



# 경영 빅데이터 프로그래밍

TA SESSION 1

---

(4) 파이썬 primer



# CONTENTS

## PRESENTATION



1. 파이썬이란?
2. 파이썬 코드 작성 규칙과 가이드
3. 파이썬 프로그래밍 첫 걸음 떼기



**파이썬(Python)이란?**

# 파이썬(Python)이란?

○ 1991년 귀도 반 로섬(Guido van Rossum)이 발표한 고급 프로그래밍 언어이다



○ 파이썬은 플랫폼 독립적이며 인터프리터식, 객체지향적, 동적 타이핑 대화형 언어이다

○ 웹 개발, 데이터베이스 개발, 데이터 분석 등 다양한 목적을 위해 널리 쓰이는 언어로 초보자 뿐만 아니라 전문가들도 애용하는 언어이다





파이썬은 [       ]하다



## 파이썬의 특징

- 파이썬은 [쉽고 간단]하다



- 'Hello World!'를 다른 언어로 표현해보자



# 파이썬의 특징

## ○ 'Hello World!'를 자바로 출력하기

```
example.java x example.py x example.cc x
1 public class HelloWorld
2 {
3     public static void main(String[] args) {
4         System.out.println("Hello World!");
5     }
6 }
```

## ○ 'Hello World!'를 c++로 출력하기

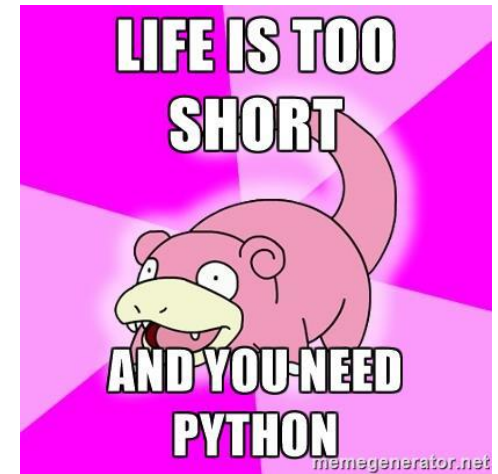
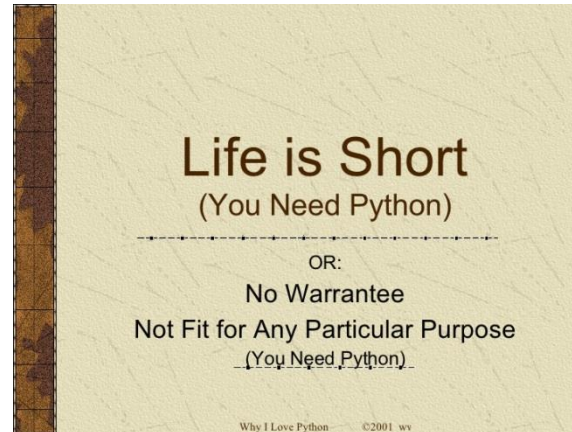
```
example.java x example.py x example.cc x
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello, World!";
7     return 0;
8 }
```

## ○ 'Hello World!'를 파이썬으로 출력하기

```
example.java x example.py x example.cc x
1 print('Hello World!')
```

# 파이썬의 특징

○ 파이썬은 [빠르고 생산적]이다



- 직관적이면서도 빠른 파이썬은 개발자들이 사랑하는 언어이다





# 파이썬의 특징

## ○ 파이썬은 [강력]하다

- 파이썬은 오픈소스(Opensource)이므로 완전히 무료로 사용할 수 있다
- 특히, 데이터 분석을 위한 다양한 패키지와 API가 무료로 제공되어 손쉽게 분석을 수행할 수 있다
  - **넘파이(Numpy)**: 수학 및 과학 연산을 위한 패키지
  - **판다스(Pandas)**: 데이터 구조화를 용이하게 해주는 패키지
  - **사이킷-런(Scikit-learn)**: 파이썬에서 기계 학습(Machine learning) 구현을 위한 가장 대중적인 패키지
  - **텐서플로(Tensorflow)**: 구글 제품에 사용되는 딥 러닝(Deep learning)에 특화된 오픈소스 라이브러리
  - **네트웍스(Networkx)**: 그래프와 네트워크 분석 및 시각화를 위한 패키지
  - **매트플롯립(Matplotlib)**: 데이터 시각화를 위한 플롯 패키지



# 파이썬의 특징

## ○ 파이썬은 [도움을 구하기 쉽]다

- 파이썬은 오픈소스(Opensource) 패키지가 많을 뿐 아니라, 유저층이 두꺼운 만큼 커뮤니티(communitiy)가 활성화 되어 있어 코딩을 하다가 막히면 도움을 받기 용이하다
- 질의응답(Social Q&A service): <https://stackoverflow.com>

The screenshot shows the Stack Overflow homepage with the search bar containing "[python]". The "Tagged Questions" section is active, displaying a list of questions. The first question is "Providing initial data django unit test" with 0 votes, 0 answers, and 8 views. The second question is "Python - Radio Button or Check Box Variables Cause 'Is Not Defined' Error" with 0 votes, 0 answers, and 5 views. On the right, there is a box for "Python Language DOCUMENTATION" and a list of "Related Tags" including django, python-2.7, python-3.x, pandas, numpy, list, matplotlib, and dictionary.

stackoverflow Questions Jobs Documentation BETA Tags Users [python] ? Log In Sign Up

Tagged Questions info newest 34 featured frequent votes active unanswered

Python is a dynamic and strongly typed programming language designed to emphasize usability. Two similar but mostly incompatible versions of Python are in widespread use (2 and 3). If you have a version-specific Python question consider using the [python-2.7] or [python-3.x] tags in addition to the [...]

learn more... top users synonyms (3) python jobs

0 votes  
0 answers  
8 views

Providing initial data django unit test

I working on a Django application which have a lot of unit tests. Some of those test functionalities on unmanaged models (Manage = False). In order to test them, there are in setUpClass the execution ...

python sql django unit-testing

asked 3 mins ago  
Giordano  
1,605 2 6 25

0 votes  
0 answers  
5 views

Python - Radio Button or Check Box Variables Cause "Is Not Defined" Error

I'm having difficulty with Python 3.6.1 and a GUI utilizing Tkinter that I built with the help of the PAGE GUI builder. When I try to get the output of radio buttons or check boxes, I'm getting an ...

python user-interface checkbox tkinter radio-button

asked 3 mins ago  
JBM  
25 10

762,893 questions tagged  
python about »

Ask Question

Python Language DOCUMENTATION  
Find a request to handle or browse 192 topics.

Related Tags

- django × 73471
- python-2.7 × 45085
- python-3.x × 40972
- pandas × 38494
- numpy × 33058
- list × 25129
- matplotlib × 20496
- dictionary × 17872

- 762,893 Questions tagged "Python"



# 파이썬의 특징

## ○ 파이썬은 [도움을 구하기 쉽]다

- 스택 오버플로(Stackoverflow)에서 답 찾기

### Search

python list append

search

6,867 results

relevance

newest

votes

active

3006

votes

#### A: append vs. extend

**append**: Appends object at end. `x = [1, 2, 3] x.append([4, 5]) print (x)` gives you: `[1, 2, 3, [4, 5]]` **extend**: Extends **list** by appending elements from the iterable. `x = [1, 2, 3] x.extend([4, 5]) print (x)` gives you: `[1, 2, 3, 4, 5]` ...

answered Oct 31 '08 by kender

1988

votes

#### Q: append vs. extend

What's the difference between the **list** methods **append()** and **extend()**? ...

python list append extend

asked Oct 31 '08 by Claudiu

19

answers

1

vote

#### Q: Python list appending issue

I ran into a rather weird problem with **python list** appending today. I was trying to create an array whose each element would be like a C struct. One of these elements was a **list** by itself ... This is the problematic code: `class players: name='placeholder' squad=list() teams=list() teams.append(players()) teams.append(players()) teams[0].name="abc" teams[1].name="xyz" teams[0].squad.append ...`

python list append

asked Apr 21 '16 by Aswin G

1

answer

Ask Question

Advanced Search Tips

results found containing  
**python list append**

### Hot Network Questions

- Attacker circumventing 2FA. How to defend?
- Password Checker in Python
- How do I put my Blender screen back to normal, to how I first got it?
- How to get children active in graphic design?
- How should I respond to a colleague who refuses to email me task requests?
- Where does the word 青年 come from?
- correlation does not imply causation, but
- How did "tech-supportcenter" phishers trick Google?
- find command not working: "illegal option --n"
- Is it okay to interrupt Memtest86+?
- Somebody who pretends to understand, but

- "When you are stuck, GOOGLE it"



# 파이썬의 특징

## ○ 파이썬은 [도움을 구하기 쉽]다

- 그 외에도 깃허브(GitHub)나 다큐멘테이션, 각종 블로그 등에서 유용한 정보를 찾아내기 쉽다

- <https://github.com>



## KoNLPy

Translations

English  
한국어

Related Pages

Home

- 사용 예시
  - Previous page: 랜덤 텍스트 생성하기
  - Next page: KoNLPy를 이용한 멀티쓰레딩

Note:

You are not using the most up to date version of the library. [0.4.4](#) is the newest version.

## 워드클라우드 그리기

다음은 파이썬만으로 웹에서 국회 의안의 내용을 수집해서 이용해 명사를 추출한 후, 워드클라우드를 그리는 예시입니다.

의안 번호(i.e., `bill_num`)를 바꿔가며 워드클라우드가 어떻게 달라지는지 확인할수도 있습니다. (ex: '1904882', '1904883', 'ZZ19098', etc)

```
#!/usr/bin/python2.7
# -*- coding: utf-8 -*-

from collections import Counter
import urllib
import random
import webbrowser

from konlpy.tag import Hannanum
from lxml import html
import pytagcloud # requires Korean font support
import sys

if sys.version_info[0] >= 3:
    urlopen = urllib.request.urlopen
else:
    urlopen = urllib.urlopen
```



# 파이썬의 특징

## ○ 파이썬은 [도움을 구하기 쉽]다

- 최근에는 한국에서 파이썬의 인기가 하늘로 치솟으며, 한글로 된 자료도 찾아보기 쉽다

2016년 8월 30일

## 파이썬 형태소 분석으로 워드클라우드 그리기

오늘은 특정 맛집에 대한 블로그 포스팅에서 형태소 분석을 통해 워드클라우드를 그려보고자 합니다.

사전에 블로그 크롤링에 대한 내용을 다뤄야하나, 블로그 별로 포맷이 상이하게 달라 본문 내용 추출에 대한 연구가 따로 필요할 듯하여 이번엔 제외하겠습니다.

해당 사이트는 <http://1moment.tistory.com/57> 입니다.

계절법상에 대한 내용이며 본문 내용만 복사하여 blog\_data.txt로 저장 해 둔 상태입니다.

필요한 패키지를 불러와 봅니다.

```

1 from collections import Counter
2 from konlpy.tag import Twitter
3 import pytagcloud

```

konlpy와 pytagcloud는 없으면 설치해주세요.

pytagcloud는 pygame 패키지에 의존적이라 pygame 패키지도 설치해야 할 수도 있습니다.

collections.Counter은 나타난 단어들을 쉽게 집계하기 위해서 사용합니다.

### 상위 글 및 페이지

- [JAVA에서 LogBack을 이용한 로깅 \(logging\) - 사육방](#)
- [Jupyter\(IPython\) 설치 및 실행](#)
- [파이썬 데이터 분석에 pandas와 matplotlib을 활용하기](#)
- [\[R\] tf-idf 에 대해 알아봅시다](#)

[한글을 이용한 데이터마이닝 및 word2vec 이용한 유사도 분석](#) | Word2Vec, 자연어 / 머신러닝, 딥러닝

2015.06.09, 00:37

<http://blog.naver.com/2feelus/220364206922> [복사](#)

[번역하기](#)

[전용뷰어 보기](#)

<http://www.lucypark.kr/courses/2015-ba/text-mining.html#>

위의 링크에서 본 것들을 위주로 정리했다. 서울대에서 진행하는 datamining course인것 같다. 많은 도움이 되었으나 나름대로 다시 한글화 해서 정리한다.

### A. 설치

1. prototyping에 python만한게 없는 것 같다.파이선설치 확인

2. python용 버전관리 소프트 pip설치(Mac기반, 다른 OS라도 python만 돌아간다면야.)  
sudo easy\_install pip

3. gensim 설치 ( Gensim 이라는 파이선 기반의 text mining library를 다운받는다. 토픽 모델링 및 ,word2vec도 지원한다.)  
sudo pip install -U gensim

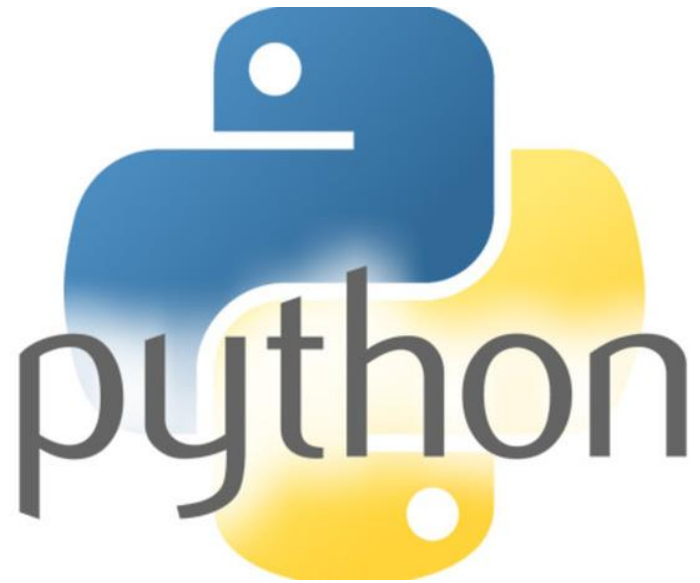
4. NLTK설치(자연어 처리를 위한 광범위하게 쓰이는 python lib)  
sudo pip install nltk

5. KoNLPy설치 (한글처리를 위해)  
sudo pip install konlpy

6. twython 설치 (twitter api쉽게 사용하기위해)  
sudo pip install twython



## 파이썬의 특징



**Why not Python?**



## 파이썬 코드 작성 규칙과 가이드 (Coding Convention)



## 파이썬 코드 작성 규칙과 가이드

- 앞서 살펴봤듯이, 파이썬은 자바나 C++에 비하면 코드 작성을 위한 규칙이 간단하고 까다롭지 않다
- 그렇지만, 파이썬 코딩 규칙과 가이드를 따를 경우 코드의 가독성(readability)을 굉장히 높일 수 있다
  - 내가 작성한 코드를 다른 사람이 이해할 수 있도록 하는 것도 프로그래밍의 중요한 부분이다
  - 뿐만 아니라, 정돈되지 않은 코드는 시간이 지난 후에 작성자가 보더라도 이해하기 힘들다
  - 그러므로, 프로그래밍을 처음 배울 때 부터 코드 작성 규칙과 가이드를 성실히 익히고 코딩을 할 때 이를 최대한 지키기 위해 노력하는 것이 필요하다
  - “One of Guido's key insights is that code is **read much more often than it is written.**”





# 파이썬 코드 작성 규칙과 가이드

## ○ 파이썬 코드 작성 가이드(PEP 8 – Style Guide for Python code)

- 페이지: <https://www.python.org/dev/peps/pep-0008/>
- 파이썬을 만든 귀도 반 로섬 등이 작성한 파이썬 코딩 “스타일 가이드”
- 일부 스타일 가이드는 강제적(지키지 않으면 오류가 발생)이지만 일부 스타일 가이드는 사용자의 선택에 따른 것으로 가독성을 높이기 위함이다
- 복잡하고 양이 많은 가이드라인을 모두 암기할 수는 없겠지만, 즐겨찾기로 두고 참고하면서 코딩을 정돈되게 하려고 노력하는 것이 중요하다



# 파이썬 코드 작성 규칙과 가이드

## ○ 파이썬 코드 작성 가이드(PEP 8 – Style Guide for Python code)

### **A Foolish Consistency is the Hobgoblin of Little Minds**

One of Guido's key insights is that code is read much more often than it is written. The guidelines provided here are intended to improve the readability of code and make it consistent across the wide spectrum of Python code. As [PEP 20](#) says, "Readability counts".

A style guide is about consistency. Consistency with this style guide is important. Consistency within a project is more important. Consistency within one module or function is the most important.

However, know when to be inconsistent -- sometimes style guide recommendations just aren't applicable. When in doubt, use your best judgment. Look at other examples and decide what looks best. And don't hesitate to ask!

In particular: do not break backwards compatibility just to comply with this PEP!

Some other good reasons to ignore a particular guideline:

1. When applying the guideline would make the code less readable, even for someone who is used to reading code that follows this PEP.
2. To be consistent with surrounding code that also breaks it (maybe for historic reasons) -- although this is also an opportunity to clean up someone else's mess (in true XP style).
3. Because the code in question predates the introduction of the guideline and there is no other reason to be modifying that code.
4. When the code needs to remain compatible with older versions of Python that don't support the feature recommended by the style guide.



# 파이썬 코드 작성 규칙과 가이드

## ○ 파이썬 코드 작성 가이드(PEP 8 – Style Guide for Python code)

### **A Foolish Consistency is the Hobgoblin of Little Minds**

One of Guido's key insights is that code is read much more often than it is written. The guidelines provided here are intended to improve the readability of code and make it consistent across the wide spectrum of Python code. As [PEP 20](#) says, "Readability counts".

A style guide is about consistency. Consistency with this style guide is important. Consistency within a project is more important. Consistency within one module or function is the most important.

**However, it is never to be inconsistent – sometimes style guide recommendations just aren't applicable. When in**  
**Style guide is all about consistency(일관성)!**

In particular: do not break backwards compatibility just to comply with this PEP!

Some other good reasons to ignore a particular guideline:

1. When applying the guideline would make the code less readable, even for someone who is used to reading code that follows this PEP.
2. To be consistent with surrounding code that also breaks it (maybe for historic reasons) -- although this is also an opportunity to clean up someone else's mess (in true XP style).
3. Because the code in question predates the introduction of the guideline and there is no other reason to be modifying that code.
4. When the code needs to remain compatible with older versions of Python that don't support the feature recommended by the style guide.



# 파이썬 코드 작성 규칙과 가이드

- 작성 규칙과 가이드를 지키지 않는 코드는 마치 띄어쓰기와 맞춤법을 무시하는 한 편의 시와 같다

거울 이상

거울속에는소리가없소  
저렇게까지조용한세상은참없을것ियो

거울속에도내게귀가있소  
내말을못알아듣는딱한귀가두개나있소

거울속의나는왼손잡ियो  
내악수를받을줄모르는-악수를모르는왼손잡ियो

거울때문에나는거울속의나를만져보지못하는구료마는  
거울이아니었던들내가어찌거울속의나를만나보기만이라도했겠  
소

나는지금거울을안가졌소마는거울속에는늘거울속의내가있소  
잘은모르지만외로된사업에골몰할게요

거울속의나는참나와는반대요마는



- 읽고 이해하기 매우 힘들다



## 파이썬 프로그래밍 첫 걸음 떼기



# 파이썬 코드

## ○ 지난 세션에 작성했던 파이썬 코드(example.py)를 다시 보자

```
x = 'Hello'
y = 'World'
z = x + y
print(z)
```

- x, y, z: 변수(variable). 현재로서는 값을 저장하는 곳이라고만 알아두자
- +: 연산자(operator). 변수들 혹은 값들 간의 연산을 수행한다
- print(): 함수(function). 괄호 안의 값을 출력한다. print() 함수는 앞으로의 가장 많이 사용되는 함수 중 하나이므로 꼭 알아두자



# 파이썬 기초 문법(Syntax)

- 파이썬 문법의 가장 큰 구별되는 특징은 중괄호({})나 세미콜론(;)이 아닌 들여쓰기(indentation)으로 블록을 구분한다는 것이다

- 이는 순전히 가독성을 높이기 위함으로, 자바와 같은 언어에서는 들여쓰기가 권장사항이지만 파이썬에서는 강제사항이다
- 파이썬 코드: x와 y변수의 값을 비교해 결과를 출력한다

```
if(x>y):  
    print("x is greater than y")  
else:  
    print("x is less than or equal to y")
```

- 자바 코드: x와 y변수의 값을 비교해 결과를 출력한다

```
if(x>y){  
    System.out.println("x is greater than y");  
} else {  
    System.out.println("x is less than or equal to y");  
}
```

자바에서는 들여쓰기를 하지 않아도 오류가 나지 않는다



# 파이썬 기초 문법(Syntax)

## ○ 파이썬에서 주석 처리(annotation)을 하는 방법에는 두 가지가 있다

- 주석처리를 하는 이유는 코드에 대한 설명을 덧붙여 이해도를 높이고 코드 재사용을 용이하게 하기 위함이다
- 단일 문장(single line) 주석의 경우 우물 정(#) 기호를 사용한다

```
# 이것은 주석입니다
```

- 복수 문장(multiple lines) 주석의 경우 쉼표 세개(''' ) 를 사용한다

```
'''이것은  
주석  
입니다  
'''
```





## 파이썬 기초 문법(Syntax)

○ 주석은 코드를 구동할 때 실행되지 않는다. 순전히 코드의 이해를 돕기 위해 삽입되는 문장이다

- 예시

```
print('주석이 아닙니다')  
# print('주석입니다')
```

- 결과

```
주석이 아닙니다
```



# 파이썬 기초 문법(Syntax)

○ 대부분의 텍스트 에디터와 IDE에서 주석은 코드와 구별되는 폰트나 색깔로 구분된다

- 노트패드++에서는 연한 녹색으로 표시된다

```
1  # This program adds two numbers
2  num1 = 1.5
3  num2 = 6.3
4
5  # Add two numbers
6  sum = float(num1) + float(num2)
7
8  # Display the sum
9  print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

- 서브라임 텍스트에서는 옅은 회색으로 표시된다

```
1  # This program adds two numbers
2  num1 = 1.5
3  num2 = 6.3
4
5  # Add two numbers
6  sum = float(num1) + float(num2)
7
8  # Display the sum
9  print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
10
```



## 변수와 데이터 타입(Data type)

○ 파이썬에서 선언되는 모든 변수는 데이터 타입을 가진다. 파이썬의 모든 데이터 타입은 아래와 같다

- 불린(boolean)
- 인티저(integer)
- 롱(long)
- 플롯(float)
- 스트링(string)
- 리스트(list)
- 닌(None)

○ 이 중 많이 사용하는 타입은 불린, 인티저, 스트링 정도로 현재로써는 이 세 가지 타입에 대해서만 알아보자

※ 변수의 이름은 문자와 숫자, 그리고 언더라인('\_')만 포함할 수 있다



## 변수와 데이터 타입(Data type)

- 불린: 참(True) 혹은 거짓(False) 값만을 가지는 데이터 타입

```
bool1 = True  
bool2 = False
```

- 인티저: 정수 값을 가지는 데이터 타입

```
i1 = 10  
i2 = 50
```

- 스트링: 문자를 값으로 가지는 데이터 타입

```
s1 = 'kim'  
s2 = 'lee'
```



## 변수와 데이터 타입(Data type)

○ print 함수를 이용하면 변수의 값을 출력할 수 있다

- 예시

```
bool = True
name = 'Kim'
num = 26
print(bool)
print(name)
print(num)
```

- 결과

```
True
Kim
26
```



## 실습(Learning by Doing)

- bool 변수(불린 타입)에 False값을, company 변수(스트링 타입)에 'Samsung' 값을, height 변수(인티저 타입)에 175 값을 입력하고 이들을 출력해 보자

- 출력 예시

```
> ex_1.py  
False  
Samsung  
175
```



# 연산자(Operator)

○ 파이썬에는 변수 간에, 값들 간에, 혹은 변수와 값 간의 연산을 수행하기 위한 연산자가 여러 가지 정의되어 있다. 이러한 연산자들은 대부분 직관적이며 이해하기 쉽다

- 산수 연산자: +, -, \*, /, % (mod)
- 비교 연산자: ==, !=, <, >, <=, >=
- 논리 연산자: and, or, not
- 지수 연산자: \*\*

## ○ 산수 연산자 예시

```
a = 20
b = 10
c = a + b
d = a - b
e = a * b
f = a / b
g = a % b
```



## 연산자(Operator)

- 산수 연산자는 인티저, 롱, 플롯 타입에 주로 적용되며 일반적인 수학의 사칙연산을 생각하면 이해가 쉽다

- 예시

```
n1 = 100  
n2 = 200  
n3 = n1 + n2  
print(n3)
```

- 결과

```
300
```





## 실습(Learning by Doing)

- n1 변수(인티저 타입)에 1000을, n2 변수에 2000을, n3 변수에 3000을 입력하고 n4 변수에 n2에서 n1을 뺀 값을, n5 변수에 n1과 n3를 더한 값을 지정해 준 다음에 n1 부터 n5 까지 값을 모두 출력해 보자

- 출력 예시

```
> ex_2.py  
1000  
2000  
3000  
1000  
4000
```



**THANK YOU!**