10장 파이썬 프로그래밍

10.1 내가 프로그램을 만들 수 있을까?

■ 프로그램을 만들려면 가장 먼저 '입력'과 '출력'을 생각하라.
- 함수이름은? gugudan로 짓자
- 입력받는 값은? 2
- 출력하는 값은? 2단(2, 4, 6, 8, ..., 18)
- 결과는 어떤 형태로 저장하지? 연속된 자료형이니까 리스트!

[ch10_programming/verify.ve01_gugudan.py]

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

def gugudan(n):
    # 이 위치에 코드를 작성한다.


print(gugudan(2))
print(gugudan(3))
print(gugudan(4))
'''
[2, 4, 6, 8, 10, 12, 14, 16, 18]
[3, 6, 9, 12, 15, 18, 21, 24, 27]
[4, 8, 12, 16, 20, 24, 28, 32, 36]
'''
```

10.2 3과 5의 배수 합하기

■ 10 미만의 자연수에서 3과 5의 배수를 구하면 3, 5, 6, 9이다. 이들의 총합은 23이다. 1000 미만의 자연수에서 3의 배수와 5의 배수의 총합을 구하라.
- 입력 받은 값은? 1부터 999까지(1000 미만의 자연수)
- 출력하는 값은? 3의 배수와 5의 배수의 총합
- 생각해 볼 것은?
  - 첫째, 3의 배수와 5의 배수는 어떻게 찾지?
  - 둘째, 3의 배수와 5의 배수가 겹칠 때는 어떻게 하지?

[ch10_programming/verify.ve02_sum.py]

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

result = 0
# 이 위치에 코드를 작성한다.


print(result)
# 233168
```

10.3 게시판 페이징하기

■ A씨는 게시판 프로그램을 작성하고 있다. 그런데 게시물의 총 건수와 한 페이지에 보여줄 게

시물 수를 입력으로 주었을 때 총 페이지수를 출력하는 프로그램이 필요하다고 한다.

- 함수 이름은? getTotalPage
- 입력 받은 값은? 게시물의 총 건수(m), 한 페이지에 보여줄 게시물 수(n)
- 출력하는 값은? 총 페이지수
- 총 페이지수 = 총 건수 / 한 페이지당 보여줄 건수 + 1

[ch10_programming/verify.ve03_paging.py]

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

# 게시판 페이징하기
def getTotalPage(m, n):
    # 이 위치에 코드를 작성한다.




print(getTotalPage(5,10))  # 1
print(getTotalPage(15,10)) # 2
print(getTotalPage(30,10)) # 3
```

10.4 간단한 메모장 만들기

■ 원하는 메모를 파일에 저장하고 추가 및 조회가 가능한 간단한 메모장을 만들어 보자

- 필요한 기능은? 메모 추가하기, 메모 조회하기
- 입력 받은 값은? 메모 내용, 프로그램 실행 옵션
- 출력하는 값은? memo.txt

[ch10_programming/ex04_memo.py]

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

import sys

if len(sys.argv) not in [2,3] :
    print("사용법: python 파일명.py -a/v/h [message]")
    sys.exit()

option = sys.argv[1]

if option == '-a':
    memo = sys.argv[2]
    f = open("memo.txt", 'a')
    f.write(memo)
    f.write('\n')
    f.close()

if option == '-v':
    f = open("memo.txt", 'r')
    memo = f.read()
    print(memo)
    f.close()

if option == '-h':
    print("사용법: python 파일명.py -a/v/h [message]")
```

10.5 탭을 4개의 공백으로 바꾸기

■ 문서 파일을 읽어서 그 문서 파일 내에 있는 탭(tab)을 공백(space) 4개로 바꾸어주는 스크립트를 작성해 보자.
- 필요한 기능은? 문서 파일 일어 들이기, 문자열 변경하기
- 입력 받은 값은? 탭을 포함한 문서 파일
- 출력하는 값은? 탭이 공백으로 수정된 문서 파일

[ch10_programming/ex05_tab.py]

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

import sys

src = sys.argv[1]
dst = sys.argv[2]

f = open(src, 'r')
tab_content = f.read()
f.close()

space_content = tab_content.replace("\t", " "*4)

f = open(dst, 'w')
f.write(space_content)
f.close()
```

10.6 하위 디렉터리 검색하기

■ 하위의 모든 파일 중 파이썬 파일(`*.py`)만 출력해 주는 프로그램을 만들어 보자.

[ch10_programming/ex06_directory.py]

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

import os

def search(dirname):
    try:
        filenames = os.listdir(dirname) # os.listdir는 해당 디렉토리에 있는 파일들의 리스트를 구한다.
        for filename in filenames:
            full_filename = os.path.join(dirname, filename)
            if os.path.isdir(full_filename):
                search(full_filename)
            else:
                ext = os.path.splitext(full_filename)[-1]
                if ext == '.py':
                    print(full_filename)
    except PermissionError:
        pass

search("c:/dev/workspace/")
```

[과제] 연습문제

Q1 문자열 압축하기

문자열을 입력 받아 같은 문자가 연속적으로 반복되는 경우에 그 반복 횟수를 표시해 문자열을 압축하여 표시해 보자.

```
입력 예시: aaabbccccccca
출력 예시: a3b2c6a1
```

[정답]

[ch10_programming/verify.ve04_compress.py]

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

def compress_string(s):
    _c = ""   # 'b'
    cnt = 0   # 3
    result = ""
    for c in s:
        if c != _c:
            _c = c   # 'b'
            if cnt:
                result += str(cnt)
            result += c  #'a3b'
            cnt = 1
        else:
            cnt += 1
    if cnt:
        result += str(cnt)
    return result

print(compress_string("aaabbbbccccccca"))
# a3b4c7a1
```

Q2 Duplicate Numbers

0~9까지의 문자로 된 숫자를 입력받았을 때, 이 입력값이 0~9까지의 모든 숫자가 각각 한 번씩만 사용된 것인지 확인하는 함수를 작성해 보자.

```
입력 예시: 0123456789 01231 01234567890 6789012345 012322456789
출력 예시: true false false true false
```

[정답]

[ch10_programming/verify.ve05_duplicate.py]

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

def duplicatedNum(s):
    result = []
    for num in s:
        if num not in result:
            result.append(num)
```

```
        else:
            return False
    return len(result) == 10

print(duplicatedNum("0123456789"))  # True 리턴
print(duplicatedNum("01234"))  # False 리턴
print(duplicatedNum("01234567890"))  # False 리턴
print(duplicatedNum("6789012345"))  # True 리턴
print(duplicatedNum("012322456789")) # False 리턴
```

~~Q3 모스 부호 해독~~

문자열 형식으로 입력받은 모스 부호(dot:, dash:-)를 해독하여 영어 문장으로 출력하는 프로그램을 작성해 보자.

```
// 글자와 글자 사이에는 공백 1개, 단어와 단어 사이는 공객 2개로 구분한다.
// 예를 들어 다음 모스 부호는 "HE SLEEPS EARLY"로 해석해야 한다.

.... . ... .-.. . . .--. ... . .- .-. .-.. -.--
```

1	●▬▬▬▬	.	●▬●▬●▬	A	●▬	O	▬▬▬
2	●●▬▬▬	,	▬▬●●▬▬	B	▬●●●	P	●▬▬●
3	●●●▬▬	?	●●▬▬●●	C	▬●▬●	Q	▬▬●▬
4	●●●●▬	/	▬●●▬●	D	▬●●	R	●▬●
5	●●●●●	+	●▬●▬●	E	●	S	●●●
6	▬●●●●	−	▬●●●●▬	F	●●▬●	T	▬
7	▬▬●●●	=	▬●●●▬	G	▬▬●	U	●●▬
8	▬▬▬●●	:	▬▬▬●●●	H	●●●●	V	●●●▬
9	▬▬▬▬●	;	▬●▬●▬●	I	●●	W	●▬▬
0	▬▬▬▬▬	(	▬●▬▬●	J	●▬▬▬	X	▬●●▬
		)	▬●▬▬●▬	K	▬●▬	Y	▬●▬▬
		‘	●▬▬▬▬●	L	●▬●●	Z	▬▬●●
		“	●▬●●▬●	M	▬▬	정정부호	●●●●●●●●
		@	●▬▬●▬●	N	▬●		

[정답]

[ch10_programming/verify.ve06_morse.py]

```
dic = {
    '.-':'A','-...':'B','-.-.':'C','-..':'D','.':'E','..-.':'F',
    '--.':'G','....':'H','..':'I','.---':'J','-.-':'K','.-..':'L',
    '--':'M','-.':'N','---':'O','.--.':'P','--.-':'Q','.-.':'R',
    '...':'S','-':'T','..-':'U','...-':'V','.--':'W','-..-':'X',
    '-.--':'Y','--..':'Z'
```

```
}

def morse(src):
    result = []
    for word in src.split("  "):
        for char in word.split(" "):
            result.append(dic[char])
        result.append(" ")
    return "".join(result)


print(morse('.... .  ... .-.. . . .--. ...  . .- .-. .-.. -.--'))
# HE SLEEPS EARLY
```