

---

Python

# 예제모음

---

김선녕(sykim.lecture@gmail.com)

---

**예제 실습**

XML 다루기

10 미만의 자연수에서 3과 5의 배수를 구하면 3, 5, 6, 9이다. 이들의 총합은 23이다.  
1000 미만의 자연수에서 3의 배수와 5의 배수의 총합을 구하라.

- 입력 받는 값은?
  - 1부터 999까지(1000 미만의 자연수)
- 출력하는 값은?
  - 3의 배수와 5의 배수의 총합
- 생각해 볼 것은?
  - 3의 배수와 5의 배수는 어떻게 찾지?
  - 3의 배수와 5의 배수가 겹칠 때는 어떻게 하지?

## 3과 5의 배수 합하기

```
In [ ]: # 1000 미만의 자연수를 구하는 방법을 생각해 보자.  
n = 1  
while n < 1000:  
    print(n)  
    n += 1
```

```
In [ ]: # range 함수 이용  
for n in range(1, 1000):  
    print(n)
```

```
In [ ]: # 3의 배수를 구하는 방법  
for n in range(1, 1000):  
    if n % 3 == 0:  
        print(n)
```

```
In [ ]: # 5의 배수를 구하는 방법  
for n in range(1, 1000):  
    if n % 5 == 0:  
        print(n)
```

## 3과 5의 배수 합하기 - 결과

- 정확한 결과를 확인해 보자

```
In [ ]: # 결과 1
result = 0
for n in range(1, 1000):
    if n % 3 == 0:
        result += n
    if n % 5 == 0:
        result += n
print(result)
```

```
In [ ]: # 결과 2
result = 0
for n in range(1, 1000):
    if n % 3 == 0 or n % 5 == 0:
        result += n
print(result)
```

게시물의 총 건수와 한 페이지에 보여줄 게시물 수를 입력으로 주었을 때 총 페이지수를 출력하는 프로그램을 구하라.

※ 게시판의 페이지 수를 보여주는 것을 "페이징" 한다고 부른다.

- 함수 이름은?
  - getTotalPage
- 입력 받는 값은?
  - 게시물의 총 건수(m), 한 페이지에 보여줄 게시물 수(n)
- 출력하는 값은?
  - 총 페이지수

게시물의 총 건수(m)	페이지당 보여줄 게시물 수(n)	총 페이지 수
5	10	1
15	10	2
25	10	3
30	10	3

```
In [2]: # 총 페이지수 = 총건수 / 한 페이지당 보여줄 건수 + 1
def getTotalPage(m, n):
    return m // n + 1

print(getTotalPage(5, 10))    # 1 출력
print(getTotalPage(15, 10))   # 2 출력
print(getTotalPage(25, 10))   # 3 출력
print(getTotalPage(30, 10))   # 4 출력(???, 생각해 보기)
```

1  
2  
3  
4

```
In [3]: def getTotalPage(m, n):
        if m % n == 0:
            return m // n
        else:
            return m // n + 1

print(getTotalPage(5, 10))
print(getTotalPage(15, 10))
print(getTotalPage(25, 10))
print(getTotalPage(30, 10))
```

1  
2  
3  
3

**Q : 만약 데이터가 없으면?**

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

(ex) `python memo.py -a "Life is too short"`

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



```
# C:/Python/memo.py
import sys

option = sys.argv[1]
memo = sys.argv[2]

print(option)
print(memo)
```

# 다음과 같이 수행

```
C:\Python>python memo.py -a "Life is too short"
-a
Life is too short
```

```
# C:/Python/memo.py - 코드 변경(입력으로 받은 메모를 파일에 쓴다.)
```

```
import sys
```

```
option = sys.argv[1]
```

```
if option == '-a':
```

```
    memo = sys.argv[2]
```

```
    f = open('memo.txt', 'a')
```

```
    f.write(memo)
```

```
    f.write('\n')
```

```
    f.close()
```

```
# 다음과 같이 수행
```

```
C:\Python>python memo.py -a "Life is too short"
```

```
C:\Python>python memo.py -a "You need python"
```

```
# 메모 확인
```

```
C:\Python>type memo.txt
```

```
Life is too short
```

```
You need python
```

```
# 메모 출력하는 부분 -v 옵션
```

```
C:\Python>python memo.py -v
```

```
# C:/Python/memo.py - 코드 변경(-v 옵션 추가)
```

```
import sys
```

```
option = sys.argv[1]
```

```
if option == '-a':
```

```
    memo = sys.argv[2]
```

```
    f = open('memo.txt', 'a')
```

```
    f.write(memo)
```

```
    f.write('\n')
```

```
    f.close()
```

```
elif option == '-v':
```

```
    f = open('memo.txt')
```

```
    memo = f.read()
```

```
    f.close()
```

```
    print(memo)
```

```
# 다음과 같이 수행
```

```
C:\Python>python memo.py -v
```

```
Life is too short
```



```
You need python
```

- 프로젝트 오일러(<http://projecteuler.net/archives>)

**Project Euler**.net

AboutArchivesRecentNewsRegisterSign In

Cache update: 49 minutes



## Problems Archives

The problems archives table shows problems 1 to 637. If you would like to tackle the 10 most recently published problems then go to [Recent problems](#). Click the description/title of the problem to view details and submit your answer.

1234567891011...13Go to Problem:

ID	Description / Title	Solved By
1	Multiples of 3 and 5	806187
2	Even Fibonacci numbers	645365
3	Largest prime factor	461378
4	Largest palindrome product	409431
5	Smallest multiple	416945
6	Sum square difference	419512
7	10001st prime	358799
8	Largest product in a series	301758
9	Special Pythagorean triplet	305557
10	Summation of primes	279838

# 예제 실습

## **XML 다루기**

다음과 같은 구조의 XML문서를 생성해 보자.

```
<note date="20120104">  
  <to>Tove</to>  
  <from>Jani</from>  
  <heading>Reminder</heading>  
  <body>Don't forget me this weekend!</body>  
</note>
```

In [1]: *# 엘리먼트(Element)를 이용하면 태그를 만들 수 있고,  
# 만들어진 태그에 텍스트 값을 추가할 수 있음을 알 수 있다.*

```
from xml.etree.ElementTree import Element, dump
```

```
note = Element("note")  
to = Element("to")  
to.text = "Tove"
```

```
note.append(to)  
dump(note)
```

```
<note><to>Tove</to></note>
```

In [2]: *# 서브엘리먼트(SubElement)를 이용하면 조금 더 편리하게 태그를 추가할 수 있다.*

```
from xml.etree.ElementTree import Element, SubElement, dump
```

```
note = Element("note")  
to = Element("to")  
to.text = "Tove"
```

```
note.append(to)
```

*# 서브엘리먼트는 태그명과 태그의 텍스트 값을 한 번에 설정할 수 있다.*

```
SubElement(note, "from").text = "Jani"
```

```
dump(note)
```

```
<note><to>Tove</to><from>Jani</from></note>
```

In [4]: *# 다음과 같이 태그 사이에 태그를 추가하거나 특정 태그를 삭제할 수도 있다.*

```
dummy = Element("dummy")  
note.insert(1, dummy)  
note.remove(dummy)
```

In [7]: `from xml.etree.ElementTree import Element, SubElement, dump`

```
note = Element("note")
to = Element("to")
to.text = "Tove"

note.append(to)
SubElement(note, "from").text = "Jani"
note.attrib["date"] = "20120104"  # 애트리뷰트 값(attribute value) 추가

dump(note)
```

`<note date="20120104"><to>Tove</to><from>Jani</from></note>`

In [8]: `# 완성된 소스`

```
from xml.etree.ElementTree import Element, SubElement, dump

note = Element("note")
note.attrib["date"] = "20120104"

to = Element("to")
to.text = "Tove"
note.append(to)

SubElement(note, "from").text = "Jani"
SubElement(note, "heading").text = "Reminder"
SubElement(note, "body").text = "Don't forget me this weekend!"
dump(note)
```

`<note date="20120104"><to>Tove</to><from>Jani</from><heading>Reminder</heading><body>Don't forget me this weekend!</body></note>`



In [9]: `from xml.etree.ElementTree import Element, SubElement, dump` *# 정렬된 XML값*

```
note = Element("note")
note.attrib["date"] = "20120104"

to = Element("to")
to.text = "Tove"
note.append(to)

SubElement(note, "from").text = "Jani"
SubElement(note, "heading").text = "Reminder"
SubElement(note, "body").text = "Don't forget me this weekend!"
```

```
def indent(elem, level=0):
    i = "\n" + level*"  "
    if len(elem):
        if not elem.text or not elem.text.strip():
            elem.text = i + " "
        if not elem.tail or not elem.tail.strip():
            elem.tail = i
        for elem in elem:
            indent(elem, level+1)
        if not elem.tail or not elem.tail.strip():
            elem.tail = i
    else:
        if level and (not elem.tail or not elem.tail.strip()):
            elem.tail = i
```

```
indent(note)
dump(note)
```

```
<note date="20120104">
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

## 파일에 쓰기(write) 수행하기

18

```
In [10]: from xml.etree.ElementTree import ElementTree  
         ElementTree(note).write("note.xml")
```

0 ▾ / 7장_xml		Name ▾	Last Modified	File size
..			몇 초 전	
python_lecture71.ipynb			Running 6분 전	
note.xml			1분 전	

- 애트리뷰트 값 읽기

```
In [10]: from xml.etree.ElementTree import ElementTree
         ElementTree(note).write("note.xml")
```

```
In [11]: from xml.etree.ElementTree import parse
         tree = parse("note.xml")
         note = tree.getroot()
```

```
In [12]: print(note.get("date"))
         print(note.get("foo", "default"))
         print(note.keys())
         print(note.items())
```

```
20120104
default
['date']
[('date', '20120104')]
```