

### 3. 함수: 같은 코딩을 두 번 하지 맙시다.

---

순천향대학교 컴퓨터공학과  
이 상 정

프로그래밍 기초

### 학습 내용

---

- 프로그램이 커질 수록 코드는 복잡
  - 복잡한 코드는 읽기도 어렵고 유지보수도 어려움
  - 함수(function)를 사용하여 복잡함을 관리
- 함수는 코드 덩어리로 프로그램 안에서 필요할 때 사용
- 함수는 공통된 행위를 따로 분리
  - 코드를 더 읽기 쉽고 관리하기 좋게 만듦



## 스타버즈 코드에 옵션 추가

- ❑ 매장에서 원두가 떨어지는 경우에 현재 가격에 관계 없이 **즉시 주문하는 옵션**을 추가
- ❑ 새 프로그램은 두 가지 옵션 제공
  - 첫 번째 옵션은 커피 원두 가격이 떨어질 때까지 기다림
  - 두 번째 옵션은 긴급 주문으로 현재 가격을 화면에 바로 보여줌
- ❑ 두 옵션 모두 **현재의 가격 정보**가 필요함
  - 현재 가격 정보를 얻는 코드를 두 번 작성하지 **함수**로 작성
  - 필요할 때 마다 함수를 재사용(**호출**)
- ❑ 파이썬 함수는 **def** 키워드를 사용

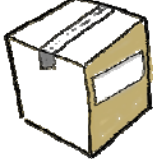


## 기존 스타버즈 코드

```
import urllib.request
import time

price = 99.99
while price > 4.74:
    time.sleep(900)
    page = urllib.request.urlopen(" http://cs.sch.ac.kr/prices-loyalty.py")
    text = page.read().decode("utf8")
    where = text.find(">$")
    start_of_price = where + 2
    end_of_price = start_of_price + 4
    price = float(text[start_of_price: end_of_price])
print("Buy!")
```

가격 정보 코드,  
재사용되는 코드



# 원두 가격 출력 함수 예 1 [실습 1]

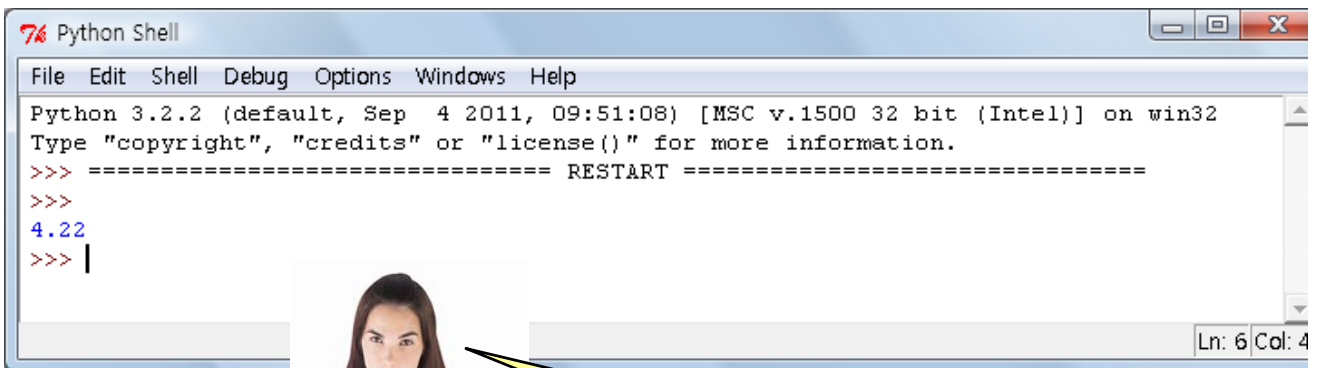
```
import urllib.request

def get_price():                # 함수 정의
    page = urllib.request.urlopen(" http://cs.sch.ac.kr/prices-loyalty.py")
    text = page.read().decode("utf8")
    where = text.find(">$")
    start_of_price = where + 2
    end_of_price = start_of_price + 4
    print(text[start_of_price: end_of_price])

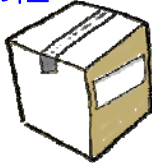
get_price()                    # 함수 호출
```



# 시험주행



함수에서 가격정보를 출력 하지 않고 가져오는 방법은?



## 원두 가격 출력 함수 예 2 [실습 2]

```
import urllib.request

def get_price():          # 함수 정의
    page = urllib.request.urlopen(" http://cs.sch.ac.kr/prices-
loyalty.py")
    text = page.read().decode("utf8")
    where = text.find(">$")
    start_of_price = where + 2
    end_of_price = start_of_price + 4
    return text[start_of_price: end_of_price]      # 가격을 반환

price = get_price()      # 함수 호출
print(price)
```

### 스타버즈 코드: 긴급 주문 옵션 [실습 3]

```
import urllib.request
import time

def get_price():          # 함수 정의
    page = urllib.request.urlopen("http://cs.sch.ac.kr/prices-loyalty.py ")
    text = page.read().decode("utf8")
    where = text.find(">$")
    start_of_price = where + 2
    end_of_price = start_of_price + 4
    return float(text[start_of_price: end_of_price])      # 가격을 숫자로 반환

price_now = input("Do you want to see the price now (Y/N)? ")
if price_now == "Y":
    print(get_price())      # 가격을 호출하여 출력
else:
    price = 99.99
    while price > 4.74:
        time.sleep(900)
        price = get_price()      # 함수 호출
    print("Buy!")
```



# 시험주행

```

Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.2 (default, Sep 4 2011, 09:51:08) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Do you want to see the price now (Y/N)? Y
4.79
>>> ===== RESTART =====
>>>
Do you want to see the price now (Y/N)? N
Buy!
>>> |
Ln: 11 Col: 4

```

# 스타버즈 CEO 추가 요구

- CEO는 가격이 휴대폰으로 전송되기를 원함
  - 외근 중일 때 가격 정보를 트위터 계정으로 받기를 원함



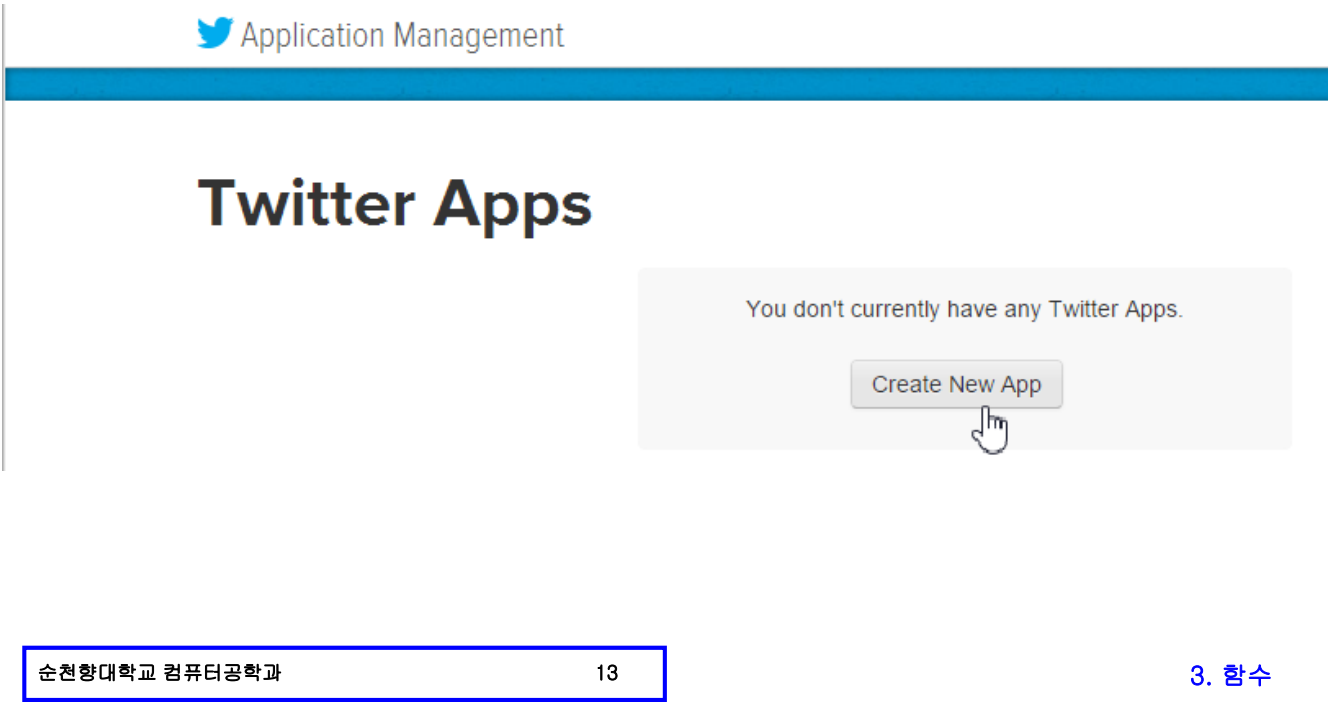
## 트위터 전송을 위한 작업

1. 트위터 계정을 만듦
  - <http://twitter.com>
2. 트위터에 애플리케이션을 등록
  - <http://dev.twitter.com/apps>
  - 소비자 키와 비밀번호, 접근 키와 비밀번호 복사
    - OAuth 표준 애플리케이션 인증 방식
3. 트위터 라이브러리 설치
4. 스타버즈 소스 코드 수정
  - 등록된 애플리케이션 소비자 키와 비밀번호, 접근 키와 비밀번호를 삽입
  - 트위터 라이브러리 사용하여 트위터에 메시지 전송

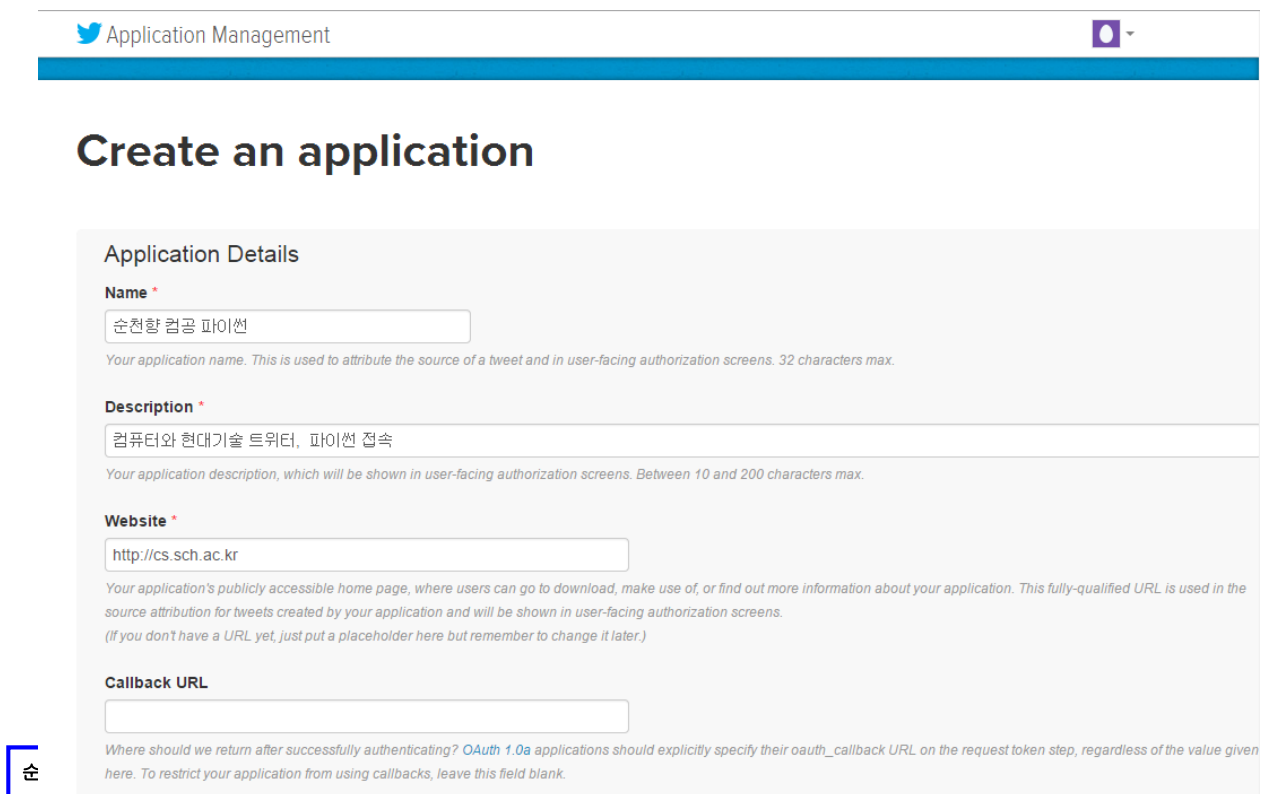
## 트위터 애플리케이션 등록 - 로그인

The screenshot shows a web browser window with the URL <https://apps.twitter.com>. The page title is "Twitter Apps" and the main heading is "Twitter Apps". Below the heading, there is a message: "Please [sign in](#) with your Twitter Account to create and maintain Twitter Apps." A mouse cursor is pointing at the "sign in" link. In the bottom right corner, there is a login form titled "트위터에 로그인하기" (Log in to Twitter). The form contains two input fields: one for the username (containing "myaccount") and one for the password (containing "\*\*\*\*\*"). Below the password field is a blue "로그인" (Log in) button. To the right of the button is a checkbox labeled "ID 저장 · 비밀번호 찾기" (Save ID · Find password), which is checked. A mouse cursor is pointing at the "로그인" button.

# 애플리케이션 등록 - 새로운 애플리케이션 생성



# 애플리케이션 생성 (1)



# 애플리케이션 생성 (2)

**Callback URL**

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth\_callback URL on the request token step, regardless of the application type. To restrict your application from using callbacks, leave this field blank.

---

**Developer Agreement**

Last Update: October 22, 2014.

This Twitter Developer Agreement ("**Agreement**") is made between you (either an individual or an entity, referred to herein as "**you**") and Twitter, Inc., on behalf of itself and its worldwide affiliates (collectively, "**Twitter**") and governs your access to and use of the Licensed Material (as defined below).

PLEASE READ THE TERMS AND CONDITIONS OF THIS AGREEMENT CAREFULLY, INCLUDING WITHOUT LIMITATION ANY LINKED TERMS AND CONDITIONS APPEARING OR REFERENCED BELOW, WHICH ARE HEREBY MADE PART OF THIS LICENSE AGREEMENT. BY USING THE LICENSED MATERIAL, YOU ARE AGREEING THAT YOU HAVE READ, AND THAT YOU AGREE TO COMPLY WITH AND TO BE BOUND BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND ALL APPLICABLE LAWS AND REGULATIONS IN THEIR ENTIRETY WITHOUT LIMITATION OR QUALIFICATION. IF YOU DO NOT AGREE TO BE BOUND BY THIS AGREEMENT, THEN YOU MAY NOT ACCESS OR OTHERWISE USE THE LICENSED MATERIAL. THIS AGREEMENT IS EFFECTIVE AS OF THE FIRST DATE THAT YOU USE THE LICENSED MATERIAL ("**EFFECTIVE DATE**").

IF YOU ARE AN INDIVIDUAL REPRESENTING AN ENTITY, YOU ACKNOWLEDGE THAT YOU HAVE THE APPROPRIATE AUTHORITY TO ACCEPT THIS AGREEMENT ON BEHALF OF SUCH ENTITY. YOU MAY NOT USE THE LICENSED MATERIAL AND MAY NOT ACCEPT THIS AGREEMENT IF YOU ARE NOT OF LEGAL AGE TO FORM A BINDING CONTRACT WITH

Yes, I agree

순천향 [Create your Twitter application](#)

3. 함수

# 애플리케이션 생성 (3)

Application Management

Your application has been created. Please take a moment to review and adjust your application's settings.

## 순천향 컴공 파이썬

Details Settings Keys and Access Tokens Permissions

컴퓨터와 현대기술 트위터, 파이썬 접속  
http://cs.sch.ac.kr

**Organization**  
Information about the organization or company associated with your application. This information is optional.

Organization	None
Organization website	None

**Application Settings**  
Your application's Consumer Key and Secret are used to [authenticate](#) requests to the Twitter Platform.

Access level	Read-only (modify app permissions)
Consumer Key (API Key)	6F3eY9ga1LrQF08hFI3obm0XM (manage keys and access tokens)
Callback URL	None

순천향대학교 컴퓨터공학과

3. 함수



# 애플리케이션 퍼미션 설정

## 순천향 컴공 파이썬

Details Settings Keys and Access Tokens Permissions

### Access

What type of access does your application need?

Read more about our Application Permission Model.

- Read only
- Read and Write
- Read, Write and Access direct messages

#### Note:

Changes to the application permission model will only reflect in access tokens obtained after the permission will need to re-negotiate existing access tokens to alter the permission level associated with each of yo

Update Settings



3. 함수



# 보충학습 - OAuth 인증 소개

## □ OAuth (Open standard for Authorization) 인증

- OAuth 인증은 소비자가 별도의 인증 (사용자 ID, 비밀번호 등) 없이 서비스 제공자의 보호된 자원에 접근을 허가하는 인증 방식

## □ 주요 용어

- 사용자 (user): 서비스 공급자와 소비자를 사용하는 계정을 가지고 있는 개인
- 소비자 (consumer): 사용자를 대신하여 서비스 공급자에 접근하는 웹사이트 또는 애플리케이션(프로그램)
- 서비스 제공자 (service provider): OAuth를 통해 접근을 허용하는 웹애플리케이션(SNS, 포털), 예) facebook, twitter, daum, naver 등
- 소비자 키 (consumer key): 서비스 제공자에게 소비자 자신임을 인증하기 위한 키
- 소비자 비번 (consumer secret): 소비자의 소비자 키 소유권만이 있는지 인증하기 위한 키
- 접근 토큰 (access Token): 소비자가 사용자의 서비스 제공자의 비밀번호를 사용하지 않고 사용자를 대신해서 접근하기 위해서 사용하는 값
- 접근 토큰 비번 (access token secret): 접근 토큰이 사용자의 것임을 인증하기 위한 값



# 애플리케이션 접근 토큰 생성 - 소비자 키 (3)

## 순천향 컴공 파이썬

Details Settings **Keys and Access Tokens** Permissions

### Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key)	TlcvC...GSXjR7...
Consumer Secret (API Secret)	ecWtLw6Gf4v...WCVyFtl5Q573sWXzNkLVY0Qsb4G91
Access Level	Read and write (modify app permissions)
Owner	
Owner ID	45...

# 애플리케이션 접근 토큰 생성 - 접근 키 (4)

## Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token.

Access Token	4566f...2-T...YZC1Cu824H...CHlgk0xl...DBa72FFOOytD
Access Token Secret	Vnhag1...HV9TKJZv8...Bn35usL3v95s...FsQDHqb7
Access Level	Read and write
Owner	
Owner ID	4566f...2

## 트위피(Tweepy) 라이브러리 설치

□ 트위피 라이브러리 모듈은 파이썬으로 트위터 접속을 지원하는 라이브러리

- PyPI (Python Package Index)를 사용하여 설치
  - PyPI는 파이썬 모듈을 제공하는 중앙 저장소(repository)  
<https://pypi.python.org/pypi>

C:W> python -m pip install tweepy

```
C:W>python -m pip install tweepy
Collecting tweepy
  Downloading tweepy-3.5.0-py2.py3-none-any.whl
Collecting requests-oauthlib>=0.4.1 (from tweepy)
  Downloading requests_oauthlib-0.8.0-py2.py3-none-any.whl
Collecting six>=1.7.3 (from tweepy)
  Downloading six-1.10.0-py2.py3-none-any.whl
Collecting requests>=2.4.3 (from tweepy)
  Downloading requests-2.13.0-py2.py3-none-any.whl (584kB)
 100% |#####|: 593kB 867kB/s
Collecting oauthlib>=0.6.2 (from requests-oauthlib>=0.4.1->tweepy)
  Downloading oauthlib-2.0.1.tar.gz (122kB)
 100% |#####|: 133kB 1.5MB/s
Installing collected packages: requests, oauthlib, requests-oauthlib, six, tweepy
  Running setup.py install for oauthlib ... done
Successfully installed oauthlib-2.0.1 requests-2.13.0 requests-oauthlib-0.8.0 six-1.10.0 tweepy-3.5.0
```

## 스타버즈 코드: 트위터 접속 [실습 5]

```
import tweepy
import time
import urllib

# 트위피 라이브러리 사용하여 트위터에 메시지 전송하는 함수
def send_to_twitter(msg):
    CONSUMER_KEY = 'TlcvCB9*****UAMdYFRm'
    CONSUMER_SECRET = 'ecWtLw6Gf4*****zNkLVY0Q Sb4G91'
    ACCESS_KEY = '4*****R6CHlgk0xH2jrDbA72FFOOytD'
    ACCESS_SECRET = 'Vh*****5sy3hFsQDHqb7'

    auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
    auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)
    api = tweepy.API(auth)
    api.update_status(msg)
```

```
def get_price():
    page = urllib.request.urlopen("http://cs.sch.ac.kr/prices-loyalty.py ")
    text = page.read().decode("utf8")
    where = text.find('>$')
    start_of_price = where + 2
    end_of_price = start_of_price + 4
    return float(text[start_of_price:end_of_price])
```

```
price_now = input("Do you want to see the price now (Y/N)? ")
```

```
if price_now == "Y":
    msg = "Buy now!, price $" + str(get_price())
    send_to_twitter(msg)
```

```
else:
    price = 99.99
    while price > 4.74:
        time.sleep(900)
        price = get_price()
    msg = "Buy!, price $" + str(price)
    send_to_twitter(msg)
```

프로그래밍 기초



시험주행

```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:15:05) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Do you want to see the price now (Y/N)? Y
>>> ===== RESTART =====
>>>
Do you want to see the price now (Y/N)? N
>>> |
```



이상정 @cssjlee · 20분  
Buy!, price \$4.6

이상정 @cssjlee · 20분  
Buy now!, price \$5.48



## 프로그래밍 도구상자 (1)

### □ 프로그래밍 도구

- 함수를 사용하여 코드를 반복해서 작성하지 않음
- 인자는 함수에 전달되는 변수
- 함수는 값을 반환
- 컴퓨터는 스택 프레임을 사용하여 변수를 기록하고 추적
- 함수를 호출하면 함수가 사용할 수 있도록 새로운 스택 프레임을 만듦
- 함수의 실행이 끝나면 스택 프레임과 지역 변수는 제거
- 코드가 변수를 볼 수 있다면 그 변수는 범위 안에 있다고 함



## 프로그래밍 도구상자 (2)

### □ 파이썬 도구

- def를 사용하여 함수를 생성
- return()을 사용하여 함수를 호출한 코드로 값을 전달
- 괄호 안에 인자를 넣어서 함수에 전달



# 보충학습 - 누적 덧셈 함수 [실습 4]

□ 1장의 누적 덧셈을 함수로 작성

```
def add(n):
    number = 1
    sum = 0
    while number <= n:
        sum = sum + number
        number = number + 1
    return sum

print("---- add(10)")
print(add(10))
print("---- add(100)")
print(add(100))
```



# 시험주행

```
Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:15:05) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
---- add(10)
55
---- add(100)
5050
>>> |
```



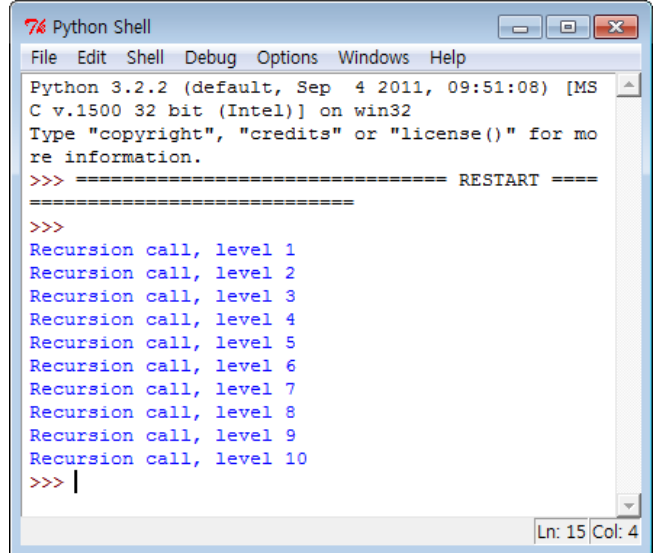
# 보충학습 - 재귀함수

## □ 재귀함수 (recursion)

- 자기 자신의 함수를 호출하는 함수
- 재귀함수 작성 시 자기 자신으로 무한히 반복되지 않도록 작성

```
# 재귀함수 수준을 제어하는 예
def f(level):
    # 현재의 수준을 출력
    print("Recursion call, level",level)
    # 수준 10을 넘어서지 않으면
    if level < 10:
        # 수준을 증가시키고 자신의 함수를 호출
        f(level+1)

# 수준 1에서 재귀함수 호출
f(1)
```



# 보충학습 - 재귀함수, 누적 덧셈 [실습 4]

## □ 누적 덧셈

- $sum(N) = 1 + 2 + 3 + \dots + (N-1) + N$   
 $= sum(N-1) + N$
- 재귀적 정의
  - 1, if N=1
  - $sum(N-1) + N$ , if N > 1

```
def sum(n):
    if (n == 1):
        return 1
    else:
        return sum(n-1)+n

print("---- sum(10)")
print(sum(10))

print("---- sum(100)")
print(sum(100))
```



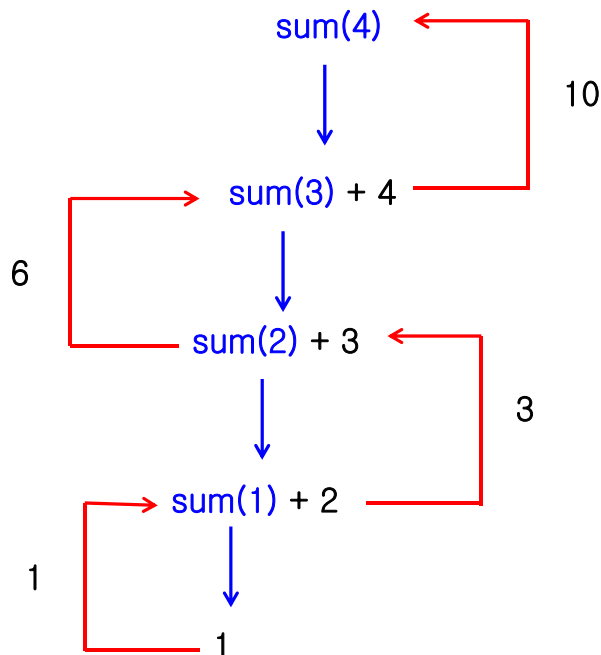


```

Python 3.4.2 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:15:05) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
---- sum(10)
55
---- sum(100)
5050
>>> |

```

## 보충학습 - 재귀함수, 누적 덧셈 과정



## 3장 실습

[실습 1] 원두 가격 출력 함수 예 1

[실습 2] 원두 가격 출력 함수 예 2

[실습 3] 스타버즈 코드: 긴급 주문 옵션

[실습 4] 앞의 누적 덧셈 함수와 재귀함수 누적 덧셈을 작성하고 실행 과정 분석

[실습 5] 스타버즈 코드: 트위터 접속”을 다음과 같이 수정하여 작성하고 실행

- 자신의 이름과 학번을 추가하여 예제의 트위터로 전송

## 3장 과제

1. [실습 1] ~ [실습 4]의 프로그램 작성 및 실행 과정
2. 자신의 트위터 계정으로 애플리케이션을 등록하여 자신의 트위터에 [실습 5]를 반복
3. 앞에서 배운 내용을 사용한 임의의 프로그램 작성
  - 프로그램 설명
  - 프로그램 소스
  - 실행 결과