재귀함수 (Recursion)

순천향대학교 컴퓨터공학과
이 상 정

학습 내용

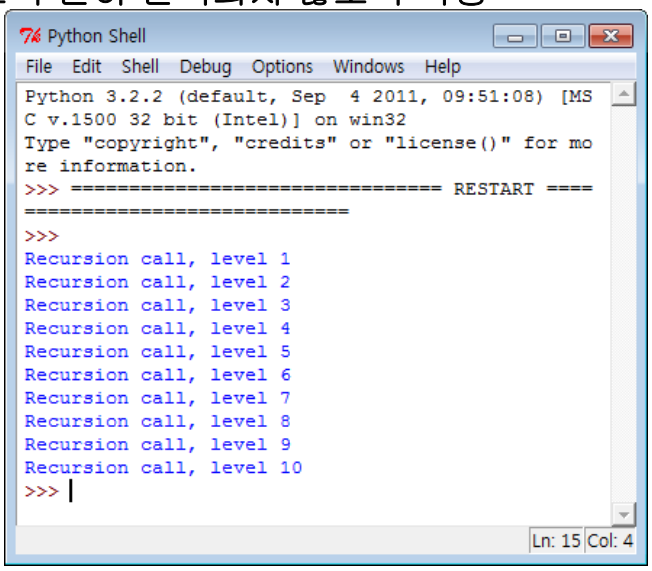
- 재귀함수 소개
- 누적 뎃셈 예
- 팩토리얼 계산
- 사각형 반복 예
- 프렉탈 그리기

재귀함수 소개

❑ 재귀함수

- 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)
```

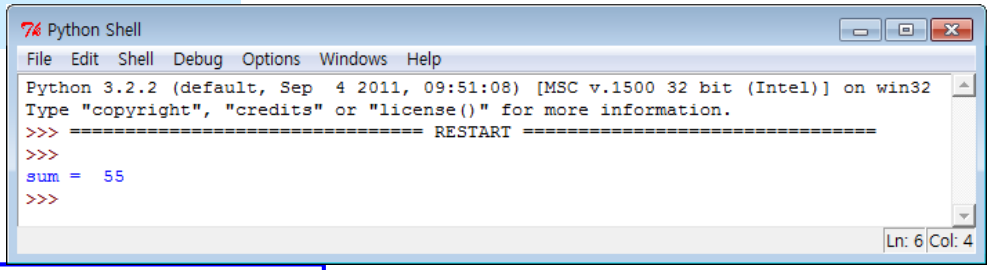


일반 함수: 1+2+ …… + N

❑ 누적 덧셈 예

- 1 + 2 + 3 + …… + N

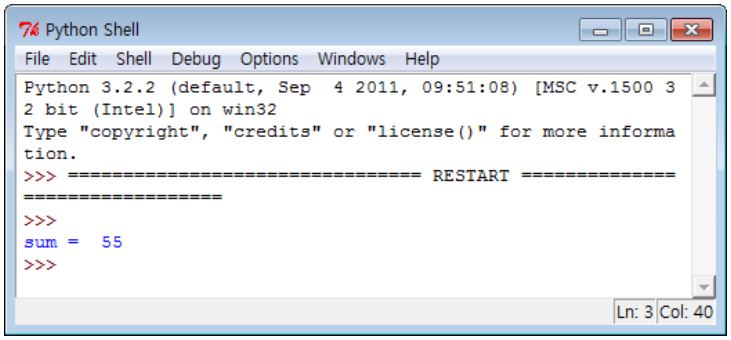
```
def sum(n):
    result = 0
    for i in range(1, n+1):
        result = result + i

print("sum = ", sum(10))
```



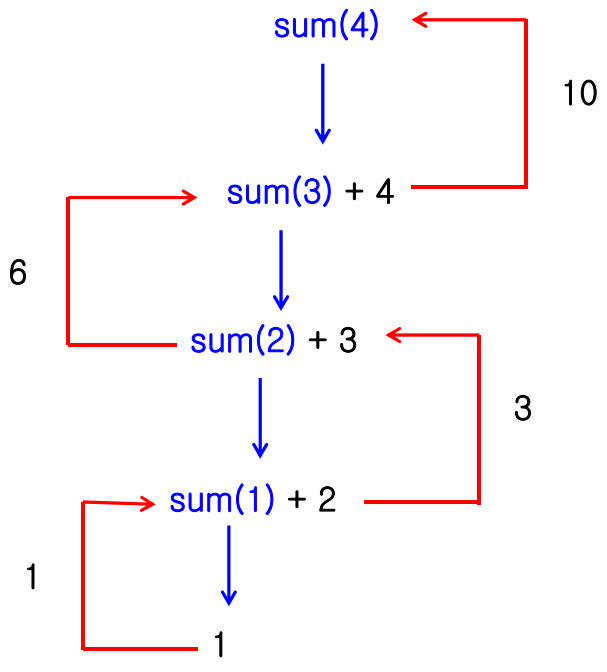
재귀함수: 1+2+ …… + N (1)

- 누적 덧셈
  - sum(N) = 1 + 2 + 3 + …… + (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 = ", sum(10))
```



재귀함수: 1+2+ …… + N (2)



팩토리얼 계산 예 (1)

- 팩토리얼 (factorial) 계산
  - N! = 1 * 2 * 3 * …… * (N−1)* N
    = (N−1)! * N
  - 재귀적 정의
    - 1, if N=1
    - (N−1)! * N, if N > 1

```
# 팩토리알 일반함수
def factorial_nonrecursive(n):
    answer = 1
    for i in range(2,n+1):
        print( i,"*",answer,"=", i*answer)
        answer = answer * i
    return answer
```

```
# 팩토리알 재귀함수
def factorial_recursive(n):
    if (n == 1):
        return  n
    else:
        x = factorial_recursive(n-1)
    print( n, "*", x, "=", n * x )
    return n * x
```


팩토리얼 계산 예 (2)

```
# 팩토리알 일반함수
def factorial_nonrecursive(n):
    answer = 1
    for i in range(2,n+1):
        print( i,"*",answer,"=", i*answer)
        answer = answer * i
    return answer

print("I can calculate a factorial!")
user_input = input ("Enter a number:")
n = int(user_input)
answer = factorial_nonrecursive(n)
print (answer)
```

```
# 팩토리알 재귀함수
def factorial_recursive(n):
    if( n == 1 ):
        return n
    else:
        x = factorial_recursive(n-1)
    print( n, "*", x, "=", n * x )
    return n * x

print("I can calculate a factorial!")
user_input = input ("Enter a number:")
n = int(user_input)
answer = factorial_recursive(n)
print (answer)
```

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 ================================
>>>
I can calculate a factorial!
Enter a number:7
2 * 1 = 2
3 * 2 = 6
4 * 6 = 24
5 * 24 = 120
6 * 120 = 720
7 * 720 = 5040
5040
I can calculate a factorial!
Enter a number:7
2 * 1 = 2
3 * 2 = 6
4 * 6 = 24
5 * 24 = 120
6 * 120 = 720
7 * 720 = 5040
5040
>>>
                                                              Ln: 22  Col: 4
```

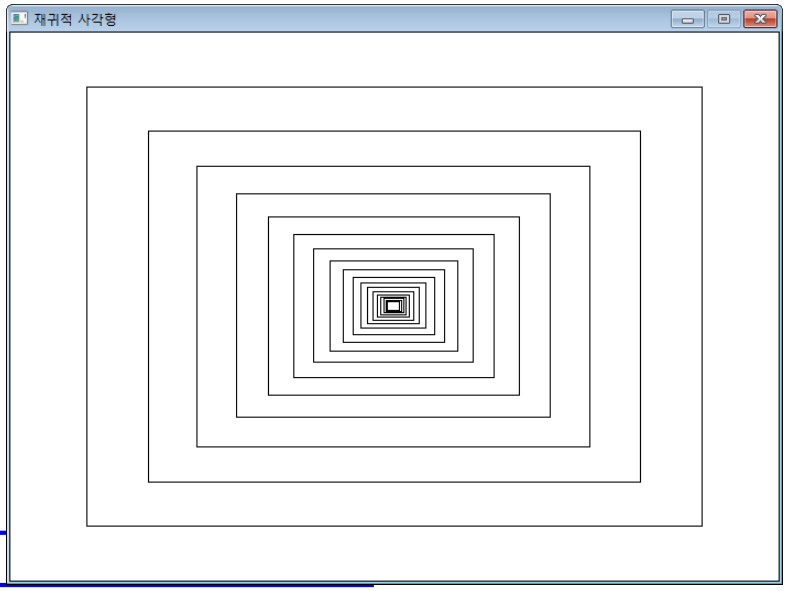

재귀함수 응용

- 재귀함수는 동일한 구조가 반복되는 문서에도 적용
- 예
  - 웹문서

  - e-메일
    - 한 e-메일이 다른 사람의 e-메일 포함(attach)
    - 다른 사람의 e-메일이 또 다른 사람의 e-메일을 포함
    - ……

사각형 반복 예 (1)

- 동일한 사각형이 20% 크기가 줄어들면서 반복적으로 그려지는 예
  - 한 변의 너비가 14 픽셀보다 크면 반복



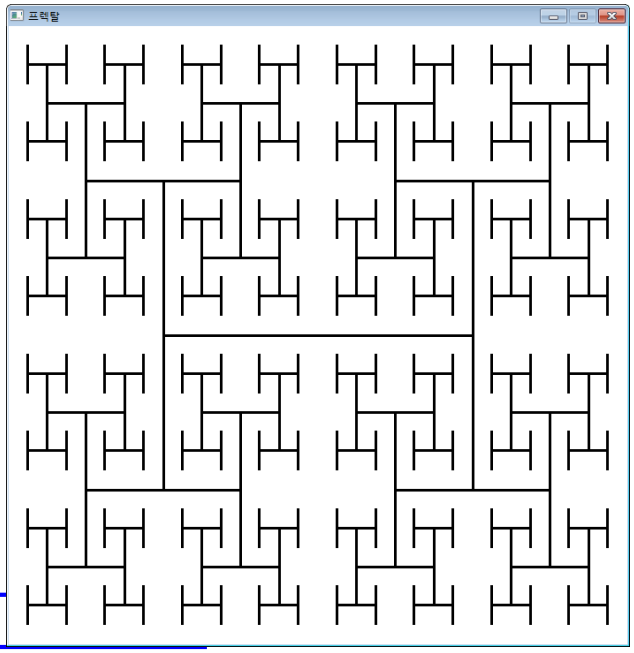
사각형 반복 예 (2)

```
import pygame

black = ( 0, 0, 0)
white = ( 255, 255, 255)

def recursive_draw(x,y,width,height):
    # 사각형 그리기
    pygame.draw.rect(screen,black,[x,y,width,height],1)
    # 너비가 14 픽셀보다 크면 반복
    if( width > 14 ):
        # 좌표 및 크기 축소 조정
        x += width * .1
        y += height * .1
        width *= .8
        height *= .8
        # 재귀함수 호출
        recursive_draw(x,y,width,height)

pygame.init()
```

```
# 윈도우 설정
size=[700,500]
screen=pygame.display.set_mode(size)
pygame.display.set_caption("재귀적 사각형")

done=False
clock=pygame.time.Clock()
while done==False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True

    screen.fill(white)
    recursive_draw(0,0,700,500)
    clock.tick(20)
    pygame.display.flip()

pygame.quit()
```

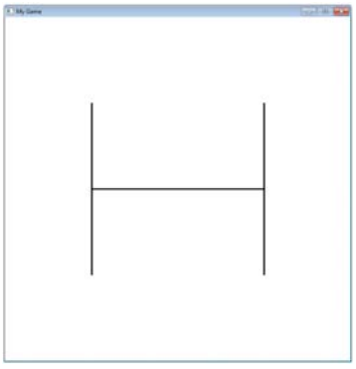
프렉탈 도형 그리기 (1)

- 프렉탈(fractal)
  - 언제나 부분이 전체를 닮는 자기 유사성(self-similarity) 기하학적인 도형

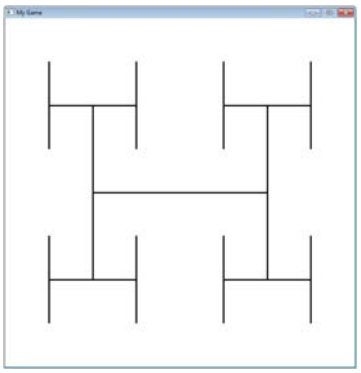


프렉탈 도형 그리기 (2)

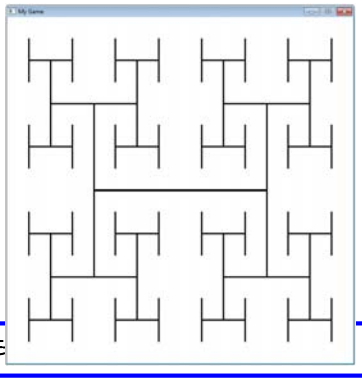
재귀적 프렉탈 레벨 0



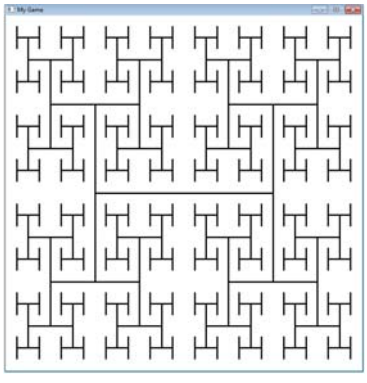
재귀적 프렉탈 레벨 1



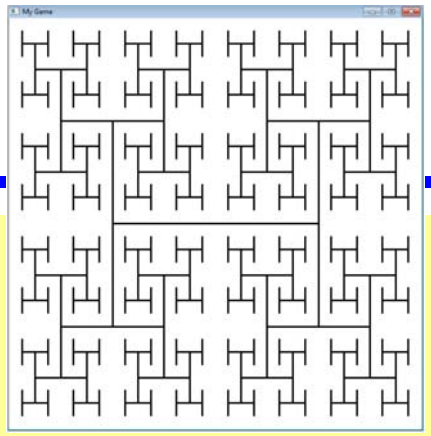
재귀적 프렉탈 레벨 2



재귀적 프렉탈 레벨 3



프렉탈 도형 그리기 (3)

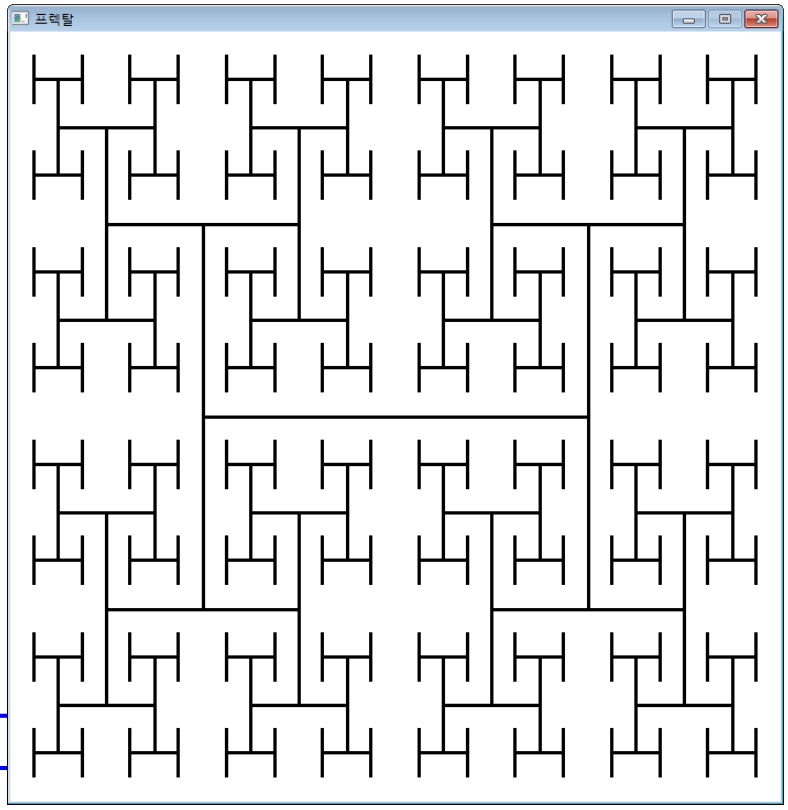


```
import pygame
black = ( 0, 0, 0)
white = ( 255, 255, 255)

def recursive_draw(x,y,width,height,count):
    # 수평선 그리기
    pygame.draw.line(screen,black,[x+width*.25,y+height*0.5],[x+width*.75,y+height*0.5],3)
    # 왼쪽 수직선 그리기
    pygame.draw.line(screen,black,[x+width*.25,y+height*.25],[x+width*.25,y+height*0.75],3)
    # 오른쪽 수직선 그리기
    pygame.draw.line(screen,black,[x+width*.75,y+height*.25],[x+width*.75,y+height*0.75],3)
    if count > 0:
        count -= 1
        # Top left
        recursive_draw(x,y, width//2,height//2,count)
        # Top right
        recursive_draw(x+width//2,y, width//2,height//2,count)
        # Bottom left
        recursive_draw(x,y+width//2, width//2,height//2,count)
        # Bottom right
        recursive_draw(x+width//2,y+width//2, width//2,height//2,count)
```


프렉탈 도형 그리기 (4)

```
pygame.init()
# 윈도우 설정
size=[700,700]
screen=pygame.display.set_mode(size)
pygame.display.set_caption("프렉탈")

done=False
clock=pygame.time.Clock()
while done==False:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done=True

    screen.fill(white)
    fractal_level=3
    recursive_draw(0,0,700,700,fractal_level)
    clock.tick(20)
    pygame.display.flip()

pygame.quit ()
```

시험주행



과제

1. 앞에서 배운 내용을 사용한 임의의 프로그램 작성
   - 프로그램 설명
   - 프로그램 소스
   - 실행 결과