

## Review

- 파일이름.ipynb
- 

## ESC + M : 설명달아주기, ESC+Y(소스코드)

## ESC + A, ESC + B, ESC+X(삭제)

In [ ]:



```
jumpCnt = 0
while (jumpCnt<10):
    jumpCnt += 1
    print("점프를 %d번 했습니다." % jumpCnt)
    if jumpCnt ==10:
        print("성공")
```

점프를 1번 했습니다.  
점프를 2번 했습니다.  
점프를 3번 했습니다.  
점프를 4번 했습니다.  
점프를 5번 했습니다.  
점프를 6번 했습니다.  
점프를 7번 했습니다.  
점프를 8번 했습니다.  
점프를 9번 했습니다.  
점프를 10번 했습니다.  
성공

In [ ]:



```

id1 = "toto"
pwd = "1234"

num = 0
while num!=5:
    yid = input("아이디 입력")
    ypwd = input("비밀번호 입력")

    if (id1 == yid) and (pwd==ypwd):
        print("로그인 성공")
        # num = 5
        break
    else:
        print("로그인 실패")
        num += 1

```

아이디 입력toto  
 비밀번호 입력jlkjl  
 로그인 실패  
 아이디 입력toto  
 비밀번호 입력1234  
 로그인 성공

In [ ]:



```

### for문
for i in range(0,10,1):
    print(i, end=" ")

```

0 1 2 3 4 5 6 7 8 9

In [ ]:



```

for i in [1,2,'a',4]:
    print(i)

```

1  
 2  
 a  
 4

In [ ]:



```

word1 = input("하나의 단어")
word2 = input("하나의 단어")
word3 = input("하나의 단어")

ac = word1[0] + word2[0] + word3[0]
print(ac)

```

하나의 단어apple  
 하나의 단어banana  
 하나의 단어price  
 abp

In [ ]:



abcdefghijklmnopqrstuvwxyz

In [ ]:



```
all_str = "" # 패스워드 문자열
for x in range(65,90):
    all_str += chr(x) # 암호만들 대상 문자열
print(all_str)
```

ABCDEFGHIJKLMNOPQRSTUVWXYZ

In [ ]:



```
import random
def passGen():
    password = ""

    # 8글자 암호
    for i in range(10):
        idx = random.randrange(len(all_str))
        password = password + all_str[idx]
    return password
```

In [ ]:



passGen()

Out[20]:

'XWNINNSVNL'

In [ ]:



### (실습 1-1) 10글자 암호, 대문자 A~Z

In [ ]:



```
top = 'toto'
def nameprint():
    print("지역 탐 :", top)

nameprint()
```

지역 탐 : toto

In [ ]:



```
top = 'toto'
def nameprint():
    top = 'banana'
    print("지역 탑 :", top)

nameprint()
print("전국 탑 :", top)
```

지역 탑 : banana  
전국 탑 : toto

In [ ]:



```
top = 'toto'
def nameprint():
    global top
    top = 'banana'
    print("지역 탑 :", top)

nameprint()
print("전국 탑 :", top)
```

지역 탑 : banana  
전국 탑 : banana

In [ ]:



하나의 문자(열) 10, 10, 20 Hello  
,, Hello

- 텍스트 파일 : 사람이 읽을 수 있는 텍스트 포함
- 이진 파일 : 사람이 읽을 수 없으나, 컴퓨터는 읽을 수 있다.

In [ ]:



```
var = open("hello.txt", "r")
```

In [ ]:



```
var.close()
```

In [ ]:



```
file = open("file01.txt", "r")
s = file.read()
print(s)
file.close()
```

fruit, 7000  
tv, 150000  
computer, 890000

## (실습 2-2) 새로운 파일을 만들어서, 파일을 열고 출력, 그리고 닫기 해보기

In [ ]:



```
file = open("file01.txt", "r")
s = file.readline()
print(s, end='')
s = file.readline()
print(s, end='')
s = file.readline()
print(s, end='')
file.close()
```

```
fruit, 7000
tv, 150000
computer, 890000
```

In [ ]:



```
### read() : 전체가져오기
###
file = open("file01.txt", "r")
line = file.readline()

while line != "": # 파일 내용이 있으면. 다음주 읽어라.
    print(line, end='')
    line = file.readline()

file.close()
```

```
fruit, 7000
tv, 150000
computer, 890000
```

## (실습 2-3) 한줄씩 출력할 때, 한줄의 길이도 함께 출력해 주세요.

- 길이 : 5, 출력 : \_\_
- 길이 : 2, 출력 : \_\_
- 길이 : 5, 출력 : \_\_

In [ ]:



```

### read() : 전체가져오기
###
file = open("file01.txt", "r")
line = file.readline()

while line != "": # 파일 내용이 있으면. 다음주 읽어라.
    print("길이:", len(line), "내용:", line, end='')
    line = file.readline()

file.close()

```

길이: 12 내용: fruit, 7000  
 길이: 11 내용: tv, 150000  
 길이: 16 내용: computer, 890000

In [ ]:



```

# 모드를 'w'로 해주면 쓰기 모드가 된다.
# 새롭게 파일이 생성된다.
f1 = open("mydata.txt", "w")
f1.write("monitor, 200000")
f1.close()

```

In [ ]:



```

# 모드를 'a'로 해주면 추가모드
f2 = open("mydata.txt", 'a')
f2.write("monitor2, 230000" + "\n")
f2.close()

```

In [ ]:



```

### read() : 전체가져오기
###
file = open("mydata.txt", "r")
line = file.readline()

while line != "": # 파일 내용이 있으면. 다음주 읽어라.
    print("길이:", len(line), "내용:", line)
    line = file.readline()

file.close()

```

길이: 32 내용: monitor, 200000monitor2, 230000

길이: 17 내용: monitor2, 230000

길이: 17 내용: monitor2, 230000

In [ ]:



```
# os 폴더 생성, 폴더에 파일 확인, 폴더 위치 확인.....
import os

# 현재 작업 위치를 알고 싶다.
print( os.getcwd() ) # 현재 작업 폴더를 표시
loc = os.getcwd()
```

In [ ]:



```
# listdir() 함수는 해당 위치의 파일 및 폴더를 보여준다.
file_list = os.listdir(loc)
file_list
```

Out[79]:

```
['.ipynb_checkpoints',
 'file01.txt',
 'hello.txt',
 'mydata.txt',
 'part01_python_1_1_202009.pdf',
 'part01_python_1_2_202009.pdf',
 'part01_python_1_2_p01.pdf',
 'part01_python_1_3.ipynb',
 'part1_1_2_python_function.ipynb',
 'smalldog.jpg',
 'Untitled.ipynb',
 '수업노트1_1_파이썬기본_20200921.txt',
 '수업노트1_2_파이썬기본_20200922.txt',
 '수업노트1_2_파이썬기본_B.txt',
 '인공지능기본과정_.pptx',
 '학생_20200921.txt']
```

In [ ]:



```
file1 = 'smalldog.jpg'
backup = "smalldog_backup.jpg"

infile = open(file1, 'rb')
outfile = open(backup, 'wb')

while True:
    copy_buffer = infile.read(1024)
    if not copy_buffer: # 읽어올 내용이 없으면(이미지내용)
        break
    outfile.write(copy_buffer)

infile.close()
outfile.close()
print("복사완료")
```

복사완료

In [ ]:

```
file1 = 'smalldog.jpg'
backup = "smalldog_back04.jpg"

infile = open(file1, 'rb')
outfile = open(backup, 'wb')

for i in range(10):
    copy_buffer = infile.read(1024)
    outfile.write(copy_buffer)

infile.close()
outfile.close()
print("복사완료")
```

복사완료

In [ ]:

```
# split() 함수는 문자열에서 해당 구분
# 문자를 이용하여 리스트 형태로 값을 반환해 준다.
line = "Bad: news: travels: fase."
word_list = line.split(":")
print(word_list)
```

['Bad', ' news', ' travels', ' fase.']

In [ ]:

```
# split() 함수는 문자열에서 해당 구분
# 문자를 이용하여 리스트 형태로 값을 반환해 준다.
line = "Bad, news, travels, fase."
word_list = line.split(",")
print(word_list)
```

['Bad, news, travels, fase.']

In [ ]:

```
# 3-1 실습 ", "구분으로 구분시켜 보기
line = "Bad, news, travels, fase."
word_list = line.split(",")
print(word_list)
```

['Bad', ' news', ' travels', ' fase.']

In [ ]:

```
for one in word_list:
    print(one)
```

```
Bad
news
travels
fase.
```



In [ ]:



```
line = "Bad, news, travels, fase."  
dir(line)
```

Out[92]:

```
['__add__',  
 '__class__',  
 '__contains__',  
 '__delattr__',  
 '__dir__',  
 '__doc__',  
 '__eq__',  
 '__format__',  
 '__ge__',  
 '__getattr__',  
 '__getitem__',  
 '__getnewargs__',  
 '__gt__',  
 '__hash__',  
 '__init__',  
 '__init_subclass__',  
 '__iter__',  
 '__le__',  
 '__len__',  
 '__lt__',  
 '__mod__',  
 '__mul__',  
 '__ne__',  
 '__new__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__rmod__',  
 '__rmul__',  
 '__setattr__',  
 '__sizeof__',  
 '__str__',  
 '__subclasshook__',  
 'capitalize',  
 'casefold',  
 'center',  
 'count',  
 'encode',  
 'endswith',  
 'expandtabs',  
 'find',  
 'format',  
 'format_map',  
 'index',  
 'isalnum',  
 'isalpha',  
 'isascii',  
 'isdecimal',  
 'isdigit',  
 'isidentifier',  
 'islower',  
 'isnumeric',
```

```
'isprintable',  
'isspace',  
'istitle',  
'isupper',  
'join',  
'ljust',  
'lower',  
'lstrip',  
'maketrans',  
'partition',  
'replace',  
'rfind',  
'rindex',  
'rjust',  
'rpartition',  
'rsplit',  
'rstrip',  
'split',  
'splitlines',  
'startswith',  
'strip',  
'swapcase',  
'title',  
'translate',  
'upper',  
'zfill']
```

In [ ]:



```
line.__str__()
```

Out[94]:

```
'Bad, news, travels, fase.'
```

In [ ]:



In [ ]:



```
def Cal():  
    pass
```

In [ ]:



```
# 클래스 기본 선언 형태  
class Cal:  
    pass
```

In [ ]:



```
### 인스턴스  
### 2대의 계산기를 만들다.  
a = Cal()  
b = Cal()
```

In [ ]:



```
# 첫번째 계산기
result = 0
def plus1(num):
    global result
    result += num
    return result

print( plus1(3) )
print( plus1(7) )
```

3  
10

In [ ]:



```
# 두번째 계산기
result1 = 0
def plus2(num):
    global result1
    result1 += num
    return result1

print( plus2(2) )
print( plus2(10) )
```

2  
12

In [ ]:



```
# 첫번째 계산기
result = 0
def plus1(num):
    global result
    result += num
    return result

print( plus1(3) )
print( plus1(7) )

# 두번째 계산기
result1 = 0
def plus2(num):
    global result1
    result1 += num
    return result1

print( plus2(2) )
print( plus2(10) )

# 세번째 계산기
result2 = 0
def plus3(num):
    global result2
    result2 += num
    return result2

print( plus3(2) )
print( plus3(10) )
```

```
3
10
2
12
2
12
```

In [ ]:



```
# 첫번째 계산기
result = 0
def plus1(num):
    global result
    result += num
    return result

print( plus1(3) )
print( plus1(7) )
```

```
3
10
```

In [ ]:



```
plus1(3)
```

Out[107]:

13

In [ ]:



```
plus1(1)
```

Out[108]:

14

In [ ]:



```
# 첫번째 계산기
result = 0
def plus1(num):
    global result
    result += num
    return result

# 두번째 계산기
result1 = 0
def plus2(num):
    global result1
    result1 += num
    return result1
```

In [ ]:



```
plus1(2)
```

Out[115]:

2

In [ ]:



```
plus1(5)
```

Out[116]:

7

In [ ]:



```
plus2(1)
```

Out[117]:

1

In [ ]:

```
class Cal:
    result = 0

    def plus(self, num):
        self.result += num
        return self.result

    def minus(self, num):
        self.result -= num
        return self.result

    def divide(self, num):
        self.result /= num
        return self.result
```

In [ ]:

```
# 인스턴스 (객체를 생성)
c1 = Cal()
c2 = Cal()
c3 = Cal()
c4 = Cal()
c5 = Cal()
```

In [ ]:

```
print( c1.plus(3) )
print( c1.plus(3) )

print( c2.plus(4) )
print( c2.plus(4) )
```

21  
24  
28  
32

**(실습 4-1) 내 계산기 클래스에 값을 빼주는 -와 /를 하는 연산을 추가해 보자.**

In [ ]:

```
print( c5.plus(4) )
```

4

In [ ]:

```
c2.plus(3)
```

Out[128]:

3

In [ ]:



```
c2.divide(3)
```

Out[129]:

1.0

In [ ]:



```
c2.minus(2)
```

Out[133]:

-3.0

In [ ]:



```
c3.plus(3)
```

Out[134]:

3

In [ ]:



```
c3.divide(0)
```

```
-----  
-  
ZeroDivisionError                                Traceback (most recent call last)  
<ipython-input-135-c03afd1fc00a> in <module>  
----> 1 c3.divide(0)  
  
<ipython-input-126-d2d530d3388b> in divide(self, num)  
     11  
     12     def divide(self, num):  
----> 13         self.result /= num  
     14         return self.result
```

ZeroDivisionError: division by zero

In [ ]:



```
# 아무 것도 없다.  
# 어떤 클래스를 상속을 받아서,  
# 해당 클래스가 가진 변수 및 메소드를 사용 가능하다.  
class Cal_change(Cal):  
    pass
```

In [ ]:



```
c_ch1 = Cal_change()

# 3을 더하고, 5를 빼기
c_ch1.plus(3)
c_ch1.minus(5)
```

Out[147]:

-2

In [ ]:



```
# 오버라이딩
# 상속받아서 사용하는데, 기존에 상속받은 존재하는 메소드를 변경하는 것.
class Cal_change(Cal):
    def divide(self, num):
        if num==0:
            return "0으로 나눌 수 없습니다."
        else:
            self.result /= num
        return self.result
```

In [ ]:



```
c_ch1 = Cal_change()

# 3을 더하고, 5를 빼기
print( c_ch1.plus(3) )
print( c_ch1.minus(5) )
print( c_ch1.divide(0) )
```

3

-2

0으로 나눌 수 없습니다.

## (실습 4-2) Cal 클래스를 상속받아서,

- 해당 클래스에 하나의 메소드(곱하기)를 추가해 봅시다.
- 그리고 계산기에 C의 기능(0으로 세팅하는) 기능을 추가해 봅시다.
- 계산기로 더하기10, 나누기0, 곱하기 5를 계산하고
  - C를 눌러 초기화하고 5를 더해보자.



In [ ]:



```
# 오버라이딩
# 상속받아서 사용하는데, 기존에 상속받은 존재하는 메소드를 변경하는 것.
class Cal_change(Cal):
    def Czero(self):
        self.result = 0
        return self.result

    def mul(self, num):
        self.result *= num
        return self.result

    def divide(self, num):
        if num==0:
            return "0으로 나눌 수 없습니다."
        else:
            self.result /= num
            return self.result
```

In [ ]:



```
c1 = Cal_change()
print( c1.plus(10) )
print( c1.divide(0) )
print( c1.mul(5) )
print( c1.Czero() )
print( c1.plus(5))
```

```
10
0으로 나눌 수 없습니다.
50
0
5
```

In [ ]:

