



Data Analytics Based Python

SECT7. 실습 프로젝트

IT Competency Improvement Training
Kim Jin Soo



- ◆ 구구단 출력하기
- ◆ 총합과 팩토리얼 테이블 출력하기
- ◆ 문자열 바꾸기
 - 대소문자 변경
 - 순서 변경
- ◆ 도서 목록 입력 및 출력
- ◆ 그래픽(Turtle Graphics) 출력하기
 - 모양 그리기, 다각형 그리기
 - 패턴을 찾아 모양을 그리기

구구단 출력하기



- ❖ 누구나 알고 있는 구구단을 1단만 화면에 출력 해보자.
 - 모든 문자는 스페이스 바로 구분
- ❖ 사용자로부터 숫자를 입력 받아서 구구단을 출력 해보자.

```
출력할 단을 입력해주세요. [2~9] 5
```

```
5 단 출력
```

```
-----
```

```
5 * 1 = 5
```

```
5 * 2 = 10
```

```
5 * 3 = 15
```

```
5 * 4 = 20
```

```
5 * 5 = 25
```

```
5 * 6 = 30
```

```
5 * 7 = 35
```

```
5 * 8 = 40
```

```
5 * 9 = 45
```

총합과 팩토리얼 테이블 출력하기



❖ 숫자값 N을 입력받아, 1~N까지의 Sum과 Factorial 구하기

- 입력값에 대한 Validation 체크 cf. 숫자값이 아닌경우 재입력 유도
- Sum과 Factorial 결과값을 List 에 저장 후, 한꺼번에 출력

```
숫자를 입력해 주세요. [1~100] => abc
입력한 값이 숫자가 아닙니다.
숫자를 입력해 주세요. [1~100] => 10
입력한 숫자 : 10
```

```
-----
10 까지의 합계 및 팩토리얼 테이블 구하기!!
-----
```

```
10 까지의 합계는 55 입니다.
아래는 팩토리얼 테이블입니다.
```

```
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
```



❖ 문자열 대소문자 변경하기

- 입력 받은 문자를 모두 대문자로 출력하라.
- 입력 받은 문자를 모두 소문자로 출력하라.
- 입력 받은 문자를 for문을 사용하여 대문자는 소문자로, 소문자는 대문자로 출력하라.
- 입력 받은 문자를 문자열형 메소드를 사용하여 대문자는 소문자로, 소문자는 대문자로 출력하라.

영어 대소문자로 이루어진 문장을 입력하세요.

The BPC find Information to design valuable society with Tech & Craft.

모두 대문자로 출력

THE BPC FIND INFORMATION TO DESIGN VALUABLE SOCIETY WITH TECH & CRAFT.

모두 소문자로 출력

the bpc find information to design valuable society with tech & craft.

대소문자 바뀌서 출력

THE bpc FIND iNFORMATION TO DESIGN VALUABLE SOCIETY WITH tECH & cRAFT.



❖ 문자열 순서 바꾸기

- 표준 입력문으로 받은 문자열의 순서를 바꿔서 역으로 산출
- 인덱스 사용법으로 역으로 산출

영어 문장을 입력하세요.

Life is too short, You need Python

nohtyP deen uoY ,trohs oot si efiL

nohtyP deen uoY ,trohs oot si efiL



❖ 다양한 데이터 구조들을 활용하는 데에 초점

- 데이터 담기
- 여러 데이터 만들어 보기
- 소스 코드 단순화 하기

❖ 내가 읽은 책 목록

제목	출판연도	출판사	쪽수	추천유무
파이썬 프로그램	2016	A	200	X
플랫폼 비즈니스	2013	B	584	O
빅데이터 마케팅	2014	A	296	O
외식경영 전문가	2010	B	526	X
십억만 벌어보자	2013	A	248	O



❖ 데이터형과 데이터구조에서 배운 내용을 토대로 담기

- 1) 여러 책을 담을 수 있는 리스트형인 books라는 변수를 선언 및 초기화
- 2) Books의 각 항목에 한 권의 책을 담기 위한 사전형 데이터를 추가
- 3) 사전형 데이터는 변수 선언 없이 리스트에 바로 추가
- 4) 사전형 데이터에는 4개의 키와 값의 쌍이 존재하며,
키는 컬럼명이고 값은 각 책에 해당하는 컬럼 값이 된다.

❖ 입력내용

```
books = list()      # 책 목록 선언

# 책 목록 만들기
books.append({'제목': '파이썬 프로그램', '출판연도': '2016', '출판사': 'A', '쪽수': 200, '추천유무': False})
books.append({'제목': '플랫폼 비즈니스', '출판연도': '2013', '출판사': 'B', '쪽수': 584, '추천유무': True})
books.append({'제목': '빅데이터 마케팅', '출판연도': '2014', '출판사': 'A', '쪽수': 296, '추천유무': True})
books.append({'제목': '외식경영 전문가', '출판연도': '2010', '출판사': 'B', '쪽수': 526, '추천유무': False})
books.append({'제목': '십억만 벌어보자', '출판연도': '2013', '출판사': 'A', '쪽수': 248, '추천유무': True})

for book in books: # 책 한 권씩 꺼내기 위한 루프 선언
    print(book)     # 책 한 권 데이터 출력
```




❖ 데이터형과 데이터구조에서 배운 내용을 토대로 담기

- 1) 여러 책을 담을 수 있는 리스트형인 books라는 변수를 선언 및 초기화
- 2) Books의 각 항목에 한 권의 책을 담기 위한 사전형 데이터를 추가
- 3) 사전형 데이터는 변수 선언 없이 리스트에 바로 추가
- 4) 사전형 데이터에는 4개의 키와 값의 쌍이 존재하며,
키는 컬럼명이고 값은 각 책에 해당하는 컬럼 값이 된다.

❖ 출력내용

```
for book in books: # 책 한 권씩 꺼내기 위한 루프 선언
    print(book)     # 책 한 권 데이터 출력
```

```
{'추천유무': False, '출판연도': '2016', '쪽수': 200, '출판사': 'A', '제목': '파이썬 프로그램'}
{'추천유무': True, '출판연도': '2013', '쪽수': 584, '출판사': 'B', '제목': '플랫폼 비즈니스'}
{'추천유무': True, '출판연도': '2014', '쪽수': 296, '출판사': 'A', '제목': '빅데이터 마케팅'}
{'추천유무': False, '출판연도': '2010', '쪽수': 526, '출판사': 'B', '제목': '외식경영 전문가'}
{'추천유무': True, '출판연도': '2013', '쪽수': 248, '출판사': 'A', '제목': '십억만 벌어보자'}
```



❖ 작성한 소스의 books를 활용하여 아래 질문에 답하기 추가하기

- 1) 내가 읽은 책 중 250쪽이 넘는 책의 제목으로 이루어진 리스트형 변수 many_page를 만든다.
- 2) 내가 읽은 책 중 추천하고 싶은 책의 제목으로 이루어진 리스트형 변수 recommends를 만든다.
- 3) 내가 읽은 책의 전체 쪽수를 담은 숫자형 변수 all_pages를 만든다.
- 4) 내가 읽은 책의 출판사를 위한 세트 형 변수 pub_company를 만든다.

❖ 출력내용

```
print('쪽수가 250 쪽 넘는 책 리스트:', many_page)
print('내가 추천하는 책 리스트:', recommends)
print('내가 읽은 책 전체 쪽수:', all_pages)
print('내가 읽은 책의 출판사 목록:', pub_companies)
```

```
쪽수가 250 쪽 넘는 책 리스트: ['플랫폼 비즈니스', '빅데이터 마케팅', '외식경영 전문가']
내가 추천하는 책 리스트: ['플랫폼 비즈니스', '빅데이터 마케팅', '십억만 벌어보자']
내가 읽은 책 전체 쪽수: 1854
내가 읽은 책의 출판사 목록: {'B', 'A'}
```

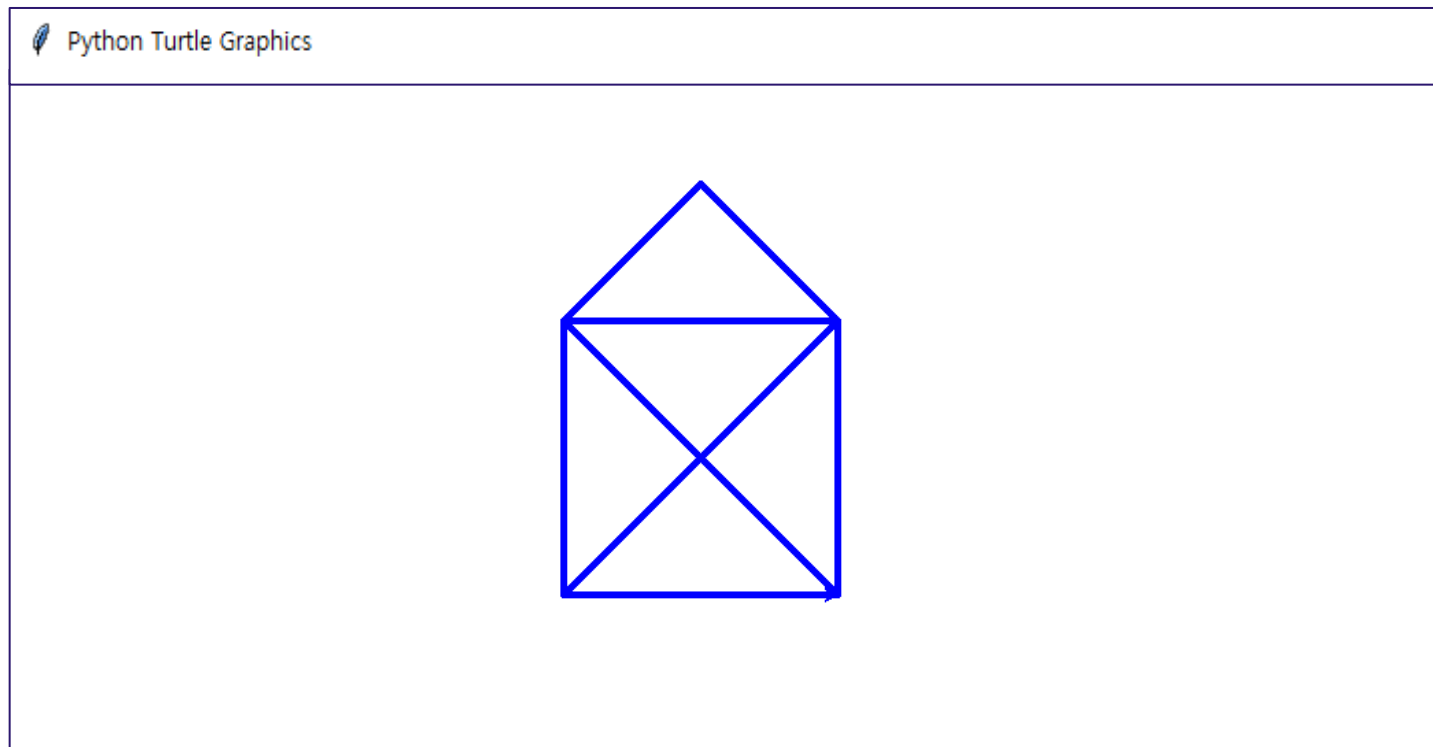


❖ 아래 모양을 최소한의 코드로 그려보기.

- Hint : 대각선 길이를 구하는 제곱근 (Root Square)

```
import math
```

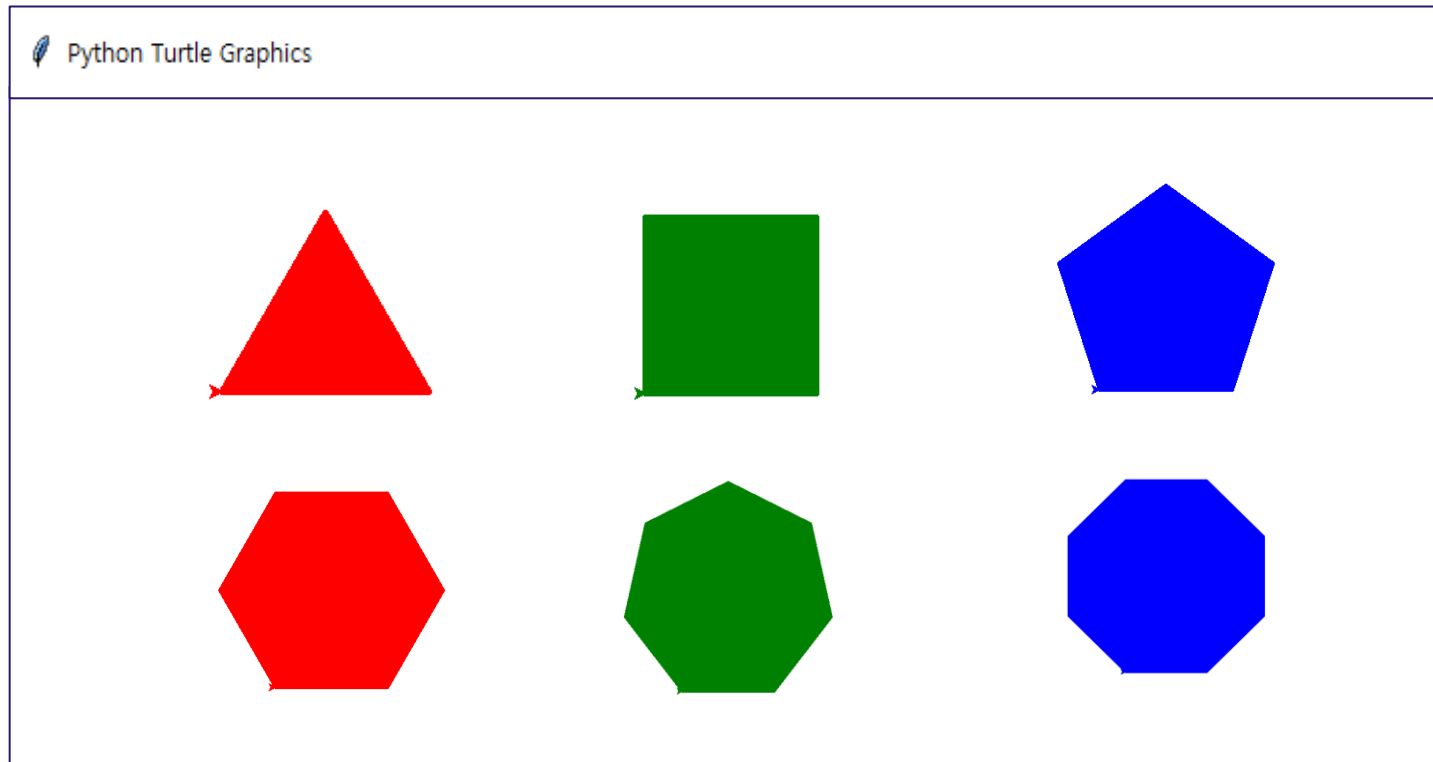
```
math.sqrt(width**2 + height**2)
```





❖ 숫자값을 입력받아 다각형을 그려보기

- 입력값1 : `var1 = input('변의 수를 입력해주세요? [3-8] ')`
- 입력값2 : `var2 = input('한변의 길이를 입력해주세요? [100-200] ')`

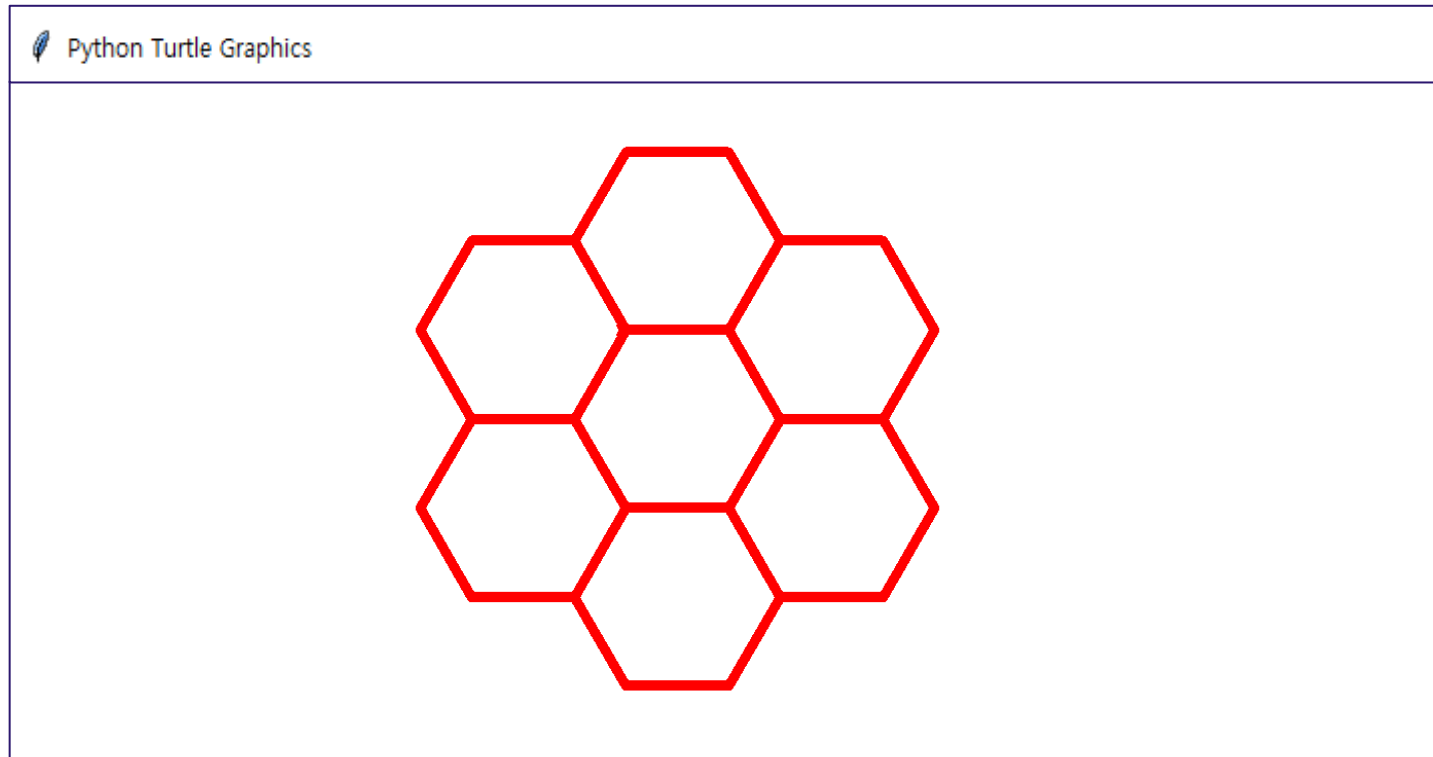


패턴을 찾아, 모양을 그려보기



❖ 반복문(for 혹은 while)을 사용하여 그려보기

- 1단계 : 먼저 정육각형을 그려보기
- 2단계 : 정육각형을 회전 이동하면서 반복하여 그려보기



복합 패턴을 찾아, 모양을 그려보기



❖ 복합적인 패턴을 찾아서, 반복문을 적용하기

- 패턴 : 각도, 길이, 굵기, 색상
- Hint : 적용된 색상 값은 아래와 같다.
 - colors = ['red', 'green', 'blue', 'yellow', 'purple', 'cyan', 'magenta', 'violet']

