

6주차

파이썬

- 파이썬 소개 및 설치
- 변수에 값 저장
- 입력
- 출력
- 조건부 수행

		강의 주제	주 차
1부 컴퓨팅 사고력	컴퓨팅 사고력의 소개	IT 사회, 소프트웨어 세상, 컴퓨팅 사고력의 소개, 컴퓨팅 사고력의 개념	1
		컴퓨팅 사고력의 개념, 주위에서 볼 수 있는 컴퓨팅 사고력, 문제해결 방법	2
	문제해결 방법, 컴퓨터	문제해결 방법, 문제해결 과정 예, 문제해결을 위한 소프트웨어 설계 사상, 컴퓨터의 특징, 소프트웨어, 유한상태기계	3
2부 소프트웨어	알고리즘	알고리즘 소개, 알고리즘의 표현 방법, 의사코드, 흐름도	4
	프로그램	프로그램의 기능, 함수, 컴파일러	5
	파이썬	파이썬 소개 및 설치, 변수에 값 저장, 입력, 출력, 조건부 수행	6
		반복, 리스트, 함수, 출력 형식	7
3부 컴퓨팅 사고력 활용하기	데이터의 표현	이진수, 아스키코드, 오디오 데이터, 이미지 데이터, 자료구조	8
		인코딩 및 압축, 오류확인	9
	데이터의 저장과 검색	배열 및 연결 리스트, 선형검색, 이분검색, 색인순차검색, 해싱	10
		이진검색트리, 최대값 및 최소값 검색	11
	알고리즘설계	정렬, 분할정복 알고리즘, 탐욕적 알고리즘	12

파이썬을 배우는 이유

1. 컴퓨터를 동작시키는 언어를 배워서 컴퓨팅 사고력을 익힌다.
 - 절차적으로 진행되는 작업을 이해
2. 컴퓨터가 할 수 있는 일과 할 수 없는 일을 알게된다.
 - 컴퓨터의 능력과 한계를 이해한다.



3. 작은 프로그램을 완성할 수 있는 능력을 키운다.

- 작은 프로그램이라고 하더라도 기본적인 기능을 갖추어야 한다.
- 대규모 프로그램도 결국 작은 프로그램들의 결합으로 완성

4. 다른 프로그래밍 언어를 배울 수 있는 토대를 갖는다.

- 하나의 프로그래밍 언어를 익히면 다른 프로그래밍 언어를 쉽게 배울 수 있다.
- 프로그래밍 언어들은 서로 유사한 부분을 공유한다.
- JAVA, C++ 등



5. 작성된 프로그램 코드를 읽을 수 있는 능력을 키운다.

- 다른 사람이 작성한 프로그램을 이해하여 본인의 작업에 도움을 받을 수 있다.
- 소프트웨어의 동작 원리를 파악하여 개선할 점을 제시할 수 있다.



파이썬

- 1991년 처음 만들어 짐
- 인터프리터 언어의 일종
- high-level 언어
- 객체지향(object-oriented) 개념을 포함
- 단순한 형식과 쉬운 가독성(readability)을 목표
- 오픈소스(open-source) 소프트웨어로 비영리 단체에 의해 운영
 - 무료 사용

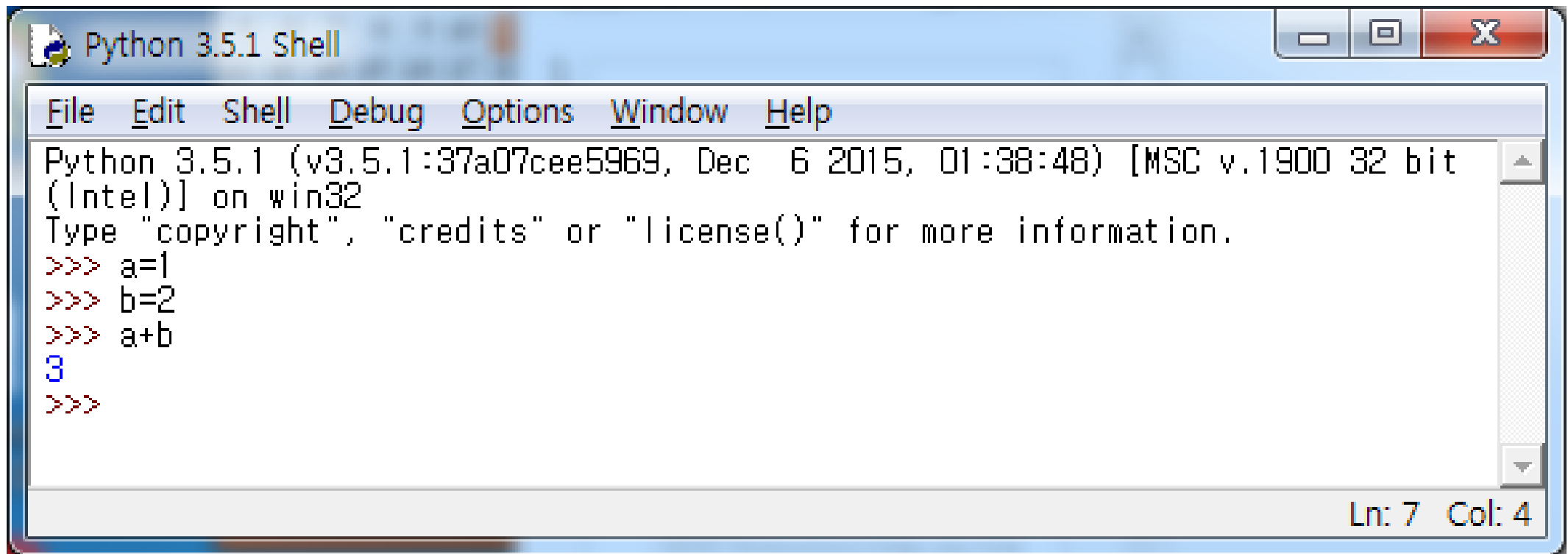


인터프리터(Interpreter) 언어

- 컴파일 과정을 거치지 않고 실행되는 컴퓨터 프로그램 언어
- Python, MATLAB, BASIC 등의 언어
- 언어가 단순하다.
- 인터프리터 : 인터프리터 언어로 작성된 프로그램을 수행하는 프로그램



- IDLE: Integrated Development and Learning Environment (Integrated Development Environment 의 일종)
- 파이썬을 이용하기 위해서는 IDLE를 설치해야 함



The screenshot shows a window titled "Python 3.5.1 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following content:

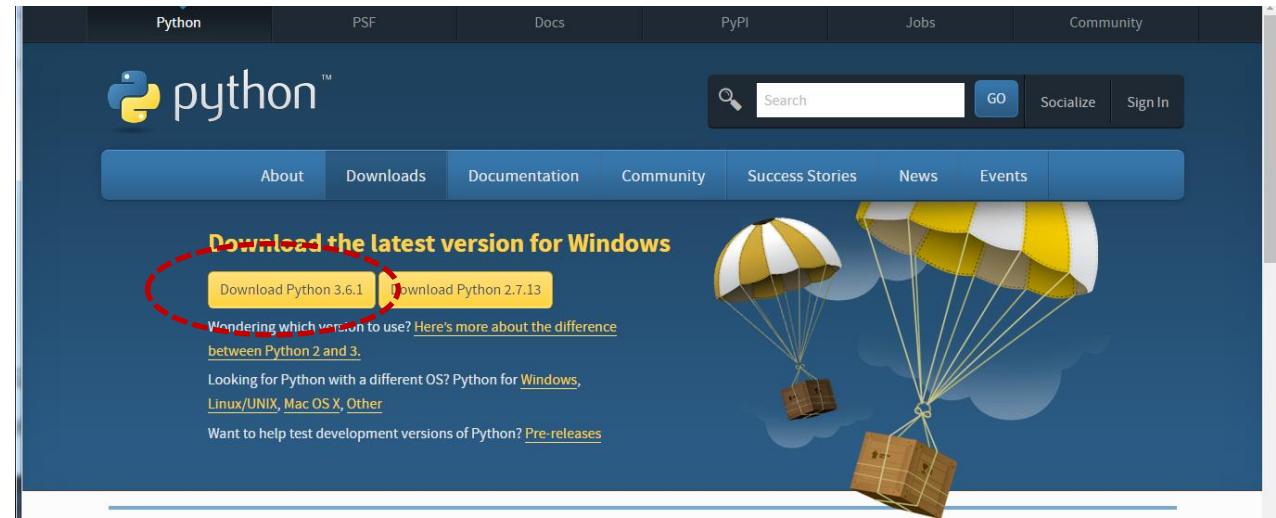
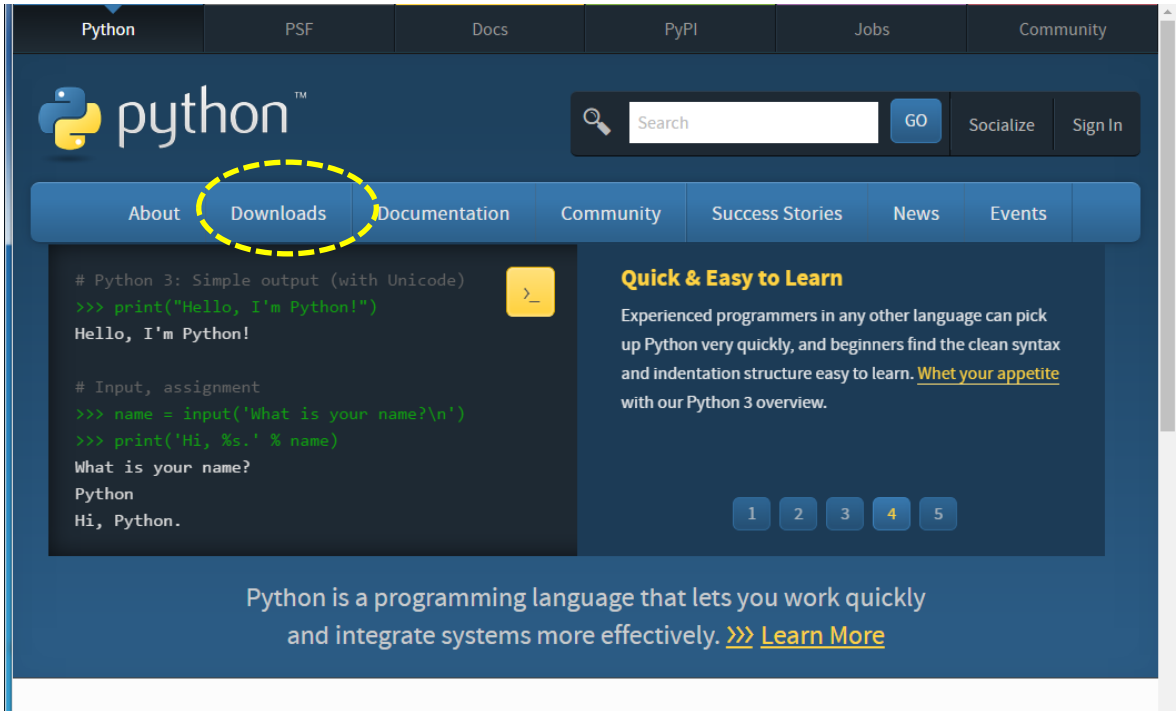
```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:38:48) [MSC v.1900 32 bit  
(Intel)] on win32  
Type "copyright", "credits" or "license()" for more information.  
>>> a=1  
>>> b=2  
>>> a+b  
3  
>>>
```

The status bar at the bottom right indicates "Ln: 7 Col: 4".

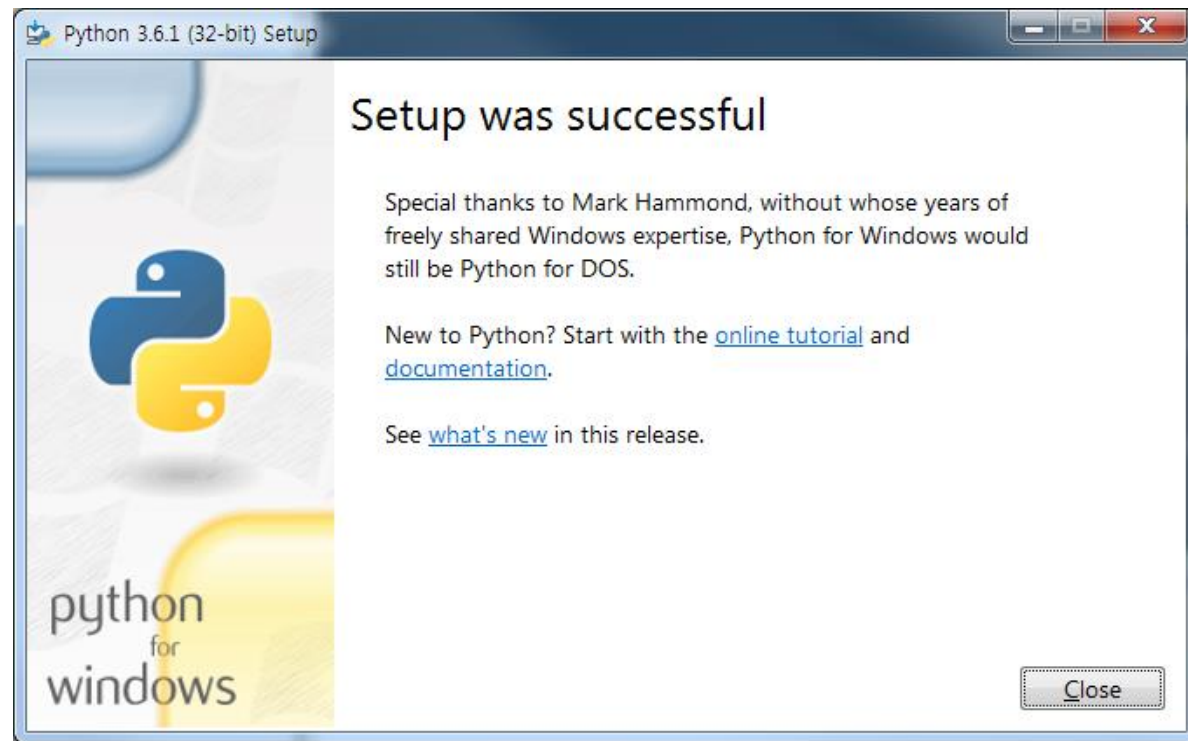
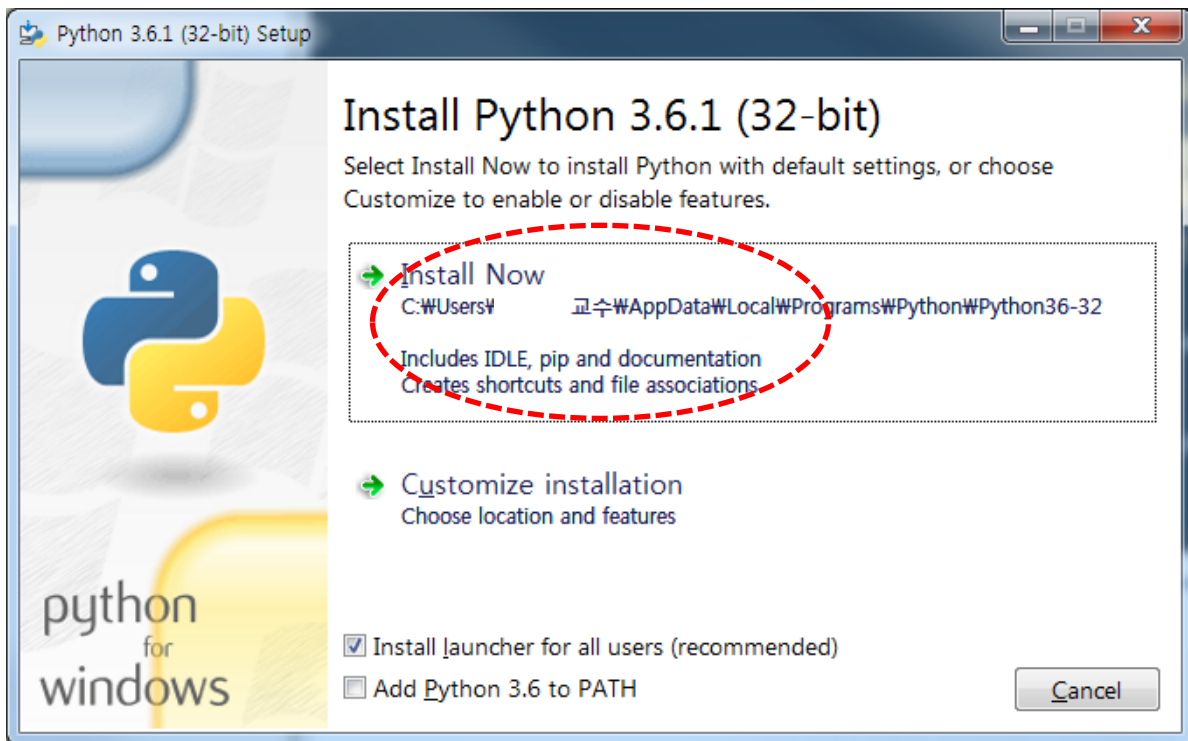


- 파이썬 다운로드

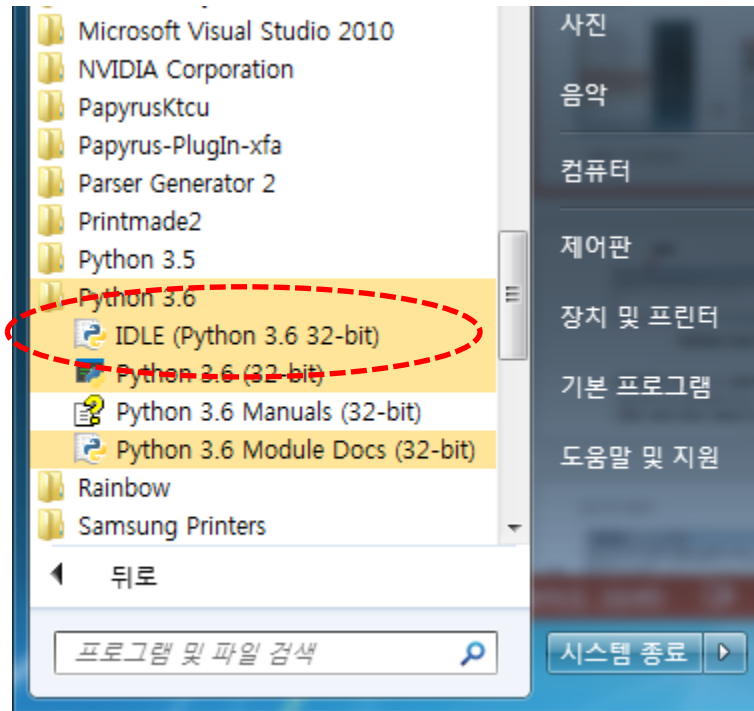
✓ www.python.org



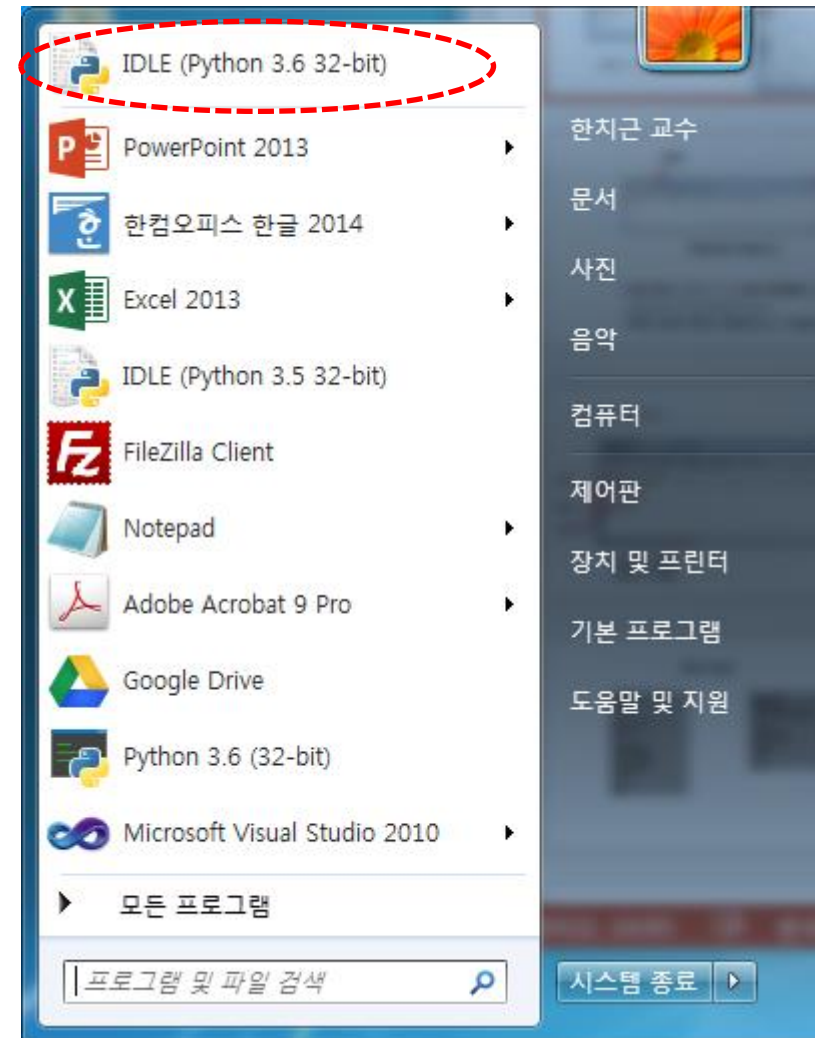
- 파이썬 설치



- 파이썬 시작



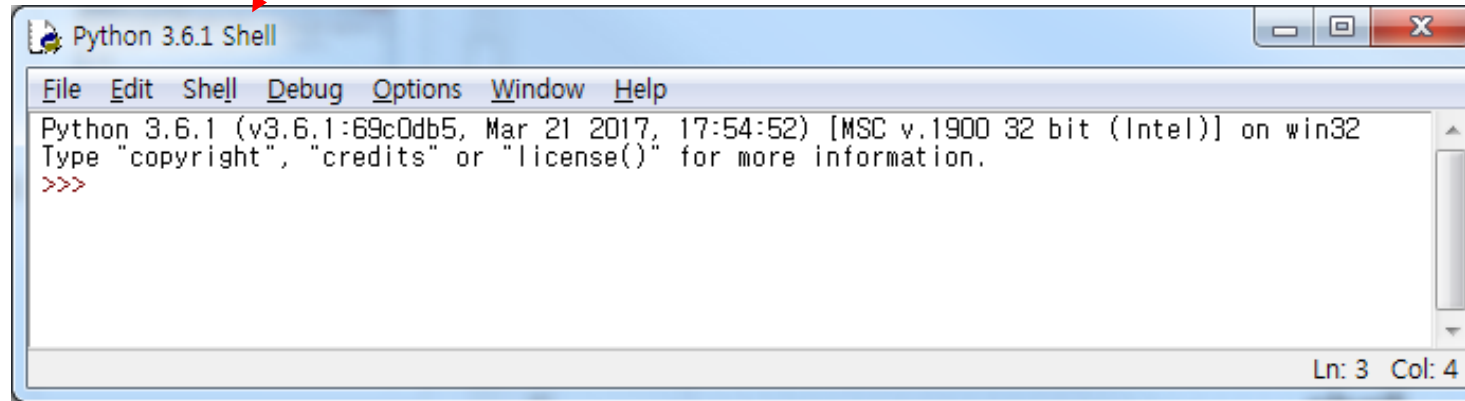
or



윈도우 시작 버튼에서



shell



IDLE shell window(창)

- shell: OS와 연결할 수 있는 user interface를 말한다.
- 명령어를 주면 값을 반환하게 된다.
- IDLE는 shell window와 editor window를 갖는다.



단순 프로그램 예

```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:38:48) [MSC v.1900 32 bit
(Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> a=1
>>> b=2
>>> a+b
3
>>>
```

입력

출력

프롬프트

Ln: 7 Col: 4

a=1을 친 후 enter
b=2를 친 후 enter
a+b를 친 후 enter



shell window

```
>>> a=3
>>> a
3
>>> a=a*a
>>> a
9
>>> b=2
>>> b=b*b
>>> b=b*b
>>> b
16
>>>
```

```
>>> a=2
>>> b=3
>>> a-b*a+b/a/3
-3.5
>>> c=5
>>> d=c*2
>>> d
10
>>>
```



shell window

```
>>> 3**5
243
>>> 2**4
16
>>> 2/4
0.5
>>> int(2/4)
0
>>> '5'
'5'
>>> 5
5
>>>
```

문자 5



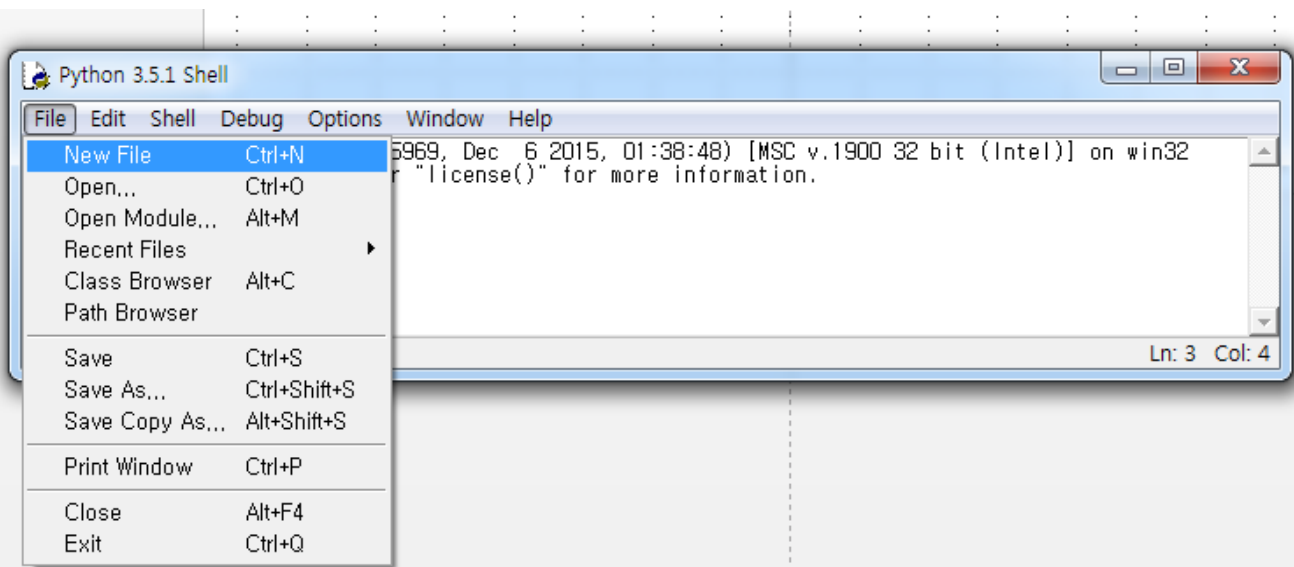
숫자 5



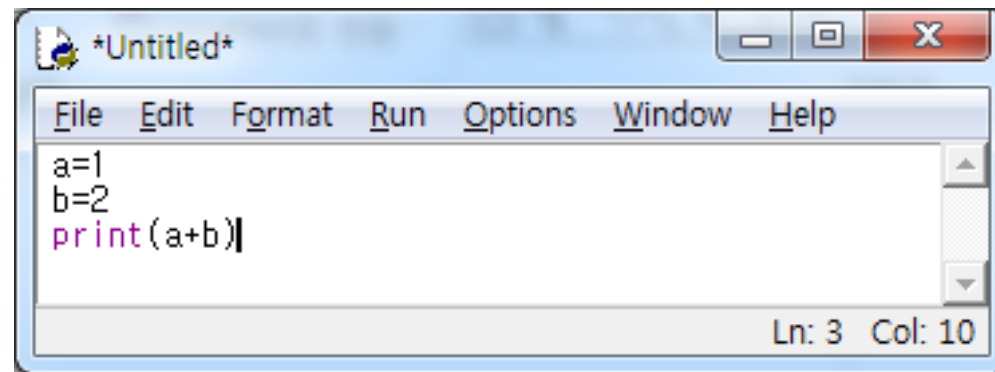
```
>>> 'abc'
'abc'
>>> "abc"
'abc'
>>> 'abc'+'def'
'abcdef'
>>> 'abc'+'d'
'abcd'
>>> abc
Traceback (most recent call last):
  File "<pyshell#53>", line 1, in <module>
    abc
NameError: name 'abc' is not defined
>>>
```



긴 프로그램을 매번 입력할 수 없으므로 파일을 생성해서 저장한다.



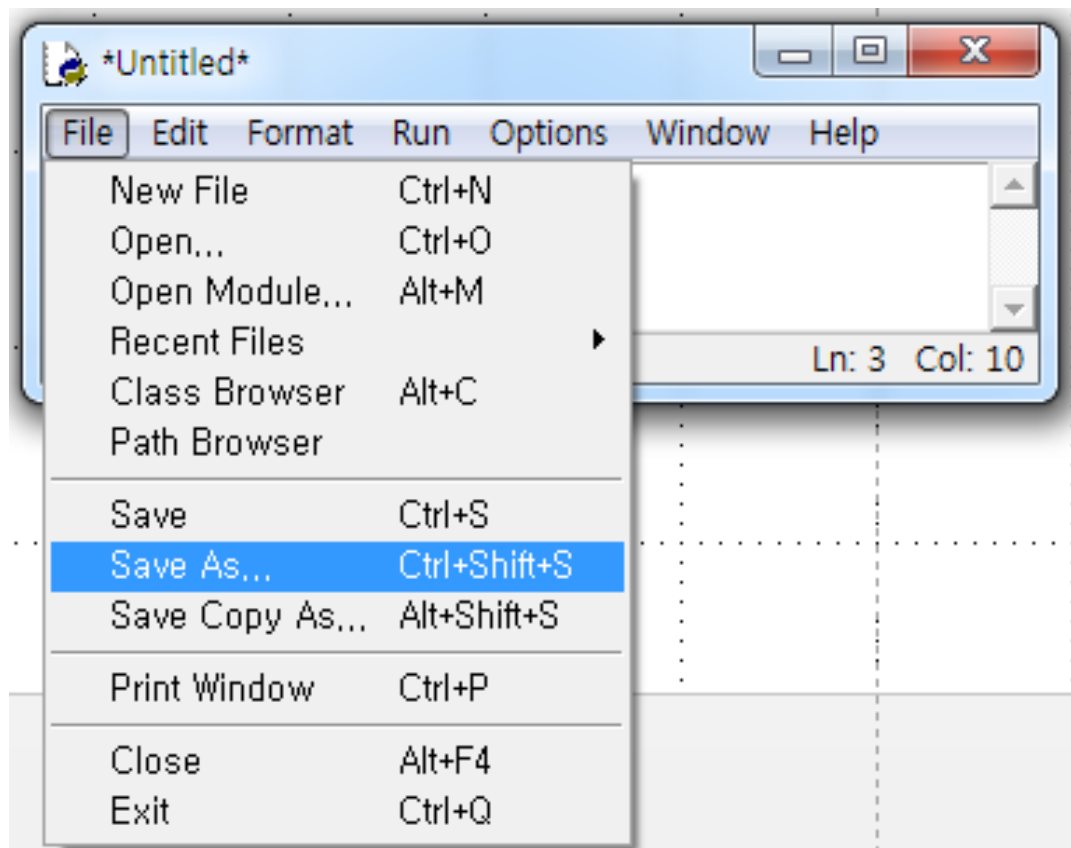
New File



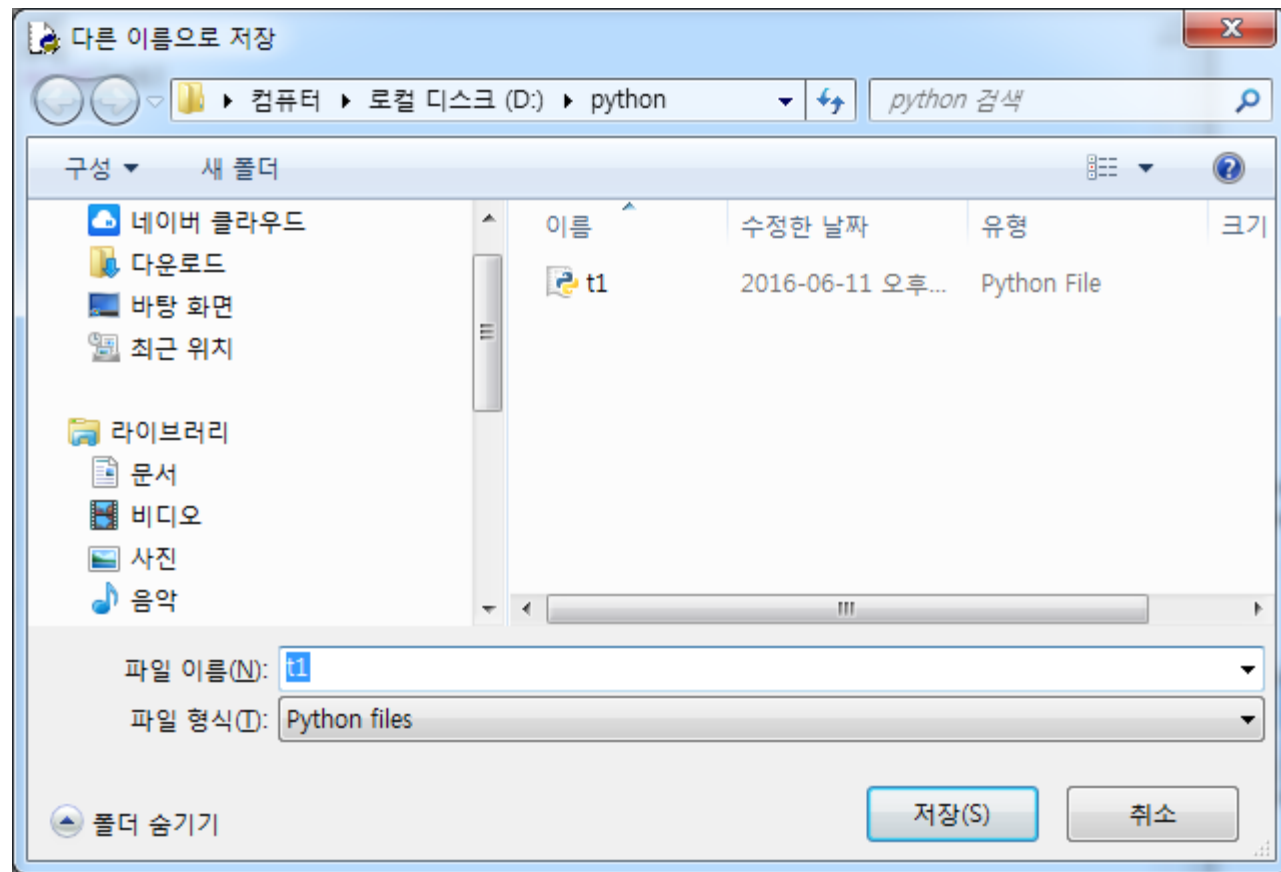
편집기 창(editor window)



t1.py로 파일 저장



Save As...

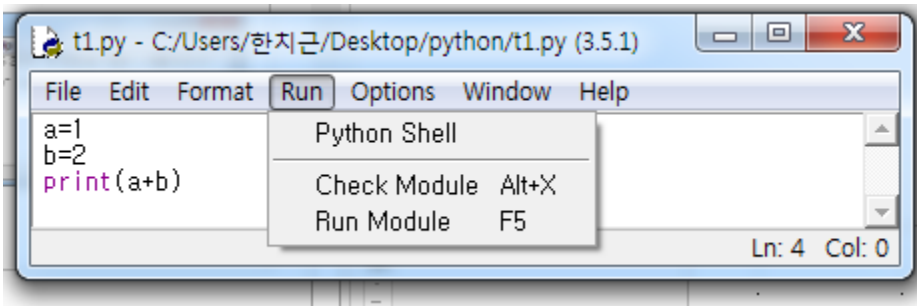


t1

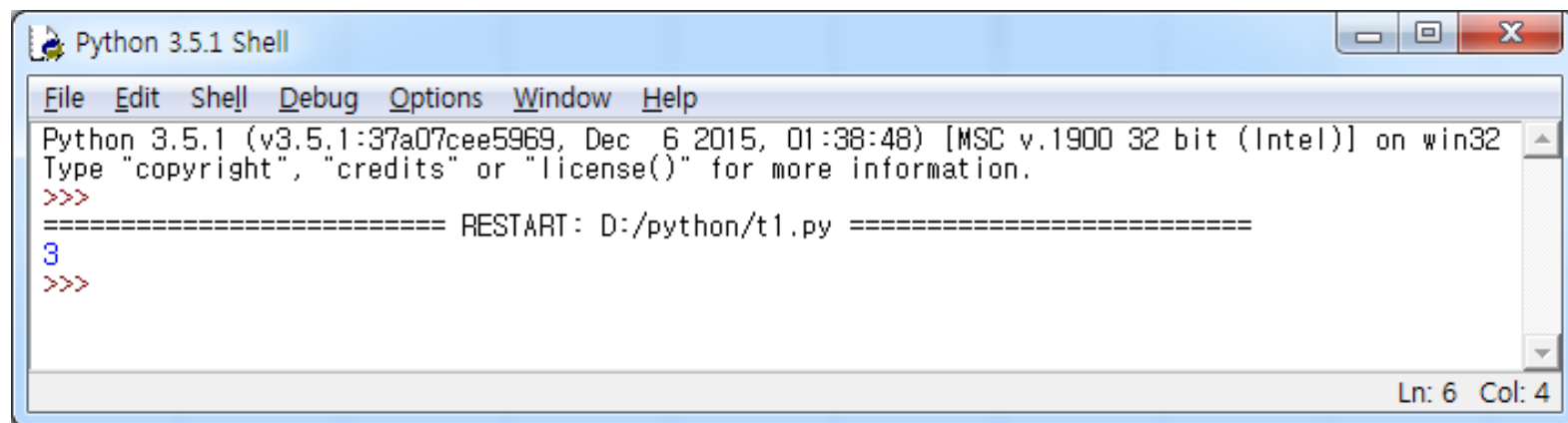


프로그램의 수행

- Run – Run Module
- F5



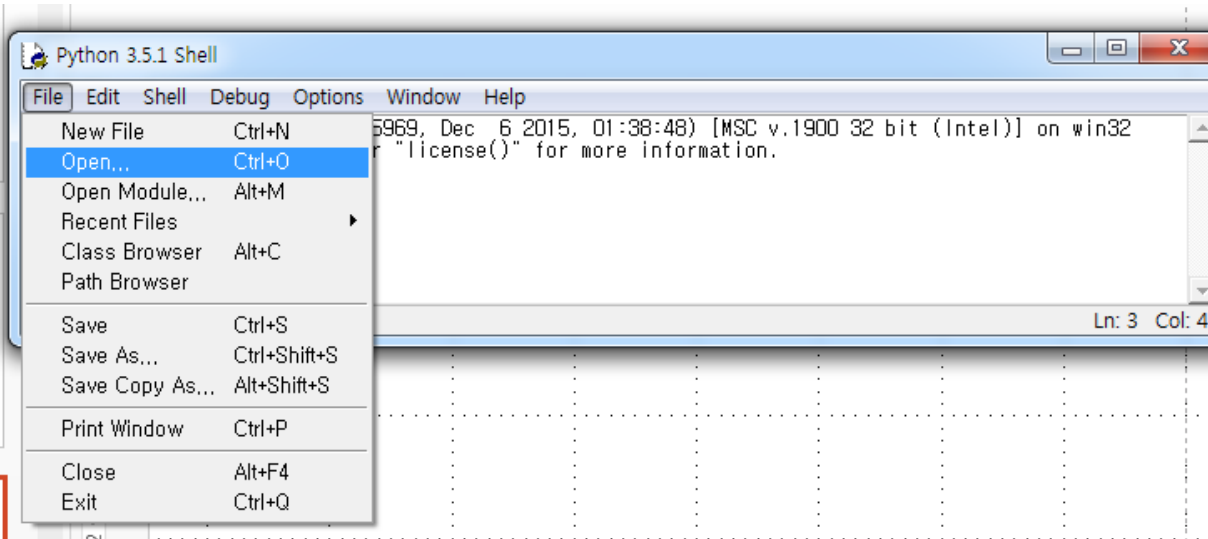
editor 창



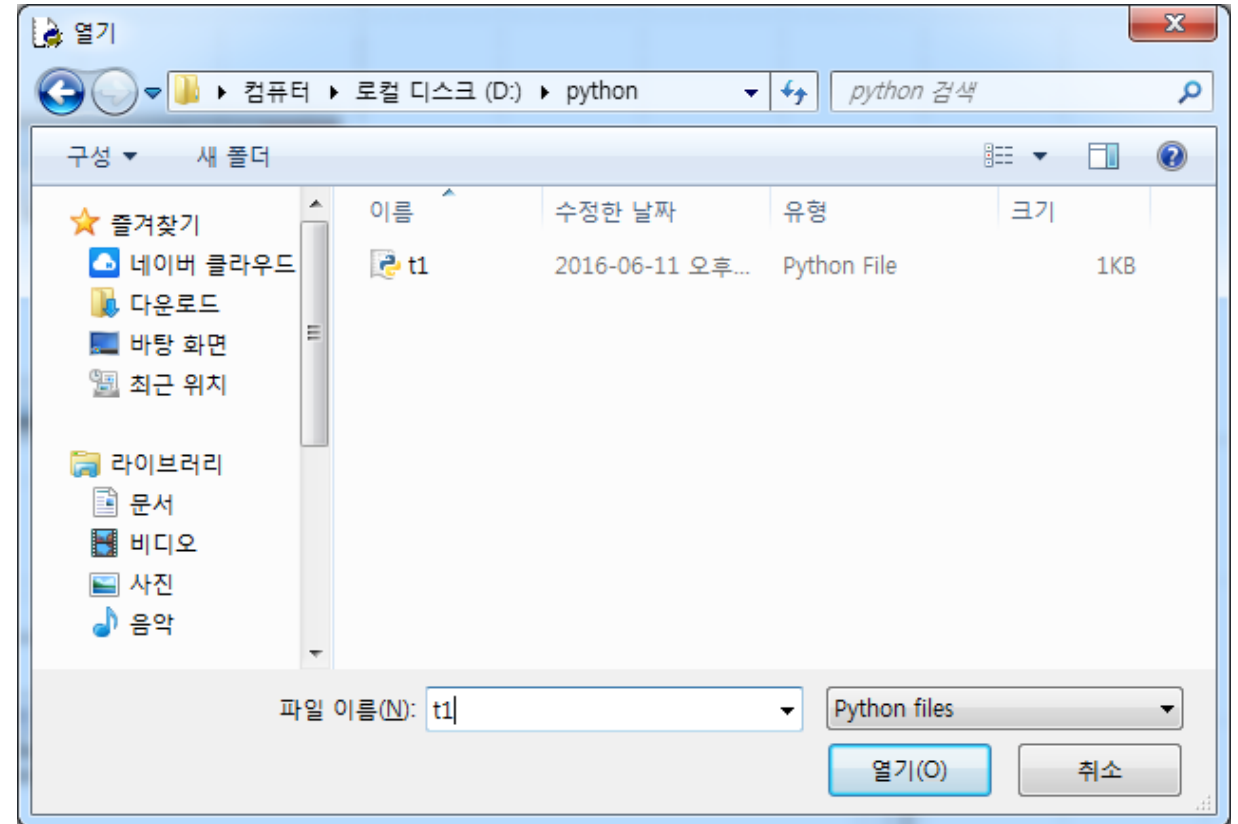
shell 창



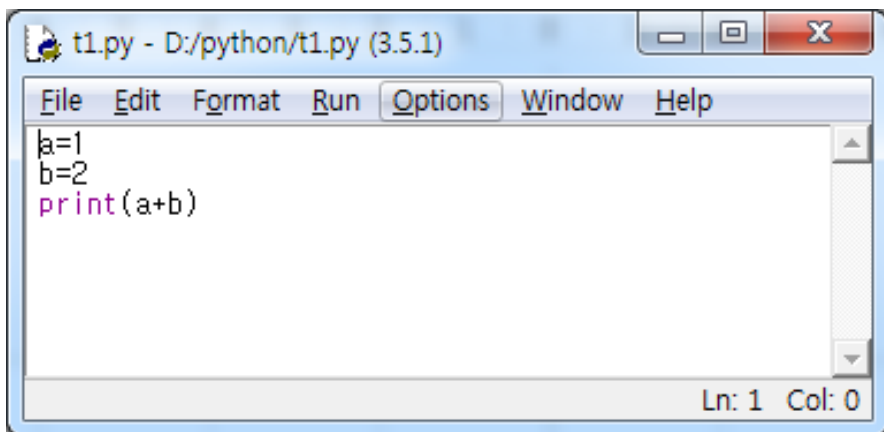
생성된 파일 t1.py를 연다



Open...

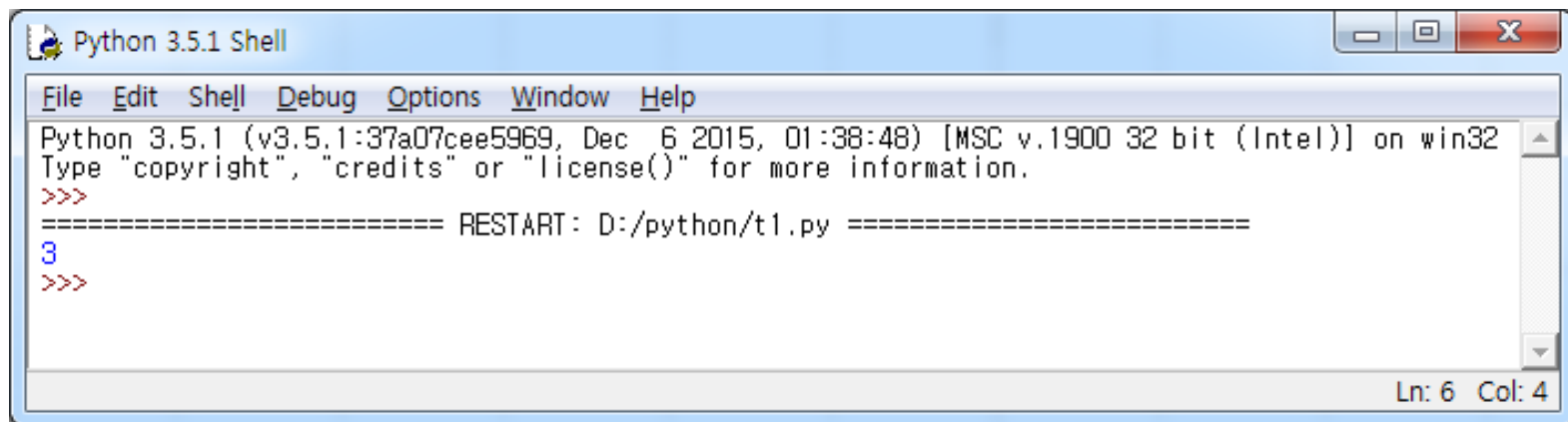


t1.py를 불러 온 후, F5를 눌러 프로그램을 수행한다.



```
a=1
b=2
print(a+b)
```

Ln: 1 Col: 0



```
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:38:48) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/python/t1.py =====
3
>>>
```

Ln: 6 Col: 4



[연습문제]

직사각형의 면적 구하기

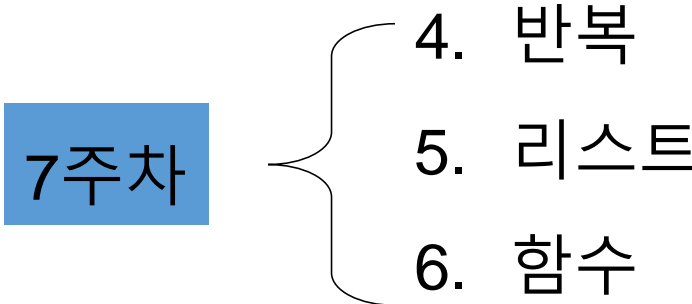
- 가로, 세로를 각각 a , b 라고 하고, 면적을 구해 본다.
- a 와 b 에 값을 부여하여 구한다.
- 파일로 저장한 후 다시 열어서 파일을 확인해 본다.
- 파일을 수행해 본다.



Python 공부 내용

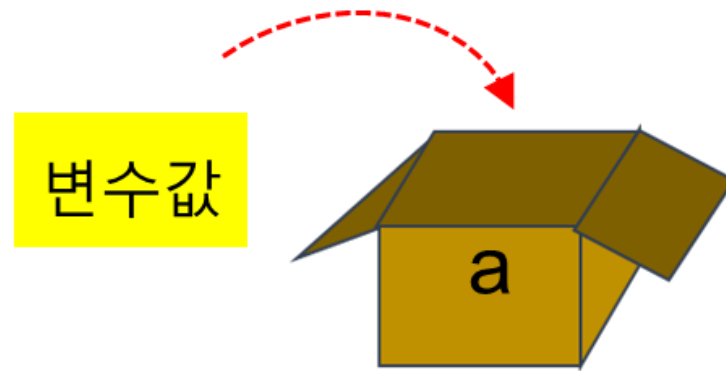
1. 변수에 값 저장
2. 입력, 출력
3. 조건부 수행

7주차

- 
4. 반복
 5. 리스트
 6. 함수



1. 변수에 값 저장



`a=53`

`word="school"`

`cost=100`

`total=sum+diff`

변수명 = 저장할 값 또는 변수명



프로그램 내부에서 변수에 값 저장

```
a=1
b=2.3
c="school"
d='day'
e=[1, "bus", 5.9]
f=a
print(a)
print(b)
print(c)
print(d)
print(e)
print(f)
```

```
1
2.3
school
day
[1, 'bus', 5.9]
1
>>>
```

✓ 변수가 갖고 있는 값은 연산에 사용 가능

```
a=1
b=2.3
print(a+1)
print(a+b)
```

```
2
3.3
>>>
```



- identifier 구성 방법

- ✓ 변수(variable)이름 이나 함수(function)이름을 identifier라고 한다.
- ✓ 알파벳 a-z, A-Z, _ (underbar, underscore)로 시작한다.
- ✓ 이후에는 복수개의 알파벳 a-z, A-Z, _, 숫자(0-9)가 올 수 있다.
- ✓ 대소문자를 구분한다. abc ≠ ABC
- ✓ 특별문자 @,\$,% 등은 사용할 수 없다.



예약어(reserved words)

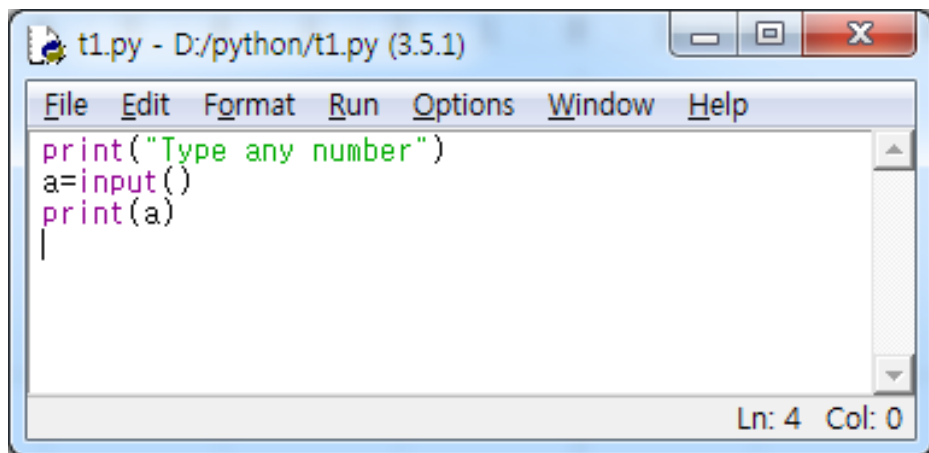
- 식별자로 사용할 수 없다.

and	del	from	not	while
as	elif	global	or	with
assert	else	if	pass	yield
break	except	import	print	
class	exec	in	raise	
continue	finally	is	return	
def	for	lambda	try	



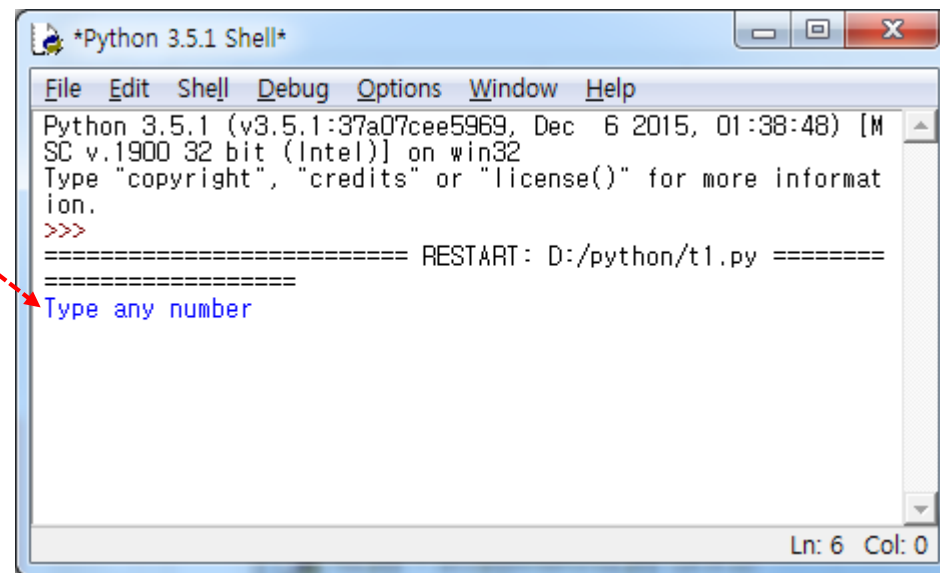
2. 입력, 출력

- ✓ input, output 사용
- ✓ input으로 받은 값은 문자로 바로 연산에 사용할 수 없다.



```
File Edit Format Run Options Window Help
print("Type any number")
a=input()
print(a)
Ln: 4 Col: 0
```

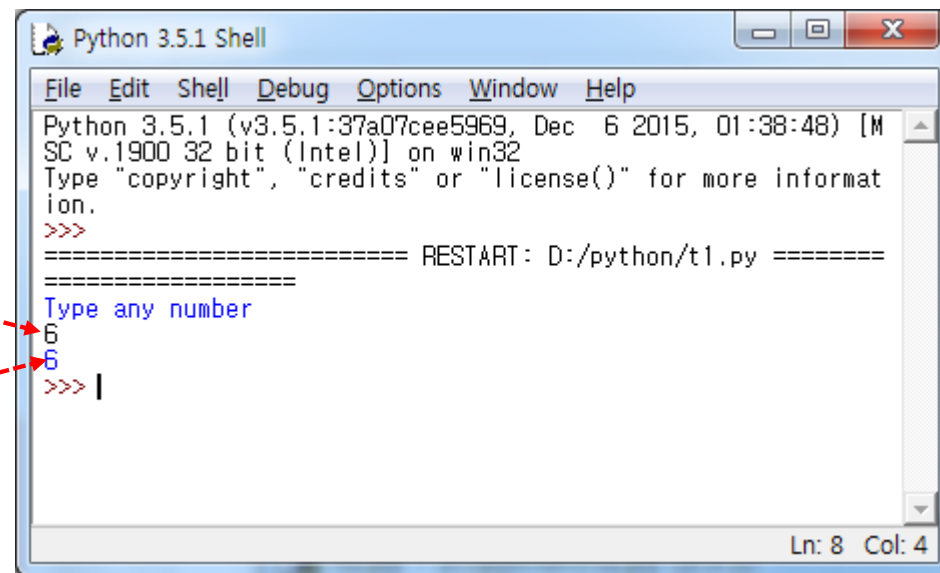
출력



```
*Python 3.5.1 Shell*
File Edit Shell Debug Options Window Help
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:38:48) [M
SC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informat
ion.
>>>
===== RESTART: D:/python/t1.py =====
Type any number
Ln: 6 Col: 0
```

입력

출력



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
Python 3.5.1 (v3.5.1:37a07cee5969, Dec 6 2015, 01:38:48) [M
SC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more informat
ion.
>>>
===== RESTART: D:/python/t1.py =====
Type any number
6
6
>>> |
Ln: 8 Col: 4
```



IDLE shell 창

```
>>> print("Test")
Test
>>> a=input()
xyz
>>> a
'xyz'
>>> a=input("Type any number ")
Type any number 5
>>> a
'5'
```

- Test를 출력한다.
- 사용자가 문자를 입력하기를 기다렸다가, 입력이 되면 변수 a에 저장한다.
- a값을 출력한다.
- 화면에 Type any number를 출력하고, 사용자가 문자를 입력하기를 기다렸다가, 입력이 되면 변수 a에 저장한다.

사용자가 입력한 문자

5라는 문자



즉, 숫자 입력을 하더라도 내부에서는 문자로 인식된다.

편집기 창

```
a=input("Type any number ")  
print(a)
```

출력

shell 창

입력

5라는 문자

```
Type any number 5  
5  
>>>
```

```
a=input("Type any number ")  
print(a+1)
```

```
Type any number 5  
Traceback (most recent call last):  
  File "D:\python\t1.py", line 2, in <module>  
    print(a+1)  
TypeError: Can't convert 'int' object to str implicitly  
>>>
```

```
a=input("Type any number ")  
print(int(a)+1)
```

5라는 문자를 숫자 5로 변환

```
Type any number 5  
6  
>>>
```

- 소수점있는 숫자로 변환할 때는 float() 사용



문자열의 연결(concatenation)

```
a=input("Type string1: ")
b=input("Type string2: ")
print(a+b)
print(a+"world")
```



```
Type string1: hello
Type string2: car
hellocar
helloworld }
>>>
```

입력

출력

+ 연산은 문자열에 대해서는 연결하는 작업을 수행한다.



- 소수점있는 숫자로 변환할 때는 float 사용

```
a=input("Type any number ")
print(float(a)+1)
```

```
Type any number 3.4
4.4
>>>
```

- 소수점있는 두 숫자를 입력 받아 덧셈을 출력하기

```
a=input("실수 입력 = ")
b=input("실수 입력 = ")
a= float(a)
b= float(b)
print("합 = ", a+b)
```

```
실수 입력 = 1.2
실수 입력 = 3.4
합 = 4.6
>>>
```

```
a, b = input("실수 입력 = "), input("실수 입력 = ")
a, b = float(a), float(b)
print("합 = ", a+b)
```



[연습문제]

- 직사각형의 두 변의 길이 a, b 를 입력을 받는다.
- 직사각형의 변의 합과
- 직사각형의 면적을 계산한 후 출력한다.

[연습문제]

- 세 개의 문자열을 입력으로 받는다.
- 문자열 순서를 다양하게 조합하여 출력한다.
- [예] 세 개의 문자열: car, book, moon
carbookmoon, mooncarbook, carcarcar를 출력한다.

3. 조건부 수행

✓ if

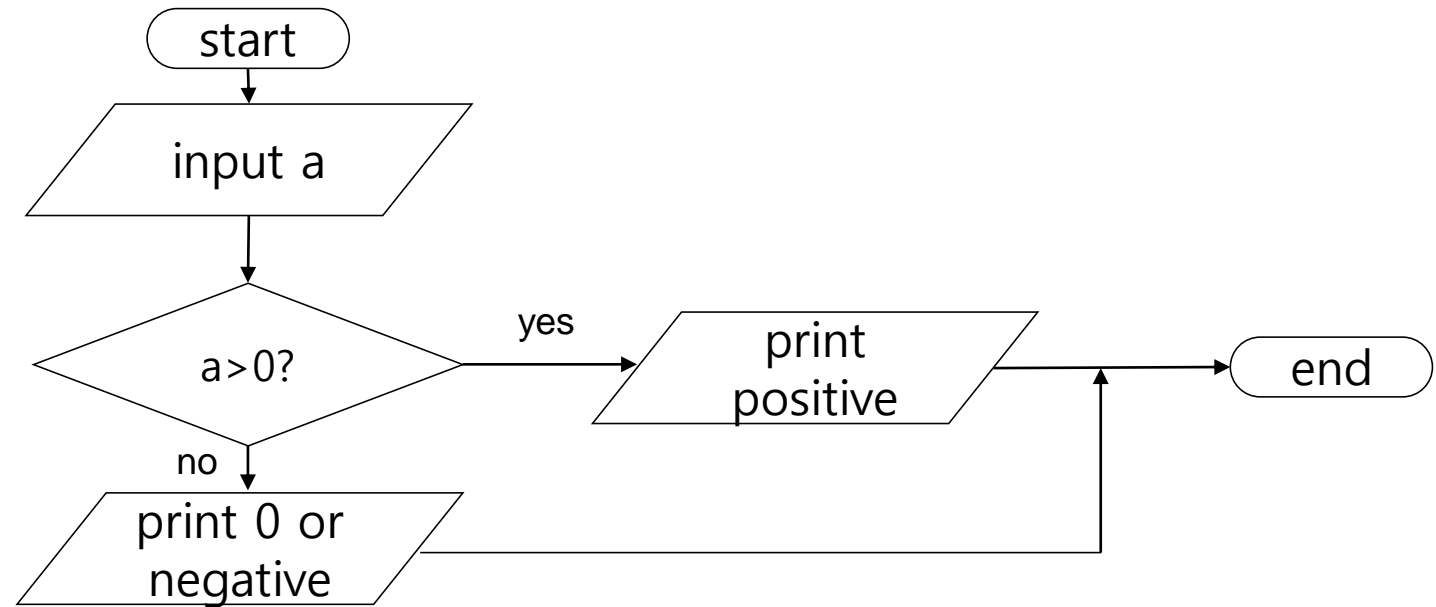
✓ else

✓ elif

```
a = 5
if(a>0):
    print("positive")
else:
    print("0 or negative")
```

if (a>0):

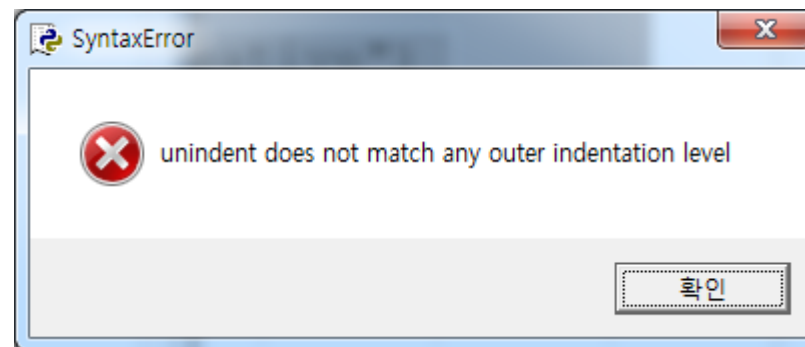
else:



들여쓰기가 달라질 때

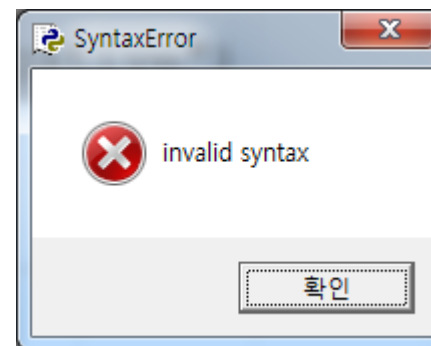
```
a = 5
if(a>0) :
    print("positive")
else:
    print("0 or negative")
```

- if 와 else가 서로 짝을 이루지 않는다.



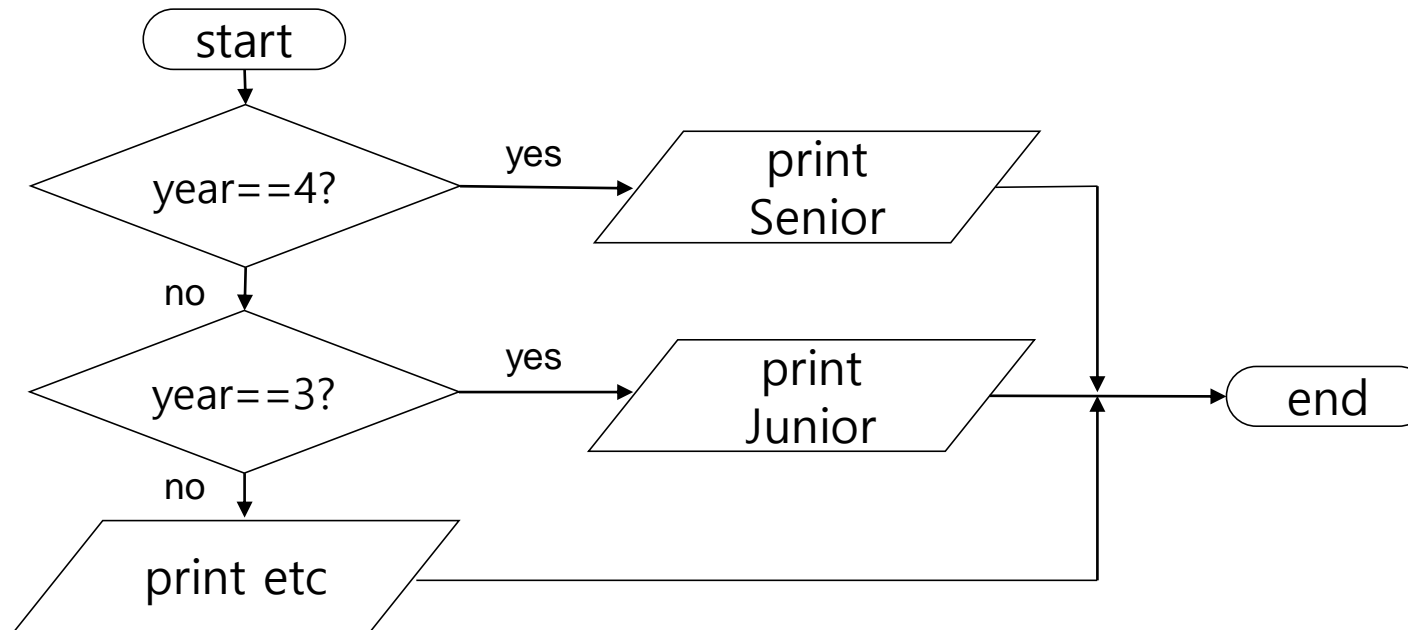
```
a = 5
if(a>0) :
    print("positive")
    else:
    print("0 or negative")
```

- if 내에 else가 if 없이 사용되었다.



```
year = 3
if year == 4:
    print("Senior")
elif year == 3:
    print("Junior")
else:
    print("etc")
```

```
Junior
>>>
```



[연습문제]

입력으로 정수를 받아 0이면 zero라고 출력하고,
양수이면 positive, 음수이면 negative라고 출력한다.

✓ 다양한 조건식을 만들 수 있다.

```
year = 3
course = "ce"
if year == 4 and course=="ce":
    print("computer Senior")
elif year == 3 or course=="ce":
    print("Junior or computer")
else:
    print("etc")
```

not A

A	not A
F	T
T	F

T: True
F: False

```
Junior or computer
>>>
```

A or B
A and B

A	B	A and B	A or B
F	F	F	F
F	T	F	T
T	F	F	T
T	T	T	T



중첩된 조건문

```
kind = "bus"
size = 2000

if kind == "bus":
    if size > 3000:
        tax = 100
    else:
        tax = 50
else:
    if size > 3000:
        tax = 200
    else:
        tax = 150

print("Tax = ", tax)
```

```
Tax = 50
>>>
```



연산자 우선순위

우선순위(and) > 우선순위(or)

```
if a>0 or b<10 and c==5 or d>7:
```

A.

```
if (a>0 or b<10) and (c==5 or d>7):
```

B.

```
if a>0 or (b<10 and c==5) or d>7 :
```



파이썬의 연산자

- arithmetic operators: + - * / % // **
- comparison operators: > < == != >= <=
- logical operators: and or not
- bitwise operators: & | ~ ^ >> <<
- assignment operators: = += -= *= /= %= //= **= &= !=
^= >>= <<=
- special operators: is 'is not'
- membership operators: in 'not in'



True와 False는 예약어

```
a=True

if a == True:
    print("abc")
else:
    print("xyz")
```

```
abc
>>>
```

```
a=True

if a :
    print ("abc")
else:
    print ("xyz")
```

```
abc
>>>
```



TRUE는 예약어가 아니다.

```
a=TRUE

if a :
    print ("abc")
else:
    print("xyz")
```

```
NameError: name 'TRUE' is not defined
>>>
```



```
a=True
b=False

if a and b:
    print("a와 b가 모두 진실")
elif (a and not b) or (not a and b):
    print("a 또는 b는 진실")
else:
    print("a와 b 모두 거짓")
```

```
a 또는 b는 진실
>>>
```

```
a=True
b=False

if a & b:
    print("a와 b가 모두 진실")
elif (a & ~b) | (~a & b):
    print("a 또는 b는 진실")
else:
    print("a와 b 모두 거짓")
```

```
a 또는 b는 진실
>>>
```



[연습문제]

- 입력으로 국가코드 (K, E, U, J, C)가 주어진다.
- 국가코드가 K이면 “내국인”라고 출력하고, E 또는 U이면 “서양인”이라고 출력한다.
- 이 외의 국가코드이면 (J 또는 C) “동양인”이라고 출력한다.
- 그 외의 국가코드는 없다고 가정한다.

[연습문제]

- 입력으로 season(summer, winter의 값을 가진다)과 temperature를 받는다.
- 먼저 season을 구분하고, summer일 때 temperature가 30도 이상이면, hot이라고 출력하고, 20도부터 29도 이하일 때는 warm이라고 출력한다. 19도 이하이면 cool이라고 출력한다.
- winter일 때는 temperature가 10도 이상이면 warm이라고 출력하고, 9도 이하이면 cold라고 출력한다.



[연습문제]

1. 자연수 한 개를 입력 받아, 만일 짝수이면 “짝수”라고 출력하고, 만일 홀수이면 “홀수” 라고 출력한다.
2. 자연수 한 개를 입력 받아, 만일 5의 배수이고, 4의 배수이면 “20의 배수”라고 출력한다. 5의 배수만 되면, “5의 배수”, 4의 배수만 되면, “4의 배수” 라고 출력한다.



6주차 강의 요약

- 파이썬 소개 및 설치
- 변수에 값 저장
- 입력
- 출력
- 조건부 수행
 - ✓ if, elif, else



6주차 끝

수고하셨습니다.