(a)

y=int(input("입력2 :"))
a.append(y)
print(a)

z=int(input("입력3 :"))
a.append(z)
print(a)

7. 값 교환

```

```

a=[10,20,30,40]
temp=None          # None : 아무값도 넣지 않은 경우
print(a)
temp=a[0]
a[0]=a[3]
a[3]=temp
print(a)

8. 넓이 계산
x=int(input("가로 : "))
y=int(input("세로 : "))
z=x*y
print("사각형 넓이는 {} 입니다".format(z))
print("사각형 넓이는 %d 입니다"%z)

9. 원 둘레 계산
p=3.141592
r=float(input("반지름 : "))
c=2*p*r
d=p*r*r
print("원의 둘레 %.2f 너비 %.2f"%(c,d))
print("원의 둘레 {0:2f} 너비는 {1:2f}".format(c,d))

10. 반복문 예제
10-1. 반복문 #1
sum=0
for i in range(1,6):
    print(i, "를 더합니다")
    sum=sum+i

print("1부터 %d까지합은 %d입니다"%(i, sum))

10-2. 구구단
x=int(input("구구단 입력 : "))
for i in range(1,10):

    print("%d * %d = %d" %(x,i,x*i))
    print("{} * {} = {}".format(x,i,x*i))

10-3. 반복문(리스트)
sum=0
a=[10,20,30,40,50]
for i in a:
    sum = sum+i
print(sum)

10-4. 반복문(리스트 카운트)
count=int(input("개수 : "))
a=[]
for i in range(1,count+1):
    temp=int(input("입력 : "))
    a.append(temp)

sum=0

for i in a:
    print(i,end=" ")
    print(i,end=" , ")
    sum=sum+i
print()
print(sum)

10-5. 최대값, 최소값
a=[15,37,25,1]
max = 0;
for i in range(0,4):
    if max < a[i]:
        max = a[i]

print("최대값 :", max)

min = 1000
for i in range(0,4):
    if min > a[i]:
        min = a[i]
print("최소값 :", min)

11. 랜덤값
import random

```

```

x=random.randint(1,10)
print(x)

11-1. 랜덤값(리스트)
import random
array=[]
for i in range(0,5):
    array.append(random.randint(1,20))
print(array)

for i in array:
    print(i,end=" ")

11-2. 작은값 ~ 큰값까지 누적
min=int(input("작은 값 : "))
max=int(input("큰 값 : "))
sum=0

for i in range(min,max+1):
    sum=sum+i
print(sum)

11-3. 두개의 값 비교 덧셈
#1
a=int(input("첫번째 값 : "))
b=int(input("두번째 값 : "))

if a < b:
    min = a
    max = b

else:
    min = b
    max = a
sum=0
for i in range(min, max+1):
    sum=sum+i
print(sum)

#2
a=int(input("첫번째 값 : "))
b=int(input("두번째 값 : "))
sum=0

if a<b:
    for i in range(a,b+1):
        sum=sum+i
        print(sum)

else:
    for i in range(b,a+1):
        sum=sum+i
        print(sum)

11-4. 입력갯수만큼 랜덤값 추가 후 출력
import random
x=random.randint(0,100)
count=int(input("개수 : "))
array=[]
for i in range(1,count+1):
    temp=random.randint(0,100)
    array.append(temp)

for i in array:
    print(i, end=" ")

# print(array.__len__()) # 리스트에 나온 결과값(개수) 출력

11-5. 입력갯수만큼 랜덤값 추가 후 총점, 평균 구하기
import random
x=random.randint(0,100)
count=int(input("개수 : "))
array=[]
for i in range(1,count+1):
    temp=random.randint(0,100)
    array.append(temp)

for i in array:
    print(i, end=" ")

```



```

print()
sum=0
for i in array:
    sum=sum+i

number=array.__len__()
print("총점 %d 평균 %.1f"%(sum,sum/number))

```

11-6. 과일 선택

```

#1
print("apple : 150")
print("banana : 100")
print("lemon : 120")

fruits=input("과일 선택 :")
count=int(input("수량 : "))

if fruits=="apple":
    sum=count*150
elif fruits=="banana":
    sum=count*100
elif fruits=="lemon":
    sum=count*120
else:
    print("팔지 않는 과일입니다")
    exit()
print("구매금액 : %d원 입니다"%sum)

```

```

#2
print("apple : 150")
print("banana : 100")
print("lemon : 120")

fruits=input("과일 선택 :")
count=int(input("수량 : "))

if fruits=="apple":
    sum=count*150
elif fruits=="banana":
    sum=count*100
elif fruits=="lemon":
    sum=count*120
else:
    print("팔지 않는 과일입니다")
    exit()
print("구매금액 : %d원 입니다"%sum)

pay=int(input("지불액 : "))
print("거스름돈은 %d 입니다"%(pay-sum))

```

12. 이중 if문

```

for i in range(1,10):
    for j in range(1,10):
        if j==5:
            break
        print(i,"행",j,"열")

```

12-1. 약수

```

a=int(input("입력 : "))
for i in range(1,a+1):
    if a%i==0:
        print(i,end=" ")

```

12-2. 공약수

```

x=int(input("입력1 : "))
y=int(input("입력2 : "))

if x<y:
    min=x
else:
    min=y

for i in range(1,min+1):
    if (x%i==0 and y%i==0):
        print(i,end=" ")

```

12-3. 최대공약수

```

x=int(input("입력1 : "))

```

```

y=int(input("입력2 : "))

if x<y:
    min=x

else:
    min=y

for i in range(1,min+1):
    if (x%i==0 and y%i==0):
        max=i
print(max)

12-4. 소수 (flag)
x=int(input("입력 : "))
flag=0
for i in range(2,x+1):
    if (x%i==0 and x!=i):
        flag=1
if flag==0:
    print("소수")
else:
    print("소수가 아닙니다")

12-5. 소수 (for ~ else)
x=int(input("입력 : "))
for i in range(2,x+1):
    if (x%i==0 and x!=i):
        print("소수가 아닙니다")
        break
else:
    print("소수")

13. while문
t=0                # 초기값 설정 필요
while t<7:
    t=t+1           # 무한루프를 방지하기 위해 필요
    if t<3:
        continue
    if t==6:
        break
    print(t)

13-1. 랜덤값 중 최대값에 해당하는 인덱스 값 출력
import random
array = []
for i in range(0, 5):
    array.append(random.randint(1, 100))

for i in range(0, 5):
    print(array[i],end=" ")
print()

max = -1;
for i in range(0,5):
    if max < array[i]:
        max = array[i]
        index=i
print("인덱스 : ",index)
print(a[index])

13-2. 랜덤값 중 최대값에 해당하는 인덱스 값과 0번째 인덱스 값 교환
import random
array = []
for i in range(0, 5):
    array.append(random.randint(1, 100))

for i in range(0, 5):
    print(array[i],end=" ")
print()

max = -1;
for i in range(0,5):
    if max < array[i]:
        max = array[i]
        index=i
print("인덱스 : ",index)
print(array[index])

```

```

temp=None
temp=array[0]
array[0]=array[index]
array[index]=temp

for i in range(0, 5):
    print(array[i],end=" ")

13-3. 점수 중 최대값에 해당하는 인덱스 출력
jum=[100,97,26,54]
name=["a","b","c","d"]
max=-1
for i in range(0,4):
    if max < jum[i]:
        max = jum[i]
        index=i
print("인덱스 : ",name[index])

14. for문 + 입력한 리스트 추가
a=[]
for i in range(1,4):
    print("%d 번째 :"%i,end=" ")
    temp=int(input())
    a.append(temp)

for i in range(0,3):
    print(a[i],end=" ")

14-1. for문 + 랜덤값 리스트 추가
import random
a = []
for i in range(1, 4):
    temp=random.randint(0, 100)
    a.append(temp)

for i in range(0,3):
    print(a[i], end=" ")

```

• 2차원 리스트 선언

```

# a=[[1,2],[3,4],[5,6]] 형태로 만들기
* a=[[1,2]] -> a[0,0],a[0,1]
a=[]
s=[]
s.append(1)
s.append(2)
a.append(s)
s=[]
s.append(3)
s.append(4)
a.append(s)
s=[]
s.append(5)
s.append(6)
a.append(s)

```

• 문자열 리스트

```

str="apple"
for t in str:
    print(t)
print(t[1:-2])

# 키값 출력
dic={1:"apple",2:"banana",3:"lemon"}
for i in dic:
    print(i,":",dic[i])    # 사전인 경우는 key가 i에 들어갑니다.

# continue
array=[1,2,3,4,5,6]
for i in array:
    if i == 3:
        continue    # 계속 진행하지 않고 바로 상단의 반복문으로 이동
    print(i)

```

```

# break
#1
array=[1,2,3,4,5,6]
for i in array:
    if i == 3:
        break          # 자신을 포함하고 있는 반복문을 중지한다
    print(i)

#2
array=[1,2,3,4,5,6]
for i in array:
    print(i)            # break 같은 중단됨 없이 정상적으로 끝났을 경우에 else 구문 실행
else
    print("the end")

#3
array=[1,2,3,4,5,6]
for i in array:
    if i==5:
        break
    print(i)
else:
    print("the end")

```

• 튜플 & dictionary

```

a(10,20,30,40)          # 튜플 : 값을 호출할 때만 사용(수정불가)
c={'1':"apple", 2:"banana"} # dictionary - {key : value} 형태로 사용
print(c["1"])

1. dictionary 예시
a={'a':"apple", 'b':"banana"}
print(a['a'])
print(a['b'])

```

• 버블정렬 : 인접한 값끼리 비교하여 작은값일 경우 교환한다

=> 1 싸이클 : 가장 큰 값이 마지막에 위치한다

=> 싸이클 횟수 : 비교하는 요소 -1

```

1. 버블정렬(오름차순)
import random
array = []
for i in range(0, 5):
    array.append(random.randint(1, 100))

for i in range(0, 5):
    print(array[i],end=" ")
print()

for count in range (0,4):
    for i in range (0,4):
        if array[i] > array[i+1]:
            temp=array[i]
            array[i]=array[i+1]
            array[i+1]=temp
    print(array)

1-1. 버블정렬(내림차순)
import random
array = []
for i in range(0, 5):
    array.append(random.randint(1, 100))

for i in range(0, 5):
    print(array[i],end=" ")
print()

for count in range (0,4):
    for i in range (0,4):
        if array[i] < array[i+1]:
            temp=array[i]

```

```

        array[i]=array[i+1]
        array[i+1]=temp
print(array)

2. 선택정렬
작은값을 순차적으로 배치

2-1. 선택정렬
import random
array = []
for i in range(0, 5):
    array.append(random.randint(1, 100))

for i in range(0, 5):
    print(array[i],end=" ")
print()

for count in range (0,4):
    for i in range (count+1,5):
        if array[i] < array[count]:
            temp=array[i]
            array[i]=array[count]
            array[count]=temp
print(array)

```

• 함수

```

def add1(x,y):      # 함수에 대한 정의 , x : 매개변수(호출과 동시에 넘어온 인수를 받기위한 변수)
    return x+y      # 실행과 동시에 종료
                    # x,y : 지역변수 (선언된 함수안에서만 사용 후 상쇄)
                    # return 이 있을 경우 return 값을 저장할 변수 필요

def add2(x,y):      # add2(a,b) 가 호출되는 시점에 실행
    print(x+y)

z=add1(3,5)         # 선언된 함수가 호출되는 시점, 인수 = argument
                    # return 받은 값(return x+y)을 받아온다

print(z)
a=3
b=8
add2(a,b)           # return 이 없을 경우 호출 (void 형)

```

1. 함수 예제

1-1. 함수호출 (구구단)

```

def gugudan(x):
    for i in range(1,10):
        print(x,"*",i,"=",x*i)

gu=int(input("출력할 구구단 : "))
gugudan(gu)

```

1-2. 최대값 호출 (return 사용)

```

def max(x,y):
    max_num = 0
    if max_num < x:
        max_num = x

    if max_num < y:
        max_num = y
    return max_num

num1=int(input("입력1 : "))
num2=int(input("입력2 : "))
z=max(num1,num2)
print("결과는 %d"%z)

```

1-3. 원의 둘레 / 너비 계산 (함수호출)

```

def circle(r):
    print("원의 둘레 %.2f 너비 %.2f" % (2*3.14*r, 3.14*r*r))
    print("원의 둘레 {0:2f} 너비는 {1:2f}".format((2*3.14*r),(3.14*r*r)))

r = float(input("반지름 : "))
circle(r)

```

1-4. 원의 둘레 / 너비 계산 (return 사용)

```

def dul(r):

```

```

    return 2 * 3.14 * r

def nuby(r):
    return 3.14 * r * r

ban = float(input("반지름 : "))
res1=dul(ban)
res2=nuby(ban)
print("원의 둘레 {0:2f} 너비는 {1:2f}".format(res1,res2))

--> print("원의 둘레 {0:2f} 너비는 {1:2f}".format(res1=dul(ban),res2=nuby(ban)))

* 호출 활용도(return) 에 따라 다른방법으로 사용 가능

1-5. 약수 구하기 (함수호출)
def yak(num):
    for i in range(1, num + 1):
        if num % i == 0:
            print(i, end=" ")

a=int(input("입력 : "))
yak(a)

★ 문제 41) 랜덤값 중 최대값 출력 (return 사용)
import random
def max(z):
    num = 0
    for i in z:
        if num < i:
            num = i
    return num

a = []
for i in range(1, 4):
    temp=random.randint(0, 100)
    a.append(temp)

for i in range(0,3):
    print(a[i], end=" ")

res=max(a)
print()
print("최대값은 %d"%res)

# 시간 대기
import time
print("3초간 대기")
time.sleep(3)
print("3초가 지났습니다")

# 함수 호출
def get_list(rec):
    print("받은 리스트를 출력합니다")
    print(rec)

sen=[]
for i in range(5):
    sen.append(i)

print(sen)
get_list(sen)

```

• 파일 내용 출력 #1 (파일 포인터 = 핸들러)

=> 모든 내용 한번에 출력

=> 편집하는 파일과 동일한 경로에 Text 문서가 존재해야함

```

f=open("data.txt","r")
data=f.read()
print(data)
f.close()

f=open("c:/test2/img/data.txt","r")    # 절대경로
f=open("img/data.txt","r")             # 상대경로

```

```

# 모드 : r - 읽기, w = 쓰기 (파일이 없을경우 생성, 파일이 존재할 경우 덮어쓴다), a = 쓰기 (동일 파일이 존재할 경우 추가)

data=f.read()
print(data)
f.close()

* 홈페이지를 만들경우 외부에서 접속하는 경로 "주소\index.html"
* 대부분 상대경로를 사용함

1. 파일 내용 출력 예시

f=open("data.txt","r")
line_number=1
line=f.readline() # 한줄씩 호출
while line: # 값이 존재하면 true
    print("{0} : {1}".format(line_number, line),end="") # {0} : line_number=1
    line=f.readline()
    line_number+=1 # line number 증가
f.close()

1-1. 리스트 형태 출력, 인덱스 값 출력
f=open("data.txt","r")
lines=f.readlines() # 저장형식 : ['파일내용1\n','파일내용2\n'] * 리스트형태의 출력
# 라인 1줄이 각각 하나의 요소가 된다

print(lines)
for line_number,line in enumerate(lines):
# enumerate는 리스트형 값들 앞에 각각 인덱스를 붙여준다
# 리스트의 값과 인덱스값을 같이 출력
    print("{0} {1}".format(line_number+1, line),end="")
f.close()

1-2. 파일 내용 출력 (쓰기)
#1
text=input('내용입력')
f=open("input_data.txt","w")
#"a" 옵션으로 열면 파일이 있으면 내용을 추가하고 없으면 새로 만든다 (덮어쓰기)
f.write(text)
f.close()

#2
text=input('내용입력')
f=open("input_data.txt","a")
#"a" 옵션으로 열면 파일이 있으면 내용을 추가하고 없으면 새로 만든다
f.write(text)
f.close()

#3
data=[]
f=open("data.txt","w")
print("입력을 끝내려면 stop 입력")
while True:
    text=input("내용 입력")
    if text=="중지":
        break
    data.append(text+'\n')
f.writelines(data)
f.close()

1-3. 파일 쓰기 + for문
f=open("output.txt","w")
count=int(input("인원수 : "))
for i in range(0,count):
    name=input("성명 : ")
    jumsu=int(input("점수 : "))
    total=name+","+str(jumsu)+"\n"
    f.write(total)
f.close()

# split (구분자)
jaro="apple,75,85,55"
res=jaro.split(",")
print(res) # 출력되는 각각의 토큰은 문자열이기 때문에 숫자로 표기하고 싶을 경우
# int함수를 통해 정수형 변환 작업 필요

1-4. 파일 전체 한번에 읽기
f = open("job/jaro.txt", "r")
all=f.read() # 파일을 읽어 올때 \n까지 읽어 드린다
print(all)
f.close()

```

```

1-5. 파일 전체 한줄씩 읽기
f = open("job/jaro.txt", "r")
line=f.readline()
while line:
    print(line,end="")
    line=f.readline()
f.close()

1-6. 파일 리스트 형태로 읽기
f = open("job/jaro.txt", "r")
all=f.readlines()
for line in all:
    print(line,end="")
f.close()

1-7. 파일 쓰기 (성적평균 구하기)
f = open("job/jaro.txt", "w")
name = input("성명 : ")
gumok1 = int(input("국어 : "))
gumok2 = int(input("영어 : "))
gumok3 = int(input("수학 : "))
sum=gumok1 + gumok2 + gumok3
avg = sum/3
f.write("{0},{1},{2},{3},{4},{5:.2f}\n".format(name,gumok1,gumok2,gumok3,sum,avg))
--> 평균값에 대한 인덱스로 인해 다른 항목도 인덱스 모두 기재
f.close()

1-8. 파일 쓰기 (입력한 인원수만큼 성적평균 구하기)
f = open("job/jaro.txt", "w")

count = int(input("인원수 : "))
for i in range(0, count):
    name = input("성명 : ")
    gumok1 = int(input("국어 : "))
    gumok2 = int(input("영어 : "))
    gumok3 = int(input("수학 : "))
    sum=gumok1 + gumok2 + gumok3
    avg = sum/3
    f.write("{0},{1},{2},{3},{4},{5:.2f}\n".format(name,gumok1,gumok2,gumok3,sum,avg))
f.close()

1-9. 선택에 따른 파일쓰기 추가
f = open("job/jaro.txt", "w")
ch='y'
while ch=='y':
    name = input("성명 : ")
    gumok1 = int(input("국어 : "))
    gumok2 = int(input("영어 : "))
    gumok3 = int(input("수학 : "))
    sum = gumok1 + gumok2 + gumok3
    avg = sum / 3
    f.write("{0},{1},{2},{3},{4},{5:.2f}\n".format(name, gumok1, gumok2, gumok3, sum, avg))
    ch=input("계속?(y or n)")
f.close()

1-10. 선택에 따른 파일쓰기 추가 (writelines 사용)
f = open("job/jaro.txt", "w")
ch='y'
res=[]
while ch=='y':
    name = input("성명 : ")
    gumok1 = int(input("국어 : "))
    gumok2 = int(input("영어 : "))
    gumok3 = int(input("수학 : "))
    sum = gumok1 + gumok2 + gumok3
    avg = sum / 3
    res.append("{0},{1},{2},{3},{4},{5:.2f}\n".format(name, gumok1, gumok2, gumok3, sum, avg))
    ch=input("계속?(y or n)")
f.writelines(res)
f.close()

1-11. 선택에 따른 파일 읽기 (readlines 사용)
f = open("job/jaro.txt", "r")
ch='y'
all=f.readlines()
for i in all:
    print(i,end="")
    ch = input("계속?(y or n)")

```



```

        if ch=='n':
            break
    f.close()

1-12. 선택에 따른 파일 읽기 (while 문)
f = open("job/jaro.txt", "r")
ch='y'
line=f.readline()
while line and ch=='y':
    print(line)
    ch = input("계속?(y or n)")
    line = f.readline()
f.close()

1-13. 선택에 따른 파일 읽기 (break)
f = open("job/jaro.txt", "r")
ch='y'
line=f.readline()
while line=='y':
    print(line)
    ch = input("계속?(y or n)")
    if ch=='n':
        break
    line = f.readline()
f.close()

1-14. 이미지 복사
bufsize=1024
f=open('img/sample.jpg', 'rb')          # rb : 2진모드, 비트로 읽고 쓰기
p=open("img/dest.jpg", "wb")
data=f.read(bufsize)                    # 1KB 크기로 쪼개서 읽어온다. 1MB 단위라면 1024*1024
while data:
    p.write(data)
    data=f.read(bufsize)
f.close()
p.close()

1-15. 파일 읽기 -> 리스트 형태 출력 -> 버블정렬(오름차순)
f = open("source.txt", "r")
temp = f.read()
a=temp.split(",")
print(a)
r=[]
for i in a:
    r.append(int(i))
print(r)

for i in range(1,r.__len__()):
    for j in range(0,r.__len__()-1):
        if r[j] > r[j+1]:
            tmp=r[j]
            r[j]=r[j+1]
            r[j+1]=tmp
print(r)
f.close()

# 빈문자열 선언 후 내용 추가
res=""                                # 빈문자열 선언
count=1
for i in range(0,r.__len__()):
    res=res+str(r[i])
    if count <= r.__len__():
        res=res+", "
    count=count+1
print(res)

1-16. 파일 읽기 -> 리스트 형태 출력 -> 버블정렬(오름차순) -> 내용추가 -> 파일 쓰기
f = open("source.txt", "r")
temp = f.read()
a=temp.split(",")
print(a)
r=[]
for i in a:
    r.append(int(i))
print(r)

for i in range(1,r.__len__()):
    for j in range(0,r.__len__()-1):
        if r[j] > r[j+1]:

```

```

        tmp=r[j]
        r[j]=r[j+1]
        r[j+1]=tmp

print(r)
res=""
count=1
for i in range(0,r.__len__()):
    res=res+str(r[i])
    if count <= r.__len__():
        res=res+", "
    count=count+1
print(res)

p=open("dest.txt", "w")
p.write(res)
f.close()
p.close()

1-17. 선택에 따른 파일 쓰기 + 내림차순 정렬
f = open("dest.txt", "w")
ch='y'
while ch != 'n':
    name = input("성명 : ")
    jumsu = input("점수 : ")
    temp=name+", "+jumsu+"\n"
    f.write(temp)
    ch=input("계속?(y or n)")
f.close()
p = open("dest.txt", "r")
r=p.readlines()
print(r)
digit=[]
for i in r:
    temp=i.split(",")
    digit.append(int(temp[1]))
print(digit)

for i in range(1,digit.__len__()):
    for j in range (0,digit.__len__()-1):
        if digit[j] < digit[j+1]:
            temp=digit[j]
            digit[j]=digit[j+1]
            digit[j+1]=temp

        temp=r[j]
        r[j]=r[j+1]
        r[j+1]=temp

print(r)
zp=open("data2.txt", "w")
zp.writelines(r)
p.close()
zp.close()

```

• 화이트 스페이스

=> \n : 줄바꿈 , \t : 간격조정

=> writelines() : 반드시 ()안에 리스트 요소가 들어가야 함

```

data="my hobby is fishing\nand you?\nlet's go to fishog tommorow"    # \의 경우 줄바꿈 기능
f=open("data.txt", "w")
f.write(data)
f.close

```

```

1. split
f=open("output.txt", "r")
line = f.readline()

while line:
    sep = line.split(',')
    readname = sep[0]
    readjumsu = int(sep[1])
    print(readname, readjumsu + 100)
    line = f.readline()
f.close()

```

```

1-1. split split + if ~ elif 문
ch=int(input("선택(1.입력/2.출력) : "))
if ch==1:
    f = open("output.txt", "w")
    count = int(input("인원수 : "))
    for i in range(0, count):
        name = input("성명 : ")
        jumsu = int(input("점수 : "))
        total = name + "," + str(jumsu) + "\n"
        f.write(total)
    f.close()
elif ch==2:
    f=open("output.txt", "r")
    line = f.readline()

    while line:
        sep = line.split(',')
        readname = sep[0]
        readjumsu = int(sep[1])
        print(readname, readjumsu)
        line = f.readline()
    f.close()
else:
    print("1이나 2중 입력하세요")
    exit()

1-2. split split + 함수호출
def inp():
    f = open("output.txt", "w")
    count = int(input("인원수 : "))
    for i in range(0, count):
        name = input("성명 : ")
        jumsu = int(input("점수 : "))
        total = name + "," + str(jumsu) + "\n"
        f.write(total)
    f.close()

def out():
    f = open("output.txt", "r")
    line = f.readline()

    while line:
        sep = line.split(',')
        readname = sep[0]
        readjumsu = int(sep[1])
        print(readname, readjumsu)
        line = f.readline()
    f.close()

ch=int(input("선택(1.입력/2.출력) : "))
if ch==1:
    inp()

elif ch==2:
    out()

else:
    print("1이나 2중 입력하세요")
    exit()
f.close()

1-3. split writelines
count = int(input("인원수 : "))
f = open("job/jaro.txt", "w")
data=[]
for i in range(0, count):
    name = input("성명 : ")
    jumsu = input("점수 : ")
    total = name + "," + jumsu + "\n"
    data.append(total)
f.writelines(data)
f.close()

```

• 데이터베이스 연동

```

import pymysql

con=pymysql.connect(host='localhost',user='root',passwd='1234',db='chic')    # 서버 접속 설정
cur=con.cursor()
# 커서 깜빡임 (제어권, 접속권한 획득)
sql="insert into student (name,kor,gen)values(%s,%s,%s)"
cur.execute(sql,a,b,c)
con.commit()    # 입력값 전송
con.close()

1. DB반영
1-1. DB에 항목 추가
import pymysql
a=input("이름은 : ")
b=input("성적 : ")
c=input("성별 : ")
con=pymysql.connect(host='localhost',user='root',passwd='1234',db='chic')
cur=con.cursor()
sql="insert into student (name,kor,gen)values(%s,%s,%s)"
cur.execute(sql,(a,b,c))
con.commit()
con.close()

1-2. DB에 항목 삭제
import pymysql
a=input("삭제할 이름은 : ")
con=pymysql.connect(host='localhost',user='root',passwd='1234',db='chic')
cur=con.cursor()
sql="delete from student where name=%s"
cur.execute(sql,a)
con.commit()
con.close()

1-3. 자판기
import pymysql
a=input("제품명 : ")
b=int(input("가격 : "))
c=int(input("수량 : "))

con=pymysql.connect(host='localhost',user='root',passwd='1234',db='chic')
cur=con.cursor()
sql="insert into dabang (name,price,count,sum)values(%s,%s,%s,%s)"
cur.execute(sql,(a,b,c,b*c))
con.commit()
con.close()

1-4. 자판기 추가, 삭제 (함수이용)
import pymysql
def add():
    import pymysql
    a = input("제품명 : ")
    b = int(input("가격 : "))
    c = int(input("수량 : "))

    con = pymysql.connect(host='localhost', user='root', passwd='1234', db='chic')
    cur = con.cursor()
    sql = "insert into dabang (name,price,count,sum)values(%s,%s,%s,%s)"
    cur.execute(sql, (a, b, c, b * c))
    con.commit()
    con.close()

def remove():
    import pymysql
    a = input("제품명 : ")

    con = pymysql.connect(host='localhost', user='root', passwd='1234', db='chic')
    cur = con.cursor()
    sql = "delete from dabang where name=%s "
    cur.execute(sql, a)
    con.commit()
    con.close()

print("1. 삽입")
print("2. 삭제")
print("3. 종료")
cho=input("선택 :")
while True:
    if cho=="1":

```

```

        add()
    elif cho=="2":
        remove()
    elif cho=="3":
        break
    print("1. 삽입")
    print("2. 삭제")
    print("3. 종료")
    cho = input("선택 :")

# 경로 진입
#1
import time                                # 실행할때 도움이 되는 함수 또는 모듈로 구성 * 기본적으로 모듈까지는 경로 설정
time.sleep(5)
print("5초가 지났습니다")

#2
from time import sleep                      # 함수까지 진입하여 실행
sleep(5)                                    # 함수이름으로 실행
print("5초가 지났습니다")

#3
import time as t
t.sleep(5)
print("5초가 지났습니다")

```

• 모듈

```

1. 함수나 클래스가 정의
#1
def add(a,b):
    return a+b
def minus(a,b):
    return a-b

#2
import mod1
ret=mod1.add(5,7)
ret2=mod1.minus(7,5)
print(ret, " ", ret2)

2. 패키지 경로 지정
#1
def add(a,b):
    return a+b
def minus(a,b):
    return a-b

#2
import pack.fire
x=pack.fire.add(5,7)
y=pack.fire.minus(9,8)
print(x, " ", y)

--> #2 간소화-1
from pack import fire
x=fire.add(5,7)
y=fire.minus(9,8)
print(x, " ", y)

--> #2 간소화-2
from pack.fire import add,minus
x=add(5,7)
y=minus(9,8)
print(x, " ", y)

```

• 상속

```

class Up:
    def add(self,a,b):
        return a+b

class down(Up):
    def sub(self,a,b):
        # Up 부모 클래스 down 클래스로 상속
        # 상속과 동시에 add 함수를 추가로 가지게 된다

```

```

        return a-b
ins=down()
print(ins.add(1,2))
print(ins.sub(1,2))

```

• 예외처리

```

#1
try:
    print("안녕하세요")
    print(a)          # 에러 발생 구간이지만 try ~ except 구간안에 넣음으로써 예외처리
except:
    print("예외 발생")    # 에러 발생 구간이 없을 시 실행하지 않음
print("계속 갑니다")

* try ~ except 를 사용하지 않으면 에러구간 발생 시 에러메세지 출력 후 종료함
  try ~ except 를 사용하면 에러구간이 발생하여도 계속 진행한다

#2
try:
    print("안녕하세요")
    #print(a)
except:
    print("예외 발생")
finally:
    print("예외가 있든 없든 무조건 실행합니다")

```

• OS 함수안의 특정 디렉토리 경로 안 폴더 리스트 출력

```

import os                                # os(운영체제) 에서 제공하는 명령어 사용
os.chdir("c:/")                          # c: 경로로 이동
dirlist = os.listdir()                  # c: 하위에 존재하는 파일, 폴더 전체를 리스트 형태로 출력
arr = []
for list in dirlist:
    isdir = os.path.isdir(list)         # is : 조건식 , isdir이 디렉토리인지 판단
    if isdir == True:
        arr.append(list)

print(arr)

1. OS 함수안의 특정 디렉토리 경로 안 폴더, 파일 리스트 형태 출력
import os
os.chdir("c:/")
dirlist = os.listdir()
dcol = []
fcol = []
for list in dirlist:
    isdir = os.path.isdir(list)
    isfiles=os.path.isfile(list)
    if isdir == True:
        dcol.append(list)
    if isfiles == True:
        fcol.append(list)

print(dcol)
print(fcol)

1-2. OS 함수안의 특정 디렉토리 경로 검색 후 없을 경우 신규로 생성
import os
os.chdir("c:/Python")
dirlist = os.listdir()
dcol = []
fcol = []
for list in dirlist:
    isdir = os.path.isdir(list)
    isfiles=os.path.isfile(list)
    if isdir == True:
        dcol.append(list)
    if isfiles == True:
        fcol.append(list)

flag=0
for i in dcol:
    if(i=="myhome"):
        flag=1

```

```

if flag==0:
    os.mkdir("myhome")

1-3. 디렉토리 생성 및 파일 존재유무 판별
import os
os.chdir("c:/")
dirlist = os.listdir()
arr = []
for list in dirlist:
    isd = os.path.isdir(list)          # 리스트의 값이 디렉토리인지 판정
    if isd == True:
        arr.append(list)              # 참일 경우 디렉토리 추가
sw=0
for x in arr:
    if x == "exdata"
        sw=1
if sw == 1:
    os.chdir("c:/exdata")              # exdata 디렉토리가 존재할 경우 해당 디렉토리로 이동
else
    os.mkdir("c:/exdata")              # exdata 디렉토리 생성 후 해당 경로로 이동
    os.chdir("c:/exdata")

filelist = os.listdir()
arr=[]
for list in filelist:
    isfile = os.path.isfile(list)
    if isfile == True :
        arr.append(list)
sw = 0
for x in arr:
    if x == "input.xlsx":
        sw = 1
if sw == 1:
    print("c:/exdata 폴더안에 input.xlsx 파일이 존재합니다")
else:
    print("파일이 존재하지 않습니다")

```

• 레이블

```

from tkinter import *

win=Tk()
win.title("타이틀 부분입니다")      # 윈도우 창 상단 타이틀 수정
win.geometry("300x100+200+100")    # 최소 생성시 프레임 위치
win.resizable(0,0)                 # 가로와 세로 크기 재조정
--> (1,1) 로 설정시 변경 가능

subject=Label(win,font=15,height=2,width=10,fg='blue',text="레이블")
subject.config(bg="yellow",anchor="sw") # 초기 레이블 생성시 설정하지 못한 내용 추가 반영
# sw : 텍스트 위치
subject.pack(side=BOTTOM)          # .pack : 서브젝트 붙이는 작업
win.mainloop()                     # tk 클래스의 마무리는 반드시 mainloop 로 끝나야한다

```

• 엔트리

```

from tkinter import *

win=Tk()
win.title("Entry 예제")
win.geometry("300x200+200+200")
win.resizable(0,0)

tf=Entry(win,font=10,width=7,bg="yellow",fg="blue")
tf.config()
tf.pack()                          # 부모 컨테이너의 x축을 다 채운다. 보통 프레임에서 사용
win.mainloop()

1. 엔트리 예제
1-1. 이름표
from tkinter import *

win=Tk()
win.title("이름표")
win.geometry("300x200+200+200")

```

```

win.resizable(0,0)

subject=Label(win,font=10,fg='black',text="나의 이름은 ?")
tf=Entry(win,font=10,width=7,fg="blue")
subject.pack()
tf.pack()

win.mainloop()

# 자료입력
from tkinter import *
from tkinter import messagebox
win = Tk()
win.title("== 자료 입력 ==")
win.geometry("500x500+200+200")
win.resizable(1,0)

def call(event):
    # 이벤트변수로부터 받은 위젯 호출
    what_button=str(event.widget)      # if event.widget==bt 이런식으로 위젯구분
    what_contents=str(tf.get())        # Entry 에 입력한 값을 가져옴
    messagebox.showinfo("타이틀입니다",what_button+what_contents)
    # 모듈로 import 했기때문에 모듈이름을 반드시 적어준다
    # showinfo의 경우 '확인'버튼을 눌러야 다음으로 진행

    tf.delete(0,END)                  # Entry에 입력했던 값을 삭제
    tf.focus_set()                    # Entry에 내용을 지우고 커서 위치하도록

tf=Entry(win)
bt=Button(win,text="누르세요")
bt.bind("<ButtonRelease-1>",call)
# ButtonRelease-1 : 왼쪽버튼 클릭 순간 / ButtonRelease-2 : 마우스 휠 작동시
# ButtonRelease-3 : 마우스 오른쪽 버튼 클릭시
# bind : 프로그램 종료시 까지 이벤트 발생 감시

tf.pack()
bt.pack()

win.mainloop()

2. 확인 / 취소 버튼에 따른 이벤트
from tkinter import *
from tkinter import messagebox
win = Tk()
win.title("== 자료 입력 ==")
win.geometry("500x500+200+200")
win.resizable(1,0)

def call(event):
    if event.widget==bt:
        what_button = str(event.widget)
        what_contents = str(tf.get())
        messagebox.showinfo("타이틀입니다", what_contents)

    elif event.widget==bt2:
        tf.delete(0,END)
        tf.focus_set()

tf=Entry(win)
bt=Button(win,text="확인")
bt.bind("<ButtonRelease-1>",call)

bt2=Button(win,text="취소")
bt2.bind("<ButtonRelease-1>",call)
tf.pack()
bt.pack()
bt2.pack()

win.mainloop()

# global
x=7
def call():
    global x
    x=5
    print(x)

call()
print()

```



```

3. 버튼 누를 시 Entry 값 이동
from tkinter import *
from tkinter import messagebox
win = Tk()
win.title("== 자료 입력 ==")
win.geometry("500x500+200+200")
win.resizable(1,0)

def call(event):
    x=tf.get()
    tf.delete(0,END)
    tf2.insert(0,x)

tf=Entry(win)
tf2=Entry(win)
bt=Button(win, text="확인")
bt.bind("<ButtonRelease-1>", call)

tf.pack()
tf2.pack()
bt.pack()

win.mainloop()

4. 간단한 덧셈 연산
from tkinter import *
from tkinter import messagebox
win = Tk()
win.title("== 자료 입력 ==")
win.geometry("500x500+200+200")
win.resizable(1,0)

def call(event):
    x=int(tf.get())
    y=int(tf2.get())
    z=x+y
    tf3.insert(0, str(z))

tf=Entry(win)
lb1=Label(win, font=10, fg='black', text="+")

tf2=Entry(win)
lb12=Label(win, font=10, fg='black', text="=")

tf3=Entry(win)
bt=Button(win, text="확인")
bt.bind("<ButtonRelease-1>", call)

tf.pack()
lb1.pack()
tf2.pack()
lb12.pack()
tf3.pack()
bt.pack()

win.mainloop()

# 버튼
from tkinter import *

master = Tk()
master.geometry("200x100+100+100")
def call():
    print("click")

b= Button(master, text="OK", command=call, width=10)
# 인스턴스 생성과 동시에 command 함수를 이용하여 바로 호출
# 오직 버튼 클릭에 대한 이벤트만 정의 (버튼이 여러개일 경우 각각 함수를 별도로 생성해야함)
b.pack()

5. 버튼 이벤트
from tkinter import *
from tkinter import messagebox

master = Tk()
master.geometry("200x100+100+100")
def call():
    messagebox.showinfo("", "안녕")

```

```

b= Button(master,text="OK", command=call,width=10)
b.pack()

mainloop()

5-1. 버튼 이벤트(바인딩 이용)

from tkinter import *
from tkinter import messagebox

master = Tk()
master.geometry("200x100+100+100")
def call(event):
    messagebox.showinfo("", "안녕")

b= Button(master,text="OK")
b.bind("<ButtonRelease-1>",call)
b.pack()

mainloop()

# 프레임
from tkinter import *

win=Tk()
win.title("=== 자료입력 ===")
win.geometry("500x500+200+200")          # 가로x세로+x축위치+y축위치
win.resizable(1,0)

inframe=Frame(win,height=10,width=20,bg="yellow")    # TK Win, 컨테이너 외 모든 항목은 위젯
# 프레임의 경우 생성할 때 반드시 옵션을 지정해줘야함

inframe.pack()
#inframe.pack(fill="both") 혹은 inframe.pack(fill=X)
label1=Label(inframe,bg="blue", text="frist")
label2=Label(inframe,bg="blue", text="second")
label3=Label(inframe,bg="blue", text="third")
label1.grid(row=0,column=0,padx=9)
# grid : 배치, row : 행 , column : 열, padx(패딩) : 위젯간 사이 간격
label2.grid(row=0,column=1,padx=18)
label3.grid(row=1,column=1,padx=30)

win.mainloop()

6. #옵션메뉴(콤보상자)
from tkinter import *

def call():
    sec=values.get()
    print(sec,"을 선택하셨습니다")
win=Tk()
values=StringVar()          # 문자열 형태의 객체(리스트형태)
                             # 컴포넌트(위젯)에 쓰이는 값은 IntVar, StringVar 등의 객체여야 한다.
                             # 선택한 값을 가져온다

values.set("Menu")          # 디폴트 값 생각 가능
op=OptionMenu(win,values,"one", "two", "three")
op.pack()
# 값을 가져올때 get 사용
btn=Button(win,text="확인",command=call)
btn.pack()

mainloop()

6-1. 콤보박스 (메뉴선택)
from tkinter import *
def call():
    sec=values.get()
    sec2=tf.get()
    print(sec,sec2,"그릇 주문하셨습니다")

win=Tk()
values=StringVar()
values.set("메뉴")
op=OptionMenu(win,values,"짜장면", "짬뽕", "울면")
op.grid(row=0,column=0)

```

```

tf=Entry(win)
tf.grid(row=0,column=1)
btn=Button(win,text="확인",command=call)
btn.grid(row=0,column=2)

mainloop()

7. 체크버튼
from tkinter import *

def call(event):
    x=event.widget
    if x==btn:
        print("확인 버튼")
    elif x==btn2:
        print("취소 버튼")

    if str1.get()==1:
        print("사과 선택")
    if str2.get()==2:
        print("바나나 선택")
    if str3.get()==1:
        print("포도 선택")

win=Tk()
win.title("=== 자료 입력 ===")
win.geometry("500x500+200+200")
win.resizable(1,0)
str1=IntVar()
str2=IntVar()
str3=IntVar()

check1=Checkbutton(win,text="apple",variable=str1)
check2=Checkbutton(win,text="banana",variable=str2)
check3=Checkbutton(win,text="grape",variable=str3)
check1.pack()
check2.pack()
check3.pack()
btn=Button(win,text="확인")
btn.bind("<ButtonRelease-1>",call)
btn.pack()

btn2=Button(win,text="취소")
btn2.bind("<ButtonRelease-1>",call)
btn2.pack()

win.mainloop()

8. 라디오버튼
from tkinter import *

def call(event):
    x=event.widget
    if x==btn:
        print("확인 버튼")
    elif x==btn2:
        print("취소 버튼")

    if var.get()==1:
        print("사과 선택")
    elif var.get()==2:
        print("바나나 선택")
    elif var.get()==3:
        print("포도 선택")

win=Tk()
win.title("=== 자료 입력 ===")
win.geometry("500x500+200+200")
win.resizable(1,0)
var=IntVar()

radio1=Radiobutton(win,text="apple",variable=var,value=1)
radio2=Radiobutton(win,text="banana",variable=var,value=2)
radio3=Radiobutton(win,text="grape",variable=var,value=3)
radio1.pack()
radio2.pack()
radio3.pack()
btn=Button(win,text="확인")
btn.bind("<ButtonRelease-1>",call)

```

```

btn2=Button(win, text="취소")
btn.bind("<ButtonRelease-1>", call)
btn.pack()
btn2.pack()
win.mainloop()

9. 레이아웃 잡기
from tkinter import *
from PIL import ImageTk, Image
win=Tk()

Label(win, text="First", bg="yellow", fg="blue").grid(row=0, sticky="w")
Label(win, text="Second", bg="yellow", fg="blue").grid(row=1, sticky="w")

ent1= Entry(win).grid(row=0, column=1)
ent2= Entry(win).grid(row=1, column=1)

chk=Checkbutton(win, text="go to fishing?")
chk.grid(row=2, column=0, columnspan=2, sticky=W)          # columnspan : 열병합

img = ImageTk.PhotoImage(Image.open("img/cart.png"))
lbl= Label(win, image=img, bg="yellow")
lbl.grid(row=0, column=2, columnspan=2, rowspan=2, sticky=W+E+N+S, padx=3, pady=3)
# rowspan : 행병합 , sticky=W+E+N+S : 중앙배치
# sticky의 경우 소문자로 사용할때는 "" 를 사용해야함
btn1=Button(win, text="확인")
btn2=Button(win, text="취소")
btn1.grid(row=2, column=2)
btn2.grid(row=2, column=3)
win.mainloop()

10. 이미지 삽입
10-1. 토트로 이미지 삽입
from tkinter import *
from PIL import ImageTk, Image
win=Tk()

Label(win, text="이웃집 토트로").grid(row=0)
img = ImageTk.PhotoImage(Image.open("img/a1.jpg"))
lbl= Label(win, image=img)
lbl.grid(row=1)
win.mainloop()

10-2. 그림선택하기
from tkinter import *
from PIL import ImageTk, Image

def call(e):
    if e.widget==btn1:
        img=ImageTk.PhotoImage(Image.open("img/a2.jpg"))
        imglbl.configure(image=img)
    # configure ~ .image : 이미지 교환시 사용해야하는 정의된 형식
        imglbl.image=img
    elif e.widget==btn2:
        img=ImageTk.PhotoImage(Image.open("img/na.jpg"))
        imglbl.configure(image=img)
        imglbl.image=img

root = Tk()
root.title("== 그림 선택하기 예제 ==")
root.geometry("600x350+200+200")
root.resizable(1,1)
fr=Frame(root)
img=ImageTk.PhotoImage(Image.open("img/a2.jpg"))
imglbl=Label(fr, image=img)
imglbl.pack(side=TOP, fill="both")
fr.pack(side=LEFT)          # side="top" or side=TOP // fill=X or fill ="x"

fr2=Frame(root, background="blue")
btn1=Button(fr2, text="토트로", width=10)
btn2=Button(fr2, text="나루토", width=10)
btn1.grid(row=0, column=0)
btn1.bind("<ButtonRelease>", call)
btn2.grid(row=0, column=1)
btn2.bind("<ButtonRelease>", call)
fr2.pack(side=RIGHT)

root.mainloop()

```

11. 엑셀시트 데이터값 호출

```
#1
from openpyxl import load_workbook

wb=load_workbook('jumsu.xlsx')
ws=wb['Sheet1']
print(ws)
#ws=wb.active
#print(ws)
print(ws['a4'].value)
print(ws['b3'].value)

arr=[]

for i in ws.rows:          # ws.rows : 엑셀 첫행에 해당(무조건 첫행부터 읽는다)
    번호 = i[0].value
    이름 = i[2].value
    tuple_data=(번호,이름)
    arr.append(tuple_data)
print(arr)

#2
from openpyxl import load_workbook

wb=load_workbook('jumsu.xlsx')

ws=wb['second']
arr=[]
for row in ws.iter_rows(min_row=2, max_row=4, max_col=3):
    a=[]
    for cell in row:
        a.append(cell.value)
    arr.append(a)
print(arr)

#3
from openpyxl import Workbook,load_workbook

wb=load_workbook("jumsu.xlsx")          # 해당워크시트를 읽기형태로 연다
ws=wb["second"]
for cell in ws["B"]:
    print(cell.value)                  # B : B열
                                        # B열 전체값 출력
print(ws["a1"].value)                 # a1 : A1 Cell 출력
```

11-2. 엑셀시트 데이터값 호출(for ~ else)

```
from openpyxl import Workbook,load_workbook

wb=load_workbook("jumsu2.xlsx")
ws=wb["second"]
for cell in ws["B"]:
    if cell.value is None:              # is None : 값이 없는 상태
        print(cell.row)
        break                          # break문을 만날 경우 근접한 반복문 중지
    print(cell.row,cell.value)
else:
    # for ~ else 의 경우 break 문을 만날 경우 else 문을 실행하지 않는다
    print(cell.row + 1)
```

11-3. 엑셀시트 내용입력(max_row)

```
from openpyxl import load_workbook
from time import localtime, strftime

start=0
wb = load_workbook("input.xlsx")
#ws = wb.active
ws = wb["second"]
ends = ws.max_row
# max_row : 무조건 가장 마지막 데이터가 있는 값을 읽어드린다.
# 사용시 max_row + 1 형태로 사용하면 된다.
if ends == 1:
    if ws["b1"].value is None:
        start = 1
    else:
        start = 2
else:
    start = ends+1
```

```

ws.cell(row=start, column=1, value=int("{}".format("1")))
--> 동일한방법 : ws.cell(row=start, column=1).value=int("{}".format("1"))
# .format 출력형태의 경우 입력값의 형태를 제한받지 않음
# 형변환과 같은 작업을 간소화 할 수 있음

ws.cell(row=start, column=2, value=int("{}".format("apple")))
ws.cell(row=start, column=3, value=int("{}".format("3")))
ws.cell(row=start, column=4, value=strftime("%Y-%m-%d", localtime()))
# "%Y-%m-%d" : 엑셀에서 인식하는 날짜 입력 방식
--> 2019년 5월 19일
print("{} 행에 자료 입력".format(start))
wb.save("input.xlsx")

11-4. 엑셀시트 내용 입력 (for문 이용)
from openpyxl import load_workbook
from time import localtime, strftime

start=0
wb = load_workbook("input.xlsx")
ws = wb["second"]
ends = ws.max_row

if ends == 1:
    if ws["b1"].value is None:
        start = 1
    else:
        start = 2
else:
    start = ends+1

count=int(input("인원수 :"))
for i in range(0,count):
    ws.cell(row=start, column=1).value = int(input("번호 :"))
    ws.cell(row=start, column=2).value = str(input("이름 :"))
    ws.cell(row=start, column=3).value = int(input("점수 :"))
    ws.cell(row=start, column=4).value=strftime("%Y-%m-%d", localtime())
    start=start+1

wb.save(filename="input.xlsx")

11-5. 엑셀시트 내용 입력(총점 계산)
from openpyxl import load_workbook
from time import localtime, strftime

start=0
wb = load_workbook("input.xlsx")
ws = wb["second"]
ends = ws.max_row

if ends == 1:
    if ws["b1"].value is None:
        start = 1
    else:
        start = 2
else:
    start = ends+1

ws.cell(row=start, column=1).value = str(input("이름 :"))
kor=int(input("국어 :"))
ws.cell(row=start, column=2).value = kor
eng=int(input("영어 :"))
ws.cell(row=start, column=3).value = eng
math=int(input("수학: "))
ws.cell(row=start, column=4).value = math
sum=kor+eng+math
ws.cell(row=start, column=5).value = sum

wb.save(filename="input.xlsx")

# 엑셀시트 내용 입력후 지정된 파일로 생성
from openpyxl import Workbook, load_workbook

wb=Workbook()
ws=wb.active
ws.title="black"
ws=wb["black"]          # 'block' 으로 이름 바꾼 시트 활성화

```

```

ws.cell(row=1,column=1).value= "번호"
ws.cell(row=1,column=2).value= "이름"
ws.cell(row=1,column=3).value= "점수"

count = int(input("인원수 : "))
start = 2
for i in range(1,count+1):
    bunho = int(input("번호 입력 : "))
    name = str(input("이름 입력 : "))
    jumsu = int(input("점수 입력 : "))
    ws.cell(row=start, column=1).value = bunho
    ws.cell(row=start, column=2).value = name
    ws.cell(row=start, column=3).value = jumsu
    start = start + 1
wb.save(filename="input_jumsu.xlsx")

# 기존 생성한 파일을 열어 추가 쓰기
from openpyxl import Workbook, load_workbook
# load_workbook : 새로운 파일을 만들거나 수정한 파일을 저장시 사용
import os

wb=load_workbook("input_jumsu.xlsx")
ws=wb["black"]

count=int(input("추가 인원수 :"))
start=1
flag=0
while True:
    if ws.cell(row=start,column=1).value is not None:
        start = start + 1
    else:
        for i in range(1,count + 1):
            bunho=int(input("번호입력 :"))
            name=str(input("이름입력 :"))
            jumsu=int(input("점수입력 :"))
            ws.cell(row=start, column=1).value = bunho
            ws.cell(row=start, column=2).value = name
            ws.cell(row=start, column=3).value = jumsu
            start = start + 1
        else:
            flag = 1
        if flag == 1:
            break
wb.save(filename="input_jumsu.xlsx")

# 행삭제
from openpyxl import Workbook, load_workbook

wb = load_workbook("input_jumsu.xlsx")
ws = wb["black"]
ws.delete_rows(1,2) #첫번째 인수는 시작행, 두번째인수는 삭제할 행 수
ws.insert_rows(2,2) #행 삽입입니다. 인수는 위와 같습니다. #2행부터 시작 2개 행 삽입

wb.save(filename="input_jumsu2.xlsx")

# 엑셀파일의 빈 행 일괄삭제
from openpyxl import Workbook,load_workbook

wb=load_workbook("jumsu.xlsx")
ws=wb["second"]
start=1
end=ws.max_row
print(end)
for i in range(1,end+1): # 삭제된 행을 포함하여 전체 반복을 위해 max_row값까지 읽어들인다
    if ws.cell(row=start,column=1).value is None:
        ws.delete_rows(start,1)
    else:
        start=start+1
wb.save(filename="input3.xlsx")

11-6. 엑셀시트 내용 입력(총점, 평균 자동계산)
from openpyxl import Workbook, load_workbook

wb=Workbook()
ws=wb.active
ws.title="black"
ws=wb["black"]

```

```

count = int(input("인원수 : "))
start = 2
sum=0

for i in range(1,count+1):
    bunho = int(input("번호 입력 : "))
    name = str(input("이름 입력 : "))
    jumsu = int(input("점수 입력 : "))
    ws.cell(row=start, column=1).value = bunho
    ws.cell(row=start, column=2).value = name
    ws.cell(row=start, column=3).value = jumsu
    sum=sum+jumsu
    start = start + 1
ws.cell(row=start, column=3).value = sum
ws.cell(row=start, column=4).value = "{0:.2f}".format(round(sum/count,2))
# round (2,2) : 3째자리에서 반올림되서 2째자리까지 출력
# 소수점 첫째자리에서 반올림하려면 0 입력
# ex) 3   3   3.   3   3
#     -3  -2  -1.  1   2
wb.save(filename="output.xlsx")

# 엑셀 차트 만들기
from openpyxl import Workbook
from openpyxl.chart import BarChart, Series, Reference
from copy import deepcopy          # 차트 복사본 생성, 원본은 그대로 보존

wb=Workbook()
ws=wb.create_sheet()

rows=[
    ('판매일', '짜장면', '짬뽕'),
    ('2019-05-01', 11, 25),
    ('2019-05-02', 15, 22),
    ('2019-05-03', 31, 12)
]
for row in rows:
    ws.append(row)
chart1 = BarChart()
chart1.type = "col"
chart1.style = 10
chart1.title = "홍콩반점"
chart1.y_axis.title='판매그릇수'
chart1.y_axis.title="판매일자"

data= Reference(ws, min_col=2, min_row=1, max_row=7, max_col=3)
cats= Reference(ws, min_col=1, min_row=2, max_row=7)
chart1.add_data(data, titles_from_data=True)    # titles_from_data=True : 범례출력
chart1.set_categories(cats)
chart1.shape=1                                # 적용불가
ws.add_chart(chart1, "A10")

chart2=deepcopy(chart1)
chart2.style = 11
chart2.type = "bar"
chart2.title = "홍콩반점2"

ws.add_chart(chart2, "G10")

chart3 = deepcopy(chart1)
chart3.type = "col"
chart3.style = 12
chart3.grouping = "stacked"
chart3.overlap = 100
chart3.title="홍콩반점3"

ws.add_chart(chart3, "A27")

chart4=deepcopy(chart1)
chart4.type="bar"
chart4.style=13
chart4.grouping="percentStacked"
chart4.overlap=100
chart4.title="홍콩반점4"

ws.add_chart(chart4, "G27")
wb.save("bar.xlsx")

# 2차원 배열(이중 for 문)
big=[]

```



```

for i in range(1,4):          # range(1,4,3) : 1부터 4까지 3씩 증가한다
    small=[]
    for j in range(i,4):      # range(i,i+3) : 4부터 시작해서 +3까지 진행
        small.append(j)
    big.append(small)
print(big)

```

11-7. 엑셀차트 만들기(2차원 배열, 급여현황)

```

from openpyxl import Workbook
from openpyxl.chart import BarChart, Series, Reference
from copy import deepcopy

```

```

wb=Workbook()
ws=wb.create_sheet()

```

```

rows = [['성명', '기본급', '보너스']]
for i in range(1,4):
    row = []
    name = str(input("이름 입력: "))
    row.append(name)
    don1 = int(input("기본급 입력: "))
    row.append(don1)
    don2 = int(input("보너스 입력: "))
    row.append(don2)
    rows.append(row)
print(rows)

```

```

for row in rows:
    ws.append(row)
chart1 = BarChart()
chart1.type = "col"
chart1.style = 10
chart1.title = "급여내역"
chart1.y_axis.title='금액'
chart1.x_axis.title="성명"

```

```

data= Reference(ws, min_col=2, min_row=1, max_row=4, max_col=3)
# 2열에서 3열까지 입력, 1~4행까지 입력
cats= Reference(ws, min_col=1, min_row=2, max_row=4)
chart1.add_data(data, titles_from_data=True)
chart1.set_categories(cats)
chart1.shape=1
ws.add_chart(chart1, "A10")
wb.save("bar1.xlsx")

```

11-8. 엑셀차트 만들기(입력한 수만큼)

```

from openpyxl import Workbook
from openpyxl.chart import BarChart, Series, Reference
from copy import deepcopy

```

```

wb=Workbook()
ws=wb.create_sheet()

```

```

count = int(input("인원수: "))
start = 2

```

```

rows = [['성명', '기본급', '보너스']]
for i in range(1,count+1):
    row = []
    name = str(input("이름 입력: "))
    row.append(name)
    don1 = int(input("기본급 입력: "))
    row.append(don1)
    don2 = int(input("보너스 입력: "))
    row.append(don2)
    rows.append(row)
print(rows)

```

```

for row in rows:
    ws.append(row)
chart1 = BarChart()
chart1.type = "col"
chart1.style = 10
chart1.title = "급여내역"
chart1.y_axis.title='금액'
chart1.x_axis.title="성명"

```

```

data= Reference(ws, min_col=2, min_row=1, max_row=4, max_col=3)
cats= Reference(ws, min_col=1, min_row=2, max_row=4)

```

```

chart1.add_data(data, titles_from_data=True)
chart1.set_categories(cats)
chart1.shape=1
ws.add_chart(chart1, "A10")
wb.save("bar2.xlsx")

11-9. 텍스트 문서를 활용한 엑셀차트 만들기
from openpyxl import Workbook
from openpyxl.chart import BarChart, Series, Reference
from copy import deepcopy

wb=Workbook()
ws=wb.create_sheet()

rows = [['성명', '기본급', '보너스']]
f=open("excel_data.txt","r")
lines=f.readlines()
cnt=lines.__len__()          # __len__ : 한줄씩 분리
for line in lines:
    name=[]
    s=line.split(",")
    name.append(s[0])
    name.append(int(s[1]))
    name.append(int(s[2]))
    rows.append(name)
    print(s)
f.close()
lines.append(rows)

for row in rows:
    ws.append(row)
chart1 = BarChart()
chart1.type = "col"
chart1.style = 10
chart1.title = "급여내역"
chart1.y_axis.title='금액'
chart1.x_axis.title="성명"

data= Reference(ws, min_col=2, min_row=1, max_row=cnt+1, max_col=3)
cats= Reference(ws, min_col=1, min_row=2, max_row=cnt+1)
chart1.add_data(data, titles_from_data=True)
chart1.set_categories(cats)
chart1.shape=1
ws.add_chart(chart1, "A10")
wb.save("bar3.xlsx")

```

```

11-10. 엑셀파일의 값을 텍스트 형태로 저장하기
from openpyxl import load_workbook

wb=load_workbook('pay.xlsx')
ws=wb.active
f=open("excel_test.txt","w")
ends = ws.max_row
array=[]
for i in range(1,ends+1):
    one=ws.cell(row=i, column=1).value
    two=ws.cell(row=i, column=2).value
    three=ws.cell(row=i, column=3).value
    print(one,two,three)
    temp=one+", "+str(two)+", "+str(three)+"\n"
    array.append(temp)
f.writelines(array)
f.close()
print("excel_test 를 생성하였습니다")

```

- **filedialog**

```

from tkinter import filedialog

textfilename = filedialog.askopenfilename(initialdir="/", title="텍스트 파일 선택", filetypes=(("all file", "*.*"), ("text files", "*.txt"), ("all images", "*.jpg")))
print(textfilename)

# initialdir="/" : 파일 오픈 첫 경로

```

```

# askopenfilenames : 파일 다중선택 가능

1. 특정경로에 있는 파일을 선택하여 내용 읽어오기
from tkinter import filedialog

fname = filedialog.askopenfilenames(initialdir="c:/Python2", title="텍스트 파일 선택" ,filetypes=(("all types","*.*"),("text files", "*.txt"),("all images","*.jpg"),))
print(fname)

f=open(fname, "r")

temp=f.readlines()
print(temp)

2. 특정경로에 있는 파일을 다중으로 선택하여 읽어오지만 원하는 특정 파일 내용만 읽어오기
from tkinter import filedialog

fname = filedialog.askopenfilenames(initialdir="c:/Python2", title="텍스트 파일 선택" ,filetypes=(("all types","*.*"),("text files", "*.txt"),("all images","*.jpg"),))
print(fname)

for i in fname:
    print(i)
    if i=="c:/Python2/out1.txt":
# if i=="C:/Python2/out1.txt": 리스트안에 ""에 해당하는 파일이 있으면 True
        f=open(str(i),"r")
        temp=f.read()
        print(temp)

3. 특정 경로의 파일 읽어오기 (split + readline 사용)
from tkinter import filedialog

fname = filedialog.askopenfilename(initialdir="c:/Python2", title="텍스트 파일 선택" ,filetypes=(("all types","*.*"),("text files", "*.txt"),("all images","*.jpg"),))
print(fname)

f=open(fname,"r")
temp=f.readline()
print(temp)
while temp:
    s=temp.split(",")
    print(s)
    temp=f.readline()
f.close()

4. 특정 경로의 파일 읽어오기 (split + readlines 사용)
from tkinter import filedialog

fname = filedialog.askopenfilename(initialdir="c:/Python2", title="텍스트 파일 선택" ,filetypes=(("all types","*.*"),("text files", "*.txt"),("all images","*.jpg"),))
print(fname)

f=open(fname,"r")
temp=f.readlines()
for i in temp:
    s=i.split(",")
    print(s)
f.close()

5. 텍스트파일 엑셀에 쓰기
from tkinter import filedialog
from openpyxl import Workbook

fname = filedialog.askopenfilename(initialdir="c:/Python2", title="텍스트 파일 선택" ,filetypes=(("all types","*.*"),("text files", "*.txt"),("all images","*.jpg"),))
print(fname)
wb=Workbook()
ws=wb.active
start=1
f=open(fname,"r")
temp=f.readlines()
for i in temp:
    s = i.split(",")
    ws.cell(row=start, column=1).value = int(s[0])
    print(s[0])
    ws.cell(row=start, column=2).value = str(s[1])
    print(s[1])
    ws.cell(row=start, column=3).value = int(s[2])
    print(s[2])

```

```

        start=start+1

f.close()
wb.save("C:/Python2/result.xlsx")

6. 텍스트에 있는 값 중 최대값을 엑셀에 쓰기
from tkinter import filedialog
from openpyxl import Workbook

fname = filedialog.askopenfilename(initialdir="c:/Python2", title="텍스트 파일 선택", filetypes=(("all types", "*..*"), ("text files", "*.txt"), ("all images", "*.jpg"),))
print(fname)
wb=Workbook()
ws=wb.active
start=1
max=0
f=open(fname, "r")
temp=f.readlines()
for i in temp:
    s = i.split(",")
    if max<int(s[2]):
        max=int(s[2])
        a=int(s[0])
        b=str(s[1])
        c=int(s[2])
ws.cell(row=1, column=1).value=a
ws.cell(row=1, column=2).value=b
ws.cell(row=1, column=3).value=c
f.close()
wb.save("C:/Python2/result2.xlsx")

7. 엑셀파일 내용 읽어와서 출력
from tkinter import filedialog
from openpyxl import load_workbook

fname = filedialog.askopenfilename(initialdir="c:/Python2", title="엑셀 파일 선택", filetypes=(("all types", "*..*"), ("text files", "*.txt"), ("all images", "*.jpg"),))
wb=load_workbook(fname)
ws=wb.active
print(ws.cell(row=1, column=1).value)
print(ws.cell(row=1, column=2).value)
print(ws.cell(row=1, column=3).value)

8. 버튼 일력시 엑셀파일에 내용 저장
from tkinter import *
from openpyxl import Workbook, load_workbook
import os

start=1
wb = Workbook()
ws = wb.active

def call(event):
    global start
    lists=os.listdir()
    if "result.xlsx" in lists:
        wb = load_workbook("result.xlsx")
        ws = wb.active
    else:
        wb=Workbook()
        ws=wb.active

    ws.cell(row=start, column=1).value = int(ent1.get())
    ws.cell(row=start, column=2).value = str(ent2.get())
    ws.cell(row=start, column=3).value = int(ent3.get())
    start=start+1
    ent1.delete(0, END)
    ent2.delete(0, END)
    ent3.delete(0, END)
    ent1.focus_set()
    wb.save("result.xlsx")

win=Tk()
win.title("타이틀 부분입니다")
win.geometry("300x100+200+100")
win.resizable(0,0)
lb1=Label(win, text="번호")
lb2=Label(win, text="이름")
lb3=Label(win, text="점수")

```

```

lb1.grid(row=0, column=0)
lb2.grid(row=0, column=1)
lb3.grid(row=0, column=2)
ent1= Entry(win,width=7)
ent2= Entry(win,width=7)
ent3= Entry(win,width=7)
ent3.bind("<Return>", call)
ent1.grid(row=1, column=0)
ent2.grid(row=1, column=1)
ent3.grid(row=1, column=2)
btn=Button(win,text="Go")
btn.grid(row=0, column=3, rowspan=2)
btn.bind("<ButtonRelease-1>", call)

win.mainloop()

9. 버튼 일러시 엑셀파일에 내용 저장 (함수호출 이용)
from tkinter import *
from openpyxl import Workbook, load_workbook
import os

start=1
wb = Workbook()
ws = wb.active

def dong():
    ent1.delete(0, END)
    ent2.delete(0, END)
    ent3.delete(0, END)
    ent1.focus_set()

def call(event):
    global start
    lists=os.listdir()
    if "result.xlsx" in lists:
        wb = load_workbook("result.xlsx")
        ws = wb.active
    else:
        wb=Workbook()
        ws=wb.active

    ws.cell(row=start, column=1).value = int(ent1.get())
    ws.cell(row=start, column=2).value = str(ent2.get())
    ws.cell(row=start, column=3).value = int(ent3.get())
    start=start+1
    wb.save("result.xlsx")
    dong()

win=Tk()
win.title("타이틀 부분입니다")
win.geometry("300x100+200+100")
win.resizable(0,0)
lb1=Label(win,text="번호")
lb2=Label(win,text="이름")
lb3=Label(win,text="점수")
lb1.grid(row=0, column=0)
lb2.grid(row=0, column=1)
lb3.grid(row=0, column=2)
ent1= Entry(win,width=7)
ent2= Entry(win,width=7)
ent3= Entry(win,width=7)
ent3.bind("<Return>", call)
ent1.grid(row=1, column=0)
ent2.grid(row=1, column=1)
ent3.grid(row=1, column=2)
btn=Button(win,text="Go")
btn.grid(row=0, column=3, rowspan=2)
btn.bind("<ButtonRelease-1>", call)

win.mainloop()

```

• 옵션메뉴

```

from tkinter import *

def ok():
    print("value is", var.get())

```

```

master.quit()

def drop(*args):          # 번지값을 받아온다
    print("you choose!")
    print(var.get)

master=Tk()
var=StringVar(master)
option = OptionMenu(master, var, "apple", "lemon", "banana")
var.set("menu")
var.trace("w", drop)
option.pack()
button=Button(master, text="OK", command=ok) # command=ok : 클릭시 ok 함수 호출
button.pack()

mainloop()

1. 옵션메뉴에서 선택시 텍스트에 저장되어 있는 값과 비교하여 일치하는 금액정보 호출하기 (readlines 사용)
from tkinter import *

def ok():
    print("value is", var.get())
    master.quit()

def drop(*args):
    ent1.delete(0, END)
    f = open("pan.txt", "r")
    temp = f.readlines()
    for i in temp:
        s = i.split(",")
        if var.get() == s[0]:
            ent1.insert(0, s[1])
    f.close()

master = Tk()
master.geometry("200x100+100+100")
master.resizable(0, 0)
var = StringVar(master)
option = OptionMenu(master, var, "apple", "lemon", "banana")
var.set("menu")
var.trace("w", drop)
option.grid(row=0, column=0)

ent1 = Entry(master, width=7)
ent1.grid(row=0, column=1)

button = Button(master, text="OK", command=ok)
button.grid(row=0, column=2)

mainloop()

2. 옵션메뉴에서 선택 후 버튼 선택 시 엑셀에 내용 입력
from tkinter import *
from openpyxl import Workbook, load_workbook
import os
start=1
def ok():
    global start
    lists = os.listdir()
    if "result.xlsx" in lists:
        wb = load_workbook("result.xlsx")
        ws = wb.active
    else:
        wb = Workbook
        ws = wb.active
    ws.cell(row=start, column=1).value = var.get()
    ws.cell(row=start, column=2).value = ent1.get()
    start = start + 1
    wb.save("result.xlsx")

def drop(*args):
    ent1.delete(0, END)
    f = open("pan.txt", "r")
    temp = f.readlines()
    for i in temp:
        s = i.split(",")
        if var.get() == s[0]:
            ent1.insert(0, s[1])
    f.close()

```

```

master = Tk()
master.geometry("200x100+100+100")
master.resizable(0, 0)
var = StringVar(master)
option = OptionMenu(master, var, "apple", "lemon", "banana")
var.set("menu")
var.trace("w", drop)
option.grid(row=0, column=0)

ent1 = Entry(master, width=7)
ent1.grid(row=0, column=1)

button = Button(master, text="OK", command=ok)
button.grid(row=0, column=2)

mainloop()

```

• 실행파일 환경 만들기

1. `get-pip.py` 다운로드
2. cmd창에서 `get-pip.py` 실행 (python `get-pip.py`)
3. 설치확인 : pip
4. 버전 업데이트 : `python -m pip install --upgrade pip`
5. 패키지 업데이트 : `pip install pyinstaller`
6. 설치 확인 : `pyinstaller --help`
7. 실행파일 생성 : `pyinstaller test.py --onefile -w`
8. 실행파일 아이콘 설정 : `pyinstaller test.py --onefile -w --icon=img/palmetree.ico calc.py`

※ 현재경로의 경우 ./로 표시

• 실행파일 만들기(GUI Tool - nsis 활용)

1. `pip install auto-py-to-exe`
2. `auto-py-to-exe`
3. nsis 다운로드
4. icon , 파일생성경로 지정 후 Convert 작업 진행
5. 생성된 폴더 Zip 파일로 변환
6. installer based on. zip file 선택 후 미리 변환한 zip 파일 선택 후 경로 지정후 generate 버튼 클릭

• 데이터베이스

DBMS : 데이터베이스를 사용하기 위한 S/W

• SQL

1. DDL : 데이터정의어
2. DML : 데이터조작어 (레코드 삽입,삭제,갱신)
3. DCL : 데이터제어어 (DBA가 주로 사용)

```

show databases;           # 생성된 데이터베이스 리스트 확인
create database chic;      # 새로운 데이터베이스 생성
use chic                   # 데이터베이스 사용
create table student(name varchar(10),kor int,gen char(1)); # 스키마 생성

```

```

* varchar(10) : 처음 지정시 변동사항이 없을 때 사용

insert into student(name,kor,gen)values('apple',85,'m');
select * from student;          # 생성된 테이블 조회
desc student;
alter table student - add address char(10);    # 추가
- change address juso char(20);              # 변경
- drop juso;                                  # 삭제

select * from student where gen = 'm';         # gen 에서 항목이 m인것 검색
select name,gen from student where gen = 'm';  # 이름과 gen 에서 항목이 m인것 검색
select * from student where kor>=80 and gen='m'; # kor 80 이상과 gen에서 항목이 m인것 검색
select * from student where name like 'a%';    # a로 시작하는 항목 검색
select gen,sum(kor),avg(kor) as ave from student where name like 'a%' group by gen order by gen desc;
# gen 그룹에서 a라는 문자가 포함된 항목에 대하여 kor 합계, kor 평균을 구하고 내림차순 정렬
# 오름차순의 경우 생략

update student set kor=kor+5 where gen='m';
update student set name='avo' where name='avocado';

delete from student where gen='f'

```

1. 성적표 데이터베이스 연계 #1

```

from tkinter import *
import pymysql

class inputform:
    def __init__(self,form):
        self.frame = Frame(form)
        self.frame.pack(fill=X)          # fill=X : x축을 가득 채운다
        self.label1 = Label(self.frame, text="번호")
        self.label1.grid(row=0, column=0, padx=5)
        self.label2 = Label(self.frame, text="이름")
        self.label2.grid(row=0, column=1, padx=5)
        self.label3 = Label(self.frame, text="국어")
        self.label3.grid(row=0, column=2, padx=5)
        self.label4 = Label(self.frame, text="영어")
        self.label4.grid(row=0, column=3, padx=5)
        self.label5 = Label(self.frame, text="수학")
        self.label5.grid(row=0, column=4, padx=5)

        self.input1=Entry(self.frame,width=10)
        self.input1.grid(row=1,column=0,padx=5)
        self.input2 = Entry(self.frame, width=10)
        self.input2.grid(row=1, column=1, padx=5)
        self.input3 = Entry(self.frame, width=10)
        self.input3.grid(row=1, column=2, padx=5)
        self.input4 = Entry(self.frame, width=10)
        self.input4.grid(row=1, column=3, padx=5)
        self.input5 = Entry(self.frame, width=10)
        self.input5.grid(row=1, column=4, padx=5)

    def exexute(self):
        con=pymysql.connect(host='localhost', user='root', passwd='1234',db='myclass')
        cur=con.cursor()                  # con.cursor (접속개체)

        sql="insert into student(num,name,kor,eng,math,total,avg)values(%s,%s,%s,%s,%s,%s,%s)"
        total=int(self.input3.get())+int(self.input4.get())+int(self.input5.get())
        avg=float(total)/3
        print("%d \t %f"%(total,avg))

        cur.execute(sql,(int(self.input1.get()),self.input2.get(),int(self.input3.get()),int(self.input4.get()),int(self
        f.input5.get()),total,avg))
        con.commit()
        con.close()

form=Tk()
form.title("1학년 1반 학생 성적부")
form.geometry("800x400+200+200")
form.resizable(1,1)
lbsubject=Label(form, text="학생 성적 정리표",font=10)
lbsubject.pack()
x=inputform(form)
button=Button(form, text="확인", width=20, command=x.exexute)
# x.exexute : 복사본 execute 함수 호출
button.pack()
form.mainloop()

```



```

1-1. 성적표 데이터베이스 연계 #2 (삭제버튼 클릭 시 테이블 내 레코드 삭제)
from tkinter import *
import pymysql

form=Tk()
form.title("1학년 1반 학생 성적부")
form.geometry("800x400+200+200")
form.resizable(1,1)
lbsubject=Label(form, text="학생 성적 정리표",font=10)
lbsubject.pack()

frame = Frame(form)
frame.pack(fill=X)
label1 = Label(frame, text="번호")
label1.grid(row=0, column=0, padx=5)
label2 = Label(frame, text="이름")
label2.grid(row=0, column=1, padx=5)
label3 = Label(frame, text="국어")
label3.grid(row=0, column=2, padx=5)
label4 = Label(frame, text="영어")
label4.grid(row=0, column=3, padx=5)
label5 = Label(frame, text="수학")
label5.grid(row=0, column=4, padx=5)
input1=Entry(frame,width=10)
input1.grid(row=1,column=0,padx=5)
input2 = Entry(frame, width=10)
input2.grid(row=1, column=1, padx=5)
input3 = Entry(frame, width=10)
input3.grid(row=1, column=2, padx=5)
input4 = Entry(frame, width=10)
input4.grid(row=1, column=3, padx=5)
input5 = Entry(frame, width=10)
input5.grid(row=1, column=4, padx=5)

def cute1():
    con=pymysql.connect(host='localhost', user='root', passwd='1234',db='myclass')
    cur=con.cursor()
    sql="insert into student(num,name,kor,eng,math,total,avg)values(%s,%s,%s,%s,%s,%s,%s)"
    total=int(input3.get())+int(input4.get())+int(input5.get())
    avg=float(total)/3
    print("%d \t %f"%(total,avg))
    cur.execute(sql, (int(input1.get()),input2.get(),int(input3.get()),int(input4.get()),int(input5.get()),total,avg))
    con.commit()
    con.close()

def cute2():
    con=pymysql.connect(host='localhost', user='root', passwd='1234',db='myclass')
    cur=con.cursor()
    sql="delete from student"
    cur.execute(sql)
    con.commit()
    con.close()

button1 = Button(form, text="확인", width=20, command=cute1)
button1.pack()

button2 = Button(form, text="삭제", width=20, command=cute2)
button2.pack()
form.mainloop()

1-2. 성적표 데이터베이스 연계 #3 (특정 레코드 지정하여 삭제하기)
from tkinter import *
import pymysql

form=Tk()
form.title("1학년 1반 학생 성적부")
form.geometry("800x400+200+200")
form.resizable(1,1)
lbsubject=Label(form, text="학생 성적 정리표",font=10)
lbsubject.pack()

frame = Frame(form)
frame.pack(fill=X)
label1 = Label(frame, text="번호")
label1.grid(row=0, column=0, padx=5)
label2 = Label(frame, text="이름")
label2.grid(row=0, column=1, padx=5)
label3 = Label(frame, text="국어")

```

```

label3.grid(row=0, column=2, padx=5)
label4 = Label(frame, text="영어")
label4.grid(row=0, column=3, padx=5)
label5 = Label(frame, text="수학")
label5.grid(row=0, column=4, padx=5)
input1=Entry(frame,width=10)
input1.grid(row=1,column=0,padx=5)
input2 = Entry(frame, width=10)
input2.grid(row=1, column=1, padx=5)
input3 = Entry(frame, width=10)
input3.grid(row=1, column=2, padx=5)
input4 = Entry(frame, width=10)
input4.grid(row=1, column=3, padx=5)
input5 = Entry(frame, width=10)
input5.grid(row=1, column=4, padx=5)

def cute1():
    con=pymysql.connect(host='localhost', user='root', passwd='1234',db='myclass')
    cur=con.cursor()
    sql="insert into student(num,name,kor,eng,math,total,avg)values(%s,%s,%s,%s,%s,%s,%s)"
    total=int(input3.get())+int(input4.get())+int(input5.get())
    avg=float(total)/3
    print("%d \t %f"%(total,avg))
    cur.execute(sql, (int(input1.get()),input2.get(),int(input3.get()),int(input4.get()),int(input5.get()),total,avg))
    con.commit()
    con.close()

def cute2():
    con=pymysql.connect(host='localhost', user='root', passwd='1234',db='myclass')
    cur=con.cursor()
    sql="delete from student where name=%s"
    cur.execute(sql,input6.get())
    con.commit()
    con.close()

button1 = Button(form, text="확인", width=20, command=cute1)
button1.pack()

input6 = Entry(form, width=10)
input6.pack()

button2 = Button(form, text="삭제", width=20, command=cute2)
button2.pack()
form.mainloop()

1-3. 성적표 데이터베이스 연계 #4 (실행시 테이블 조회하기)
from tkinter import *
from tkinter.ttk import Treeview
import pymysql

form = Tk()
form.title("1학년 1반 학생 성적부")
form.geometry("800x400+200+200")
form.resizable(1, 1)

con = pymysql.connect(host='localhost', user='root', passwd='1234', db='myclass')
cur = con.cursor()
tree = Treeview(form, columns=("a", "b"), selectmode="extended")
tree.column('a', width=50, anchor=CENTER)
tree.column('b', width=100, anchor=CENTER)
tree.heading('a', text="num")
tree.heading("b", text="kor")
tree['show'] = 'headings'
sql = "select * from test"
cur.execute(sql)
many = cur.fetchall()

for row in many:
    tree.insert("", END, values=(row[0], row[1],))
con.commit()
con.close()
tree.pack()
form.mainloop()

1-4. 성적표 데이터베이스 연계 #5 (특정이름 필터하여 테이블 조회하기)
from tkinter import *
from tkinter.ttk import Treeview
import pymysql

```

```

def call():
    global tree, fr
    tree.destroy()
    con = pymysql.connect(host='localhost', user='root', passwd='1234', db='myclass')
    cur = con.cursor()
    tree = Treeview(fr, columns=("a", "b"), selectmode="extended")
    tree.column('a', width=50, anchor=CENTER)
    tree.column('b', width=100, anchor=CENTER)
    tree.heading("a", text="num")
    tree.heading("b", text="kor")
    tree['show'] = 'headings'
    sql = "select * from test where name=%s"
    cur.execute(sql, input1.get())
    many = cur.fetchall()
    for row in many:
        tree.insert("", END, values=(row[0], row[1],))
    con.commit()
    con.close()
    tree.pack()

form = Tk()
form.title("1학년 1반 학생 성적부")
form.geometry("800x400+200+200")
form.resizable(1, 1)
fr=Frame(form)

con = pymysql.connect(host='localhost', user='root', passwd='1234', db='myclass')
cur = con.cursor()
tree = Treeview(fr, columns=("a", "b"), selectmode="extended")
tree.column('a', width=50, anchor=CENTER)
tree.column('b', width=100, anchor=CENTER)
tree.heading("a", text="num")
tree.heading("b", text="kor")
tree['show'] = 'headings'
sql = "select * from test"
cur.execute(sql)
many = cur.fetchall()

for row in many:
    tree.insert("", END, values=(row[0], row[1],))
con.commit()
con.close()
tree.pack()
fr.pack()

input1 = Entry(form, width=10)
input1.pack()

button = Button(form, text="확인", width=20, command=call)
button.pack()
form.mainloop()

1-5. 성적표 데이터베이스 연계 #6 (메뉴표 테이블 값 입력)
from tkinter import *
from tkinter.ttk import Treeview
import pymysql

def call():
    con = pymysql.connect(host='localhost', user='root', passwd='1234', db='myclass')
    cur = con.cursor()
    sql = "insert into hong(menu,price,count)values(%s,%s,%s)"
    cur.execute(sql, (ent1.get(), int(ent2.get()), int(ent3.get())))
    con.commit()
    con.close()

win=Tk()
win.title("메뉴판")
win.geometry("300x100+200+100")
win.resizable(0,0)
lb1=Label(win, text="메뉴")
lb2=Label(win, text="가격")
lb3=Label(win, text="수량")
lb1.grid(row=0, column=0)
lb2.grid(row=0, column=1)
lb3.grid(row=0, column=2)
ent1= Entry(win,width=7)
ent2= Entry(win,width=7)
ent3= Entry(win,width=7)
ent1.grid(row=1, column=0)

```

```
ent2.grid(row=1, column=1)
ent3.grid(row=1, column=2)
btn=Button(win,text="Go",command=call)
btn.grid(row=2, column=1)
win.mainloop()
```

1-6. 성적표 데이터베이스 연계 #7 (메뉴표 테이블 값 입력 - bind 사용)

```
from tkinter import *
from tkinter.ttk import Treeview
import pymysql

def call():
    con = pymysql.connect(host='localhost', user='root', passwd='1234', db='myclass')
    cur = con.cursor()
    sql = "insert into hong(menu,price,count)values(%s,%s,%s)"
    cur.execute(sql,(ent1.get(), int(ent2.get()), int(ent3.get())))
    con.commit()
    con.close()

def call2(event):
    con = pymysql.connect(host='localhost', user='root', passwd='1234', db='myclass')
    cur = con.cursor()
    sql = "insert into hong(menu,price,count)values(%s,%s,%s)"
    cur.execute(sql, (ent1.get(), int(ent2.get()), int(ent3.get())))
    con.commit()
    con.close()

win=Tk()
win.title("메뉴판")
win.geometry("300x100+200+100")
win.resizable(0,0)
lb1=Label(win,text="메뉴")
lb2=Label(win,text="가격")
lb3=Label(win,text="수량")
lb1.grid(row=0, column=0)
lb2.grid(row=0, column=1)
lb3.grid(row=0, column=2)
ent1= Entry(win,width=7)
ent2= Entry(win,width=7)
ent3= Entry(win,width=7)

ent1.grid(row=1, column=0)
ent2.grid(row=1, column=1)
ent3.grid(row=1, column=2)
ent3.bind("<Return>",call2)
btn=Button(win,text="Go",command=call)
btn.grid(row=2, column=1)
win.mainloop()
```

1-7. 성적표 데이터베이스 연계 #8 (Treeview 값 bind)

```
# 정규식
1) 패턴 객체 생성
2) match, search 명령 사용

import re

temp="apple is 54 count, apple is red"
p=re.compile("apple")
result=p.match(temp)          # 반드시 첫번째 문자열부터 일치해야함
result=p.search(temp)         # 처음부터 일치하지 않아도 검색가능
result=p.findall(temp)        # 원하는 패턴 모두 검색하여 리스트 형태로 출력
result=p.finditer(temp)
for i in result:              # 원하는 패턴 모두 검색하여 match 값으로 리턴
    print(i.group())          # span, start 모두 사용가능

print(result, group(0))
```

```
2. treeview
from tkinter import *
from tkinter.ttk import Treeview
import pymysql

form = Tk()
form.title("1학년 1반 학생 성적부")
```

```

form.geometry("800x400+200+200")
form.resizable(1, 1)
tree = Treeview(form, columns=("a", "b", "c", "d", "e", "f", "g"), selectmode="extended")

lbsubject = Label(form, text="학생 성적 정리표", font=10)
lbsubject.pack()
frame = Frame(form)
frame.pack()
label1 = Label(frame, text="번호")
label1.grid(row=0, column=0, padx=5)
label2 = Label(frame, text="이름")
label2.grid(row=0, column=1, padx=5)
label3 = Label(frame, text="국어")
label3.grid(row=0, column=2, padx=5)
label4 = Label(frame, text="영어")
label4.grid(row=0, column=3, padx=5)
label5 = Label(frame, text="수학")
label5.grid(row=0, column=4, padx=5)

input1 = Entry(frame, width=10)
input1.grid(row=1, column=0, padx=5)
input2 = Entry(frame, width=10)
input2.grid(row=1, column=1, padx=5)
input3 = Entry(frame, width=10)
input3.grid(row=1, column=2, padx=5)
input4 = Entry(frame, width=10)
input4.grid(row=1, column=3, padx=5)
input5 = Entry(frame, width=10)
input5.grid(row=1, column=4, padx=5)

def cute():
    con = pymysql.connect(host='localhost', user='root', passwd='1234', db='myclass')
    cur = con.cursor()

    sql = "insert into student(num,name,kor,eng,math,total,avg)values(%s,%s,%s,%s,%s,%s,%s)"
    total = int(input3.get()) + int(input4.get()) + int(input5.get())
    avg = round(float(total) / 3, 2)
    cur.execute(sql,(int(input1.get()), input2.get(), int(input3.get()), int(input4.get()), int(input5.get()), total, av
g))
    con.commit()
    con.close()
    update()

def update():
    global tree
    tree.destroy()
    # tree.selection()
    con = pymysql.connect(host='localhost', user='root', passwd='1234', db='myclass')
    cur = con.cursor()

    tree = Treeview(form, columns=("a", "b", "c", "d", "e", "f", "g"), selectmode="extended")
    # tree["columns"]="one","two"

    tree.column('a', width=50, anchor=CENTER)
    tree.column('b', width=100, anchor=CENTER)
    tree.column('c', width=50, anchor=CENTER)
    tree.column('d', width=50, anchor=CENTER)
    tree.column('e', width=50, anchor=CENTER)
    tree.column('f', width=50, anchor=CENTER)
    tree.column('g', width=70, anchor=CENTER)

    tree.heading("a", text="num")
    tree.heading("b", text="name")
    tree.heading("c", text="Kor")
    tree.heading("d", text="Eng")
    tree.heading("e", text="Math")
    tree.heading("f", text="Sum")
    tree.heading("g", text="Avg")

    sql = "select * from student"
    cur.execute(sql)
    many = cur.fetchall()
    tree['show'] = 'headings'
    for row in many:
        tree.insert("", END, values=(row[0], row[1], row[2], row[3], row[4], row[5], round(row[6], 2)))
    con.commit()
    con.close()

```

```

tree.pack()

def createtree():
    global tree          # 전역변수 : 메인에서 선언된 함수
    tree.destroy()       # 초기 선언 시 삭제할 데이터가 없기 때문에 별도로 선언 불필요
    con = pymysql.connect(host='localhost', user='root', passwd='1234', db='myclass')
    cur = con.cursor()

    tree = Treeview(form, columns=("a", "b", "c", "d", "e", "f", "g"), selectmode="extended")
    # selectmode="extended" : 다중선택
    # selectmode="none" : 선택불가
    # selectmode="single" : 단일선택
    # tree["columns"]="("one", "two")

    tree.column('a', width=50, anchor=CENTER)
    tree.column('b', width=100, anchor=CENTER)
    tree.column('c', width=50, anchor=CENTER)
    tree.column('d', width=50, anchor=CENTER)
    tree.column('e', width=50, anchor=CENTER)
    tree.column('f', width=50, anchor=CENTER)
    tree.column('g', width=70, anchor=CENTER)

    tree.heading("a", text="num")
    tree.heading("b", text="name")
    tree.heading("c", text="Kor")
    tree.heading("d", text="Eng")
    tree.heading("e", text="Math")
    tree.heading("f", text="Sum")
    tree.heading("g", text="Avg")

    sql = "select * from student"
    cur.execute(sql)
    many = cur.fetchall()          # Select 쿼리 실행 결과 정보를 받아오기 위해 사용
    tree['show'] = 'headings'
    for row in many:
        tree.insert("", END, values=(row[0], row[1], row[2], row[3], row[4], row[5], round(row[6], 2)))
# "", END : 한줄씩 써내려간다
    con.commit()
    con.close()
    tree.pack()

button = Button(form, text="확인", width=20, command=cute)
button.pack()
createtree()
form.mainloop()

```

• match 객체함수의 종류

```

group() :
start() : 처음 시작 하는 위치값 출력
end() : 마지막 위치값 출력
span() : 처음 ~ 마지막까지 위치값 출력

* re.compile 객체의 함수 : match(), search(), findall(), finditer()

p=re.compile("[0-9][0-9]")          # 숫자2자리 패턴 검색
                                   # []를 하나의 패턴으로 인식함
                                   # [0-9] -> \d (소문자)
                                   # 반대 패턴 : [^0-9] = \D (대문자)

result=p.search(temp)
print(result.group())

```

• 패턴

[a-z0-9] = \w --> 반대패턴 [^a-z 0-9]=\W

\s = \n \t " " --> 반대패턴 \S

ex) 주민번호 : 999999-111111

7[0-9]+ : + 는 0 ~ 무한대 (괄호앞 문자가 반복되며 무조건 1번이상 반복되어야함)

7[0-9]* : * 는 0 ~ 무한대 (괄호앞 문자가 반복됨, 반복횟수 상관없음)

75{1,5} : 괄호앞의 문자가 1~5번 횟수

+ --> {1, 공백}
* --> {0, 공백}

```
1. 숫자로 시작하는 모든 패턴 출력
import re

temp="apple is 54 count, apple is red banana is 55 yellow apple is 8888 expensive"
p=re.compile("\d+")
result=p.findall(temp)
for i in result:
    print(str(i),end=" ")

# r \b : 한단어가 끝나는 구분 문자 (r을 적어줘야 단어가 끝나는 부분을 구분자로 인식한다)
# . : 화이트스페이스에서 \n 문자를 제외한 모든 문자 출력

2. 5로 시작하는 숫자패턴 출력
import re

temp="apple is 5555 5 count, apple is red banana is 55 yellow apple is 8888 expensive"
p=re.compile("5\d+")          # + : +앞에 존재하는 숫자로 시작하는 항목이 반드시 1개이상인 항목 출력
                              # * : *앞에 숫자가 존재하지 않아도 됨
result=p.findall(temp)
for i in result:
    print(str(i),end=" ")

3. 숫자와 문자가 연결된 패턴 출력
import re

temp="apple is 5555 5 count, apple is red banana is 55yellow apple is 8888expensive"
p=re.compile("\d+[a-z]+")
result=p.findall(temp)
for i in result:
    print(str(i),end=" ")

4. 숫자로 시작하여 0-9, a-z 등 /n을 제외한 모든 문자 출력
temp="apple is 5555 5 count, apple is red banana is 55yellow apple is 8888expensive"
p=re.compile("\d+.+")
result=p.findall(temp)
for i in result:
    print(str(i),end=" ")

5. 마침표(.) 를 반드시 포함한 문자 출력
import re

temp="apple is 5555 5 count, apple is red banana is 55yellow apple is 8888expensive"
p=re.compile("\d+[.]+")
result=p.findall(temp)
for i in result:
    print(str(i),end=" ")

6. 숫자가 1개로 시작하는 숫자 전체 출력
import re

temp="apple is 5555 5 count, apple is red banana is 55yellow apple is 8 7 888expensive"
p=re.compile("\d?")
result=p.findall(temp)
for i in result:
    print(str(i),end=" ")

7. 숫자가 1개(단일) 인것만 출력
import re

temp="apple is 5555 5 count, apple is red banana is 55yellow apple is 8 7 888expensive"
p=re.compile("\d\s")
result=p.findall(temp)
for i in result:
    print(str(i),end=" ")

8. \로 시작하는 문자 1개 인것 출력
import reennneennnnn9nnnnnnnnn

temp="apple is 5555 5 \count, apple is red banana is 55yellow apple is 8 7 888expensive"
p=re.compile(r"\\w")      # \b, \ 사용과 같은 특수한 경우를 위해 r 사용
result=p.findall(temp)
for i in result:
    print(str(i),end=" ")
```

9. \n을 포함하는 문자까지 출력 (마침표(.) 와 반대개념)

```
import re

temp="a\nb"
p=re.compile("a.b",re.DOTALL)
result=p.findall(temp)
for i in result:
    print(str(i),end=" ")
```

10. 대, 소문자 구분없이 3개로 시작하는 문자 출력

```
import re

temp="abb 777 A7 A7B"
p=re.compile("\w{3}",re.IGNORECASE)
result=p.findall(temp)
for i in result:
    print(str(i),end=" ")
```

11. 첫문자가 'python' 으로 시작하는 문자 출력

```
import re

temp="""python one
life is too short
python two y7u
you need python
python three
"""
p=re.compile("^python\s\w+")
result=p.findall(temp)
for i in result:
    print(str(i),end=" ")
```

12. 각 줄의 'python' 으로 시작하는 문자 출력

```
import re

temp="""python one
life is too short
python two
you need python
python three
"""
p=re.compile("^python\s\w+",re.MULTILINE)
result=p.findall(temp)
for i in result:
    print(str(i),end=" ")
```

13. 마지막 문자가 'one' 으로 시작하는 문자 출력

```
import re

temp="""python one
life is too short
python two
you need python
python three
"""
p=re.compile("one$",re.MULTILINE)
result=p.findall(temp)
for i in result:
    print(str(i),end=" ")
```

• HTML 연계

1. 기본구조

```
<html>
<head>                                     # 선행되어야 될 내용이 삽입된다
<title> 안녕하세요 </title>
</head>
<body>

<h1>본문내용입니다</h1><br>             #<br> : 줄바꿈
<h2>본문내용2 입니다<h2>               #<h2> : 숫자가 클수록 폰트 사이즈 작아진다

</body>
</html>
```


2. 기본형식

```
from bs4 import BeautifulSoup
import requests

req = requests.get("주소")
sentence = req.text
soup = BeautifulSoup(sentence, 'html.parser')

for link in soup.find_all('a'):
    print(link.get('href')) # 링크 뽑아내기
```