



1. 파이썬 시작



1.1 파이썬 설치

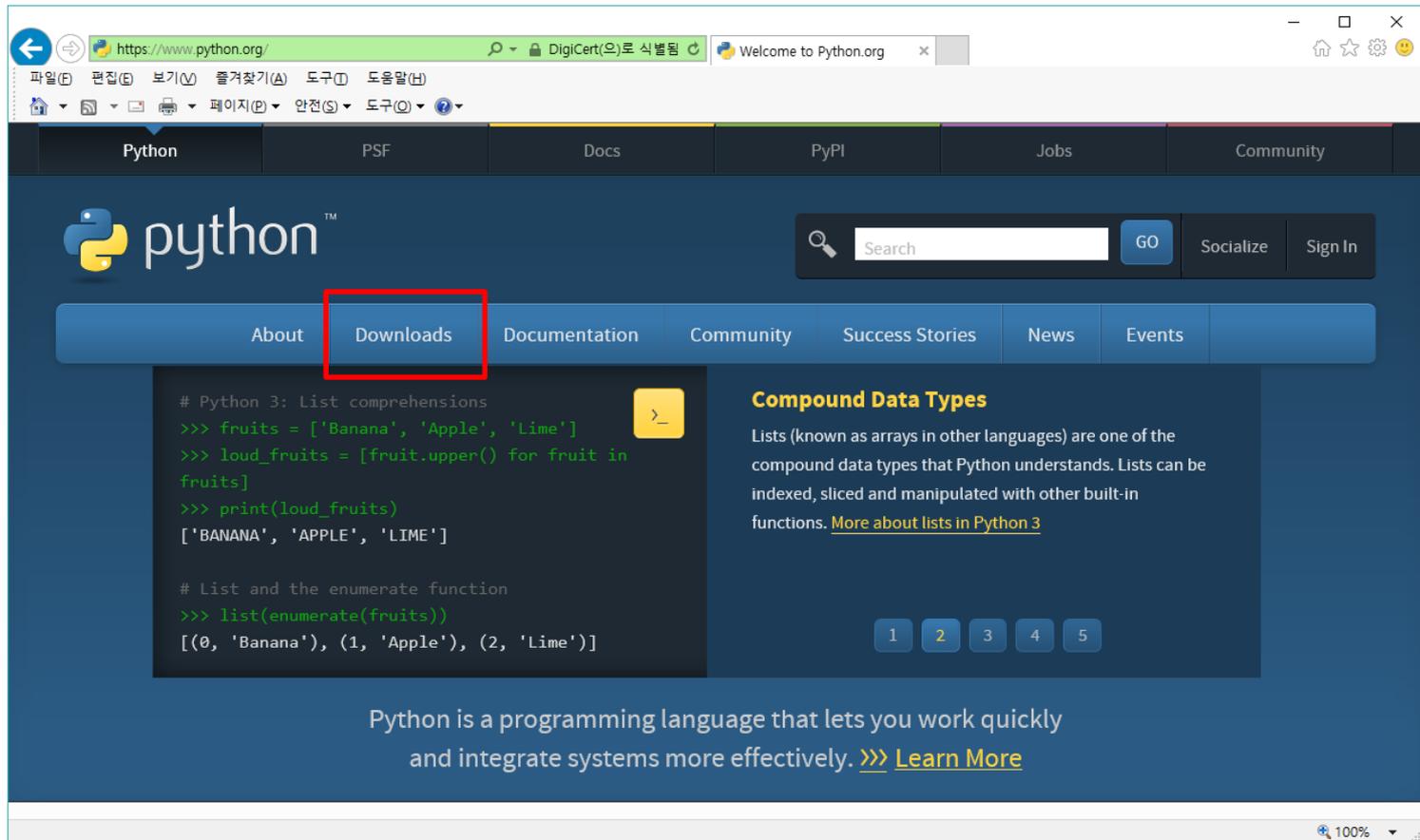
파이썬(Python)

■ 파이썬(Python)

- 귀로 반 로섬(Guido van Rossum)이 만든 고급언어.
- 많은 대학에서 프로그래밍을 배우는 첫 언어로 사용하고 있음
- 인터프리터 방식의 프로그래밍 언어
 - 호환이 되지 않는 2.x 버전과 3.x 버전이 있음
 - 교재에서는 3.6버전을 사용하고 있음
- 다양한 라이브러리 모듈을 활용할 수 있음
 - **Psyco, Numpy** 등의 모듈을 사용하여 데이터과학 등 여러 분야에서 사용됨
- 자유 소프트웨어(**free software**)로 무료로 이용 가능
- 홈페이지 <https://www.python.org>

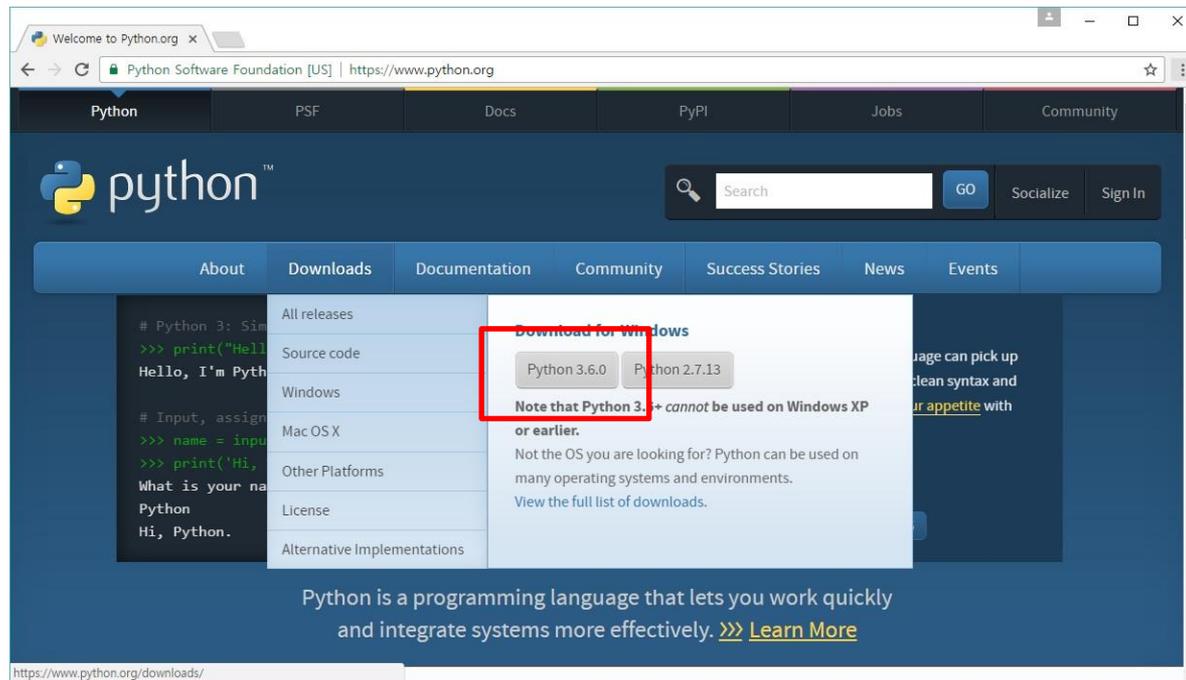
파이썬 홈페이지

- <https://www.python.org>



파이썬 설치

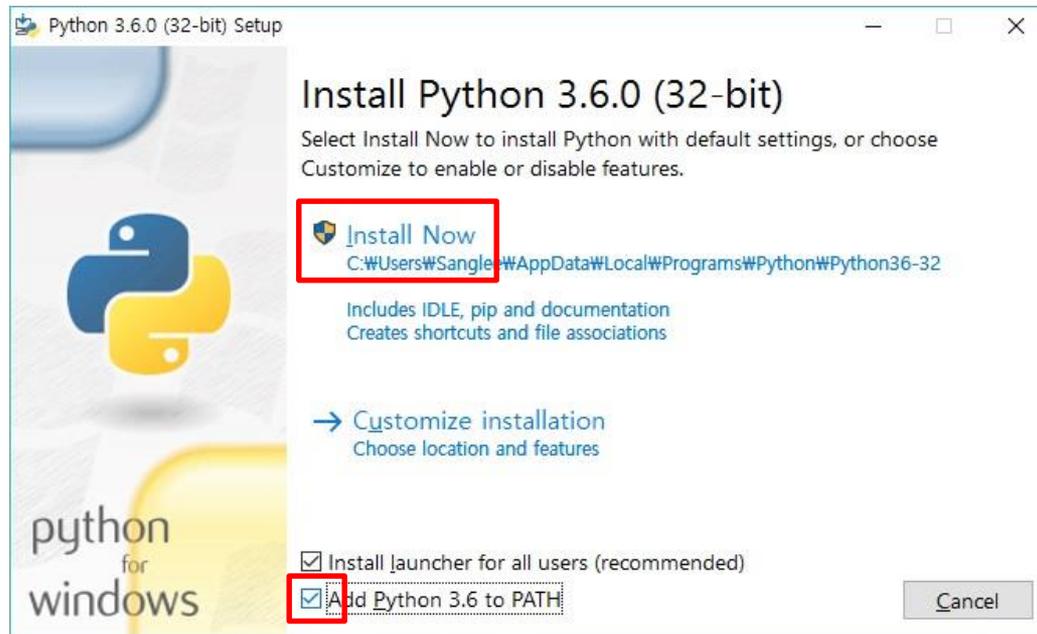
- 홈페이지의 내 메뉴바의 'Downloads' 버튼 클릭



- 'Python 3.6.0' 버튼 클릭(2.7.x 버전은 교과서와 다를 수 있음)
- 지시에 따라 설치 진행

파이썬 설치

- 'Add Python 3.6 to PATH' 체크 박스 체크
 - 명령프롬프트 임의의 디렉터리에서 사용 가능



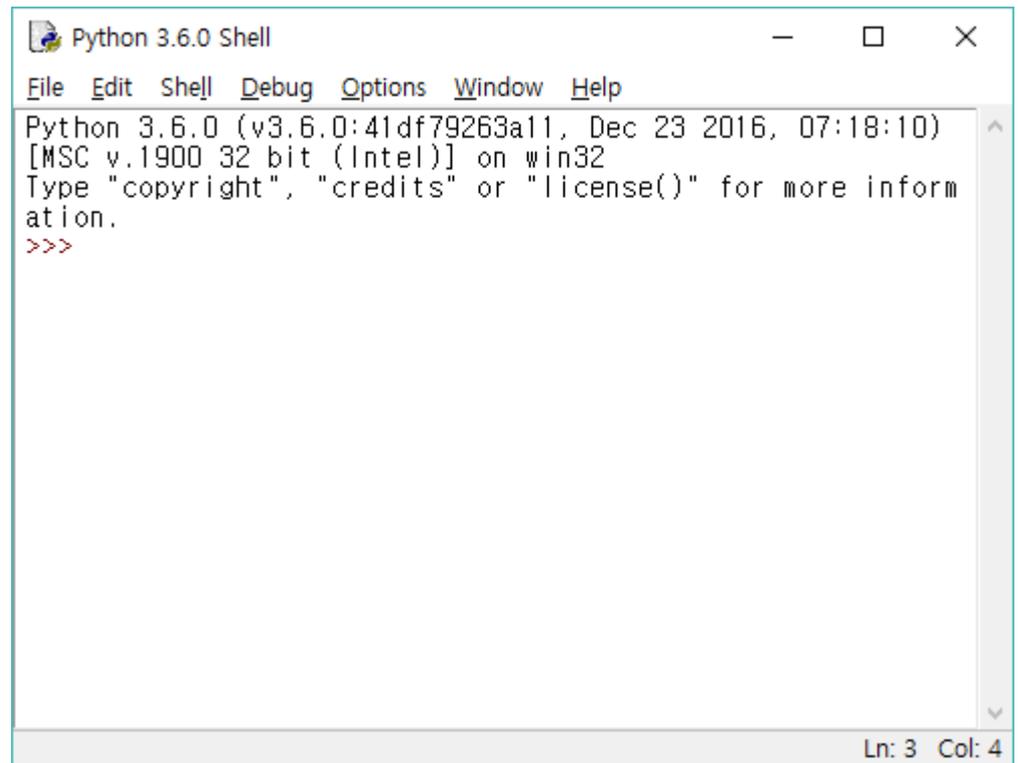
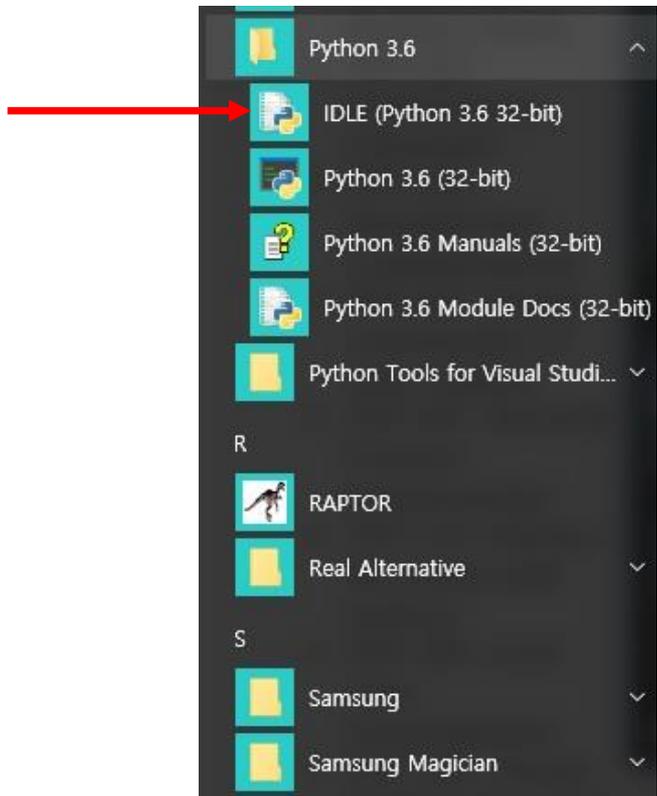
- 'Install Now' 클릭
- 지시에 따라 설치 진행



1.2 파이썬 **IDLE** 소개

파이썬 통합개발환경 실행

- 파이썬 IDLE 실행: 윈도우 시작버튼의 프로그램 목록에서 'IDLE(Python 3.6 32-bit)' 클릭



파이썬 프로그램

■ 명령문 혹은 문장(statement)

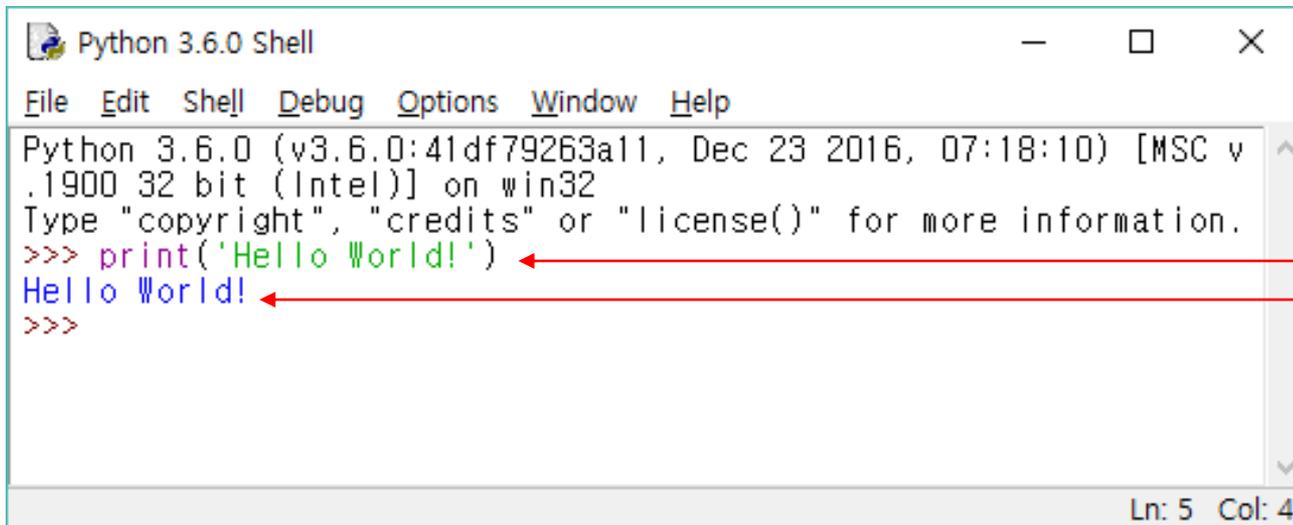
- 컴퓨터가 이해하여 수행할 수 있는 명령어들을 프로그래밍 언어로 표현한 문장.

■ 프로그램(program)

- 여러 개의 명령문(혹은 문장)들로 구성된다.
- 컴퓨터는 그 프로그램의 첫 번째 명령문부터 읽어서 그 의미를 파악하여 해당 명령문을 수행
- 명령문을 순차적으로 수행하여 프로그래머가 목적하는 일을 수행
- 이러한 과정을 마지막 명령문에 도달할 때까지 계속하고 종료

파이썬 IDLE

- 인터프리터 방식: 명령문을 한 줄 입력하면 바로 실행



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v
.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> print('Hello World!')
Hello World!
>>>
```

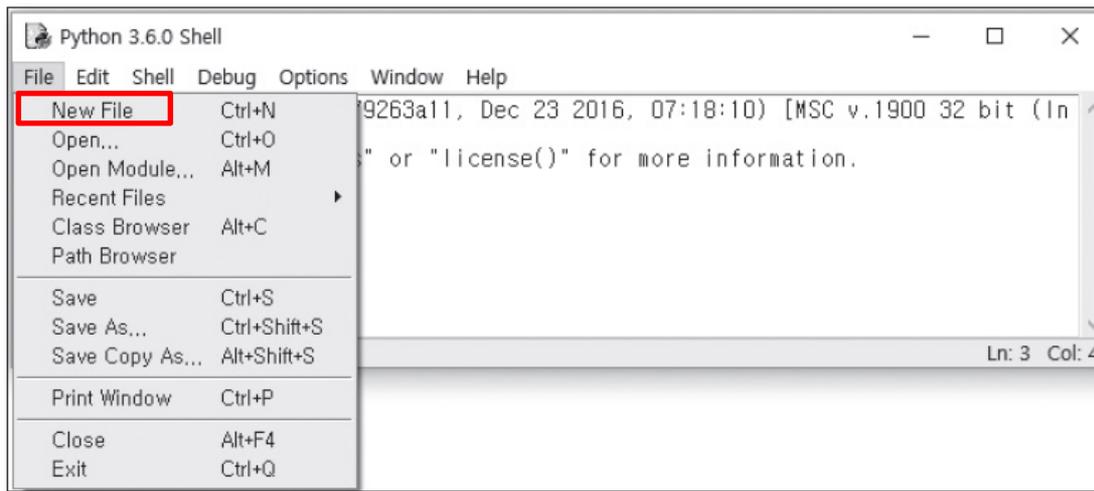
출력
실행
명령
결과
입력



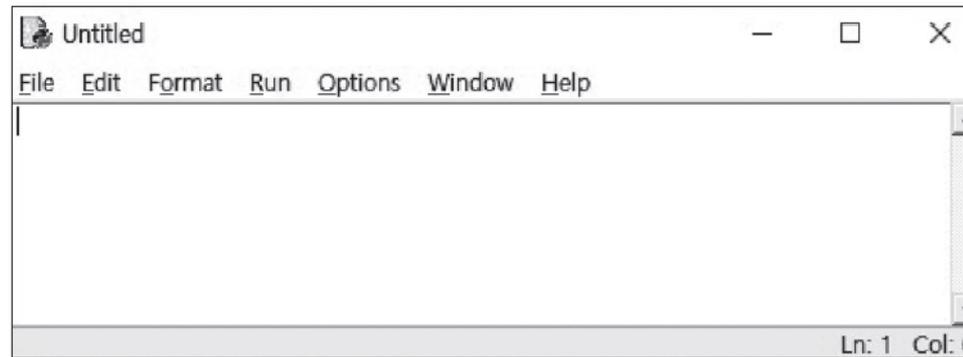
1.3 편집기 사용

IDLE 편집 셸

- IDLE의 'File' 메뉴에서 'New file'을 선택하여 편집셸 실행

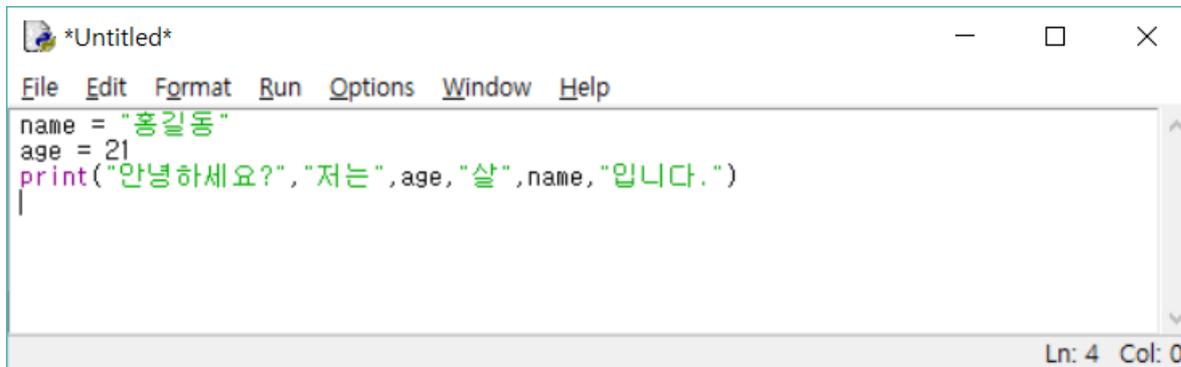


- 편집셸:



IDLE 편집 셸

- 대화형 셸: 명령문마다 실행
- 편집 셸: 프로그램 모두 작성 후 한번에 실행
 - 프로그램 입력



A screenshot of the IDLE Python IDE. The window title is '*Untitled*'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code editor contains the following Python code:

```
name = "홍길동"  
age = 21  
print("안녕하세요?", "저는", age, "살", name, "입니다.")
```

The status bar at the bottom right shows 'Ln: 4 Col: 0'.

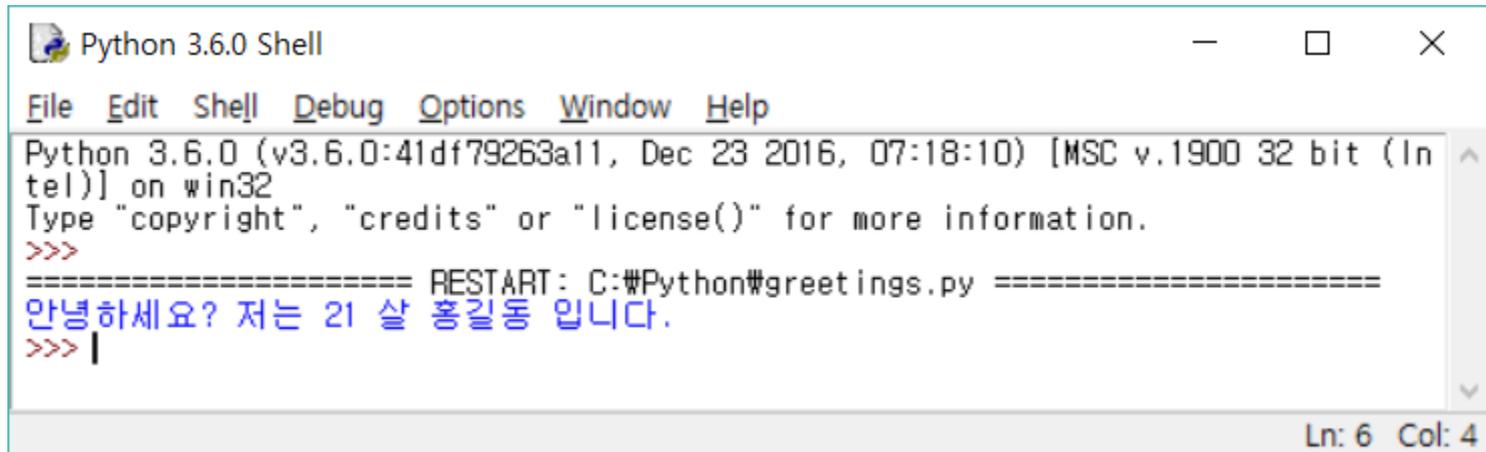
- 프로그램 실행: Run 메뉴의 'Run Module' 혹은 F5 키



A screenshot of the IDLE Python IDE, similar to the previous one, but with the 'Run' menu open. The menu items are: Python Shell, Check Module (Alt+X), and Run Module (F5). The 'Run Module' option is highlighted in blue. The code editor shows the same Python code as in the previous screenshot.

IDLE 편집 셸

■ 실행화면



```
Python 3.6.0 Shell
File Edit Shell Debug Options Window Help
Python 3.6.0 (v3.6.0:41df79263a11, Dec 23 2016, 07:18:10) [MSC v.1900 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Python#greetings.py =====
안녕하세요? 저는 21 살 홍길동 입니다.
>>> |
```

Ln: 6 Col: 4

□ 실행 결과는 대화형 셸에 나타난다

그래픽 프로그램

- 거북그래픽 모듈을 이용하여 사각형 그리기
 - turtle 그래픽 모듈은 파이썬에 기본적으로 포함되어 있음
 - 거북을 움직여 지나는 자리에 그림을 그릴 수 있음
 - 그러나 사용하려면 모듈을 프로그램으로 불러들인 후 사용
 - import 명령 사용하여 모듈 포함
 - 예, `import turtle`
 - 기본명령
 - `forward(d)` : d 픽셀 거리 만큼 앞으로 이동
 - `left(a)` : a도 각도 만큼 왼쪽으로 회전
 - 점 표기법
 - 대상 객체의 메소드 사용
 - 예, `t.forward(100)` : 거북 t를 100픽셀 전진

세 번째 프로그램

■ 정사각형 그리기

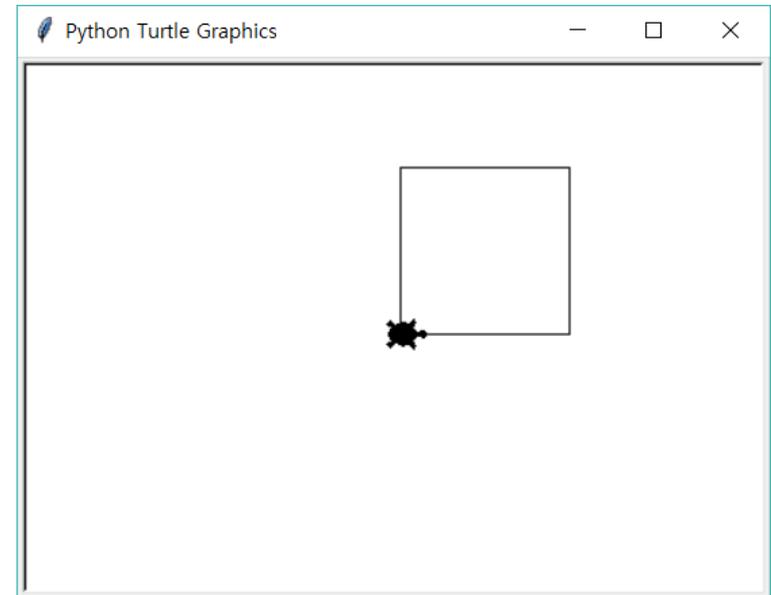
프로그램 3.4 거북그래픽 정사각형그리기

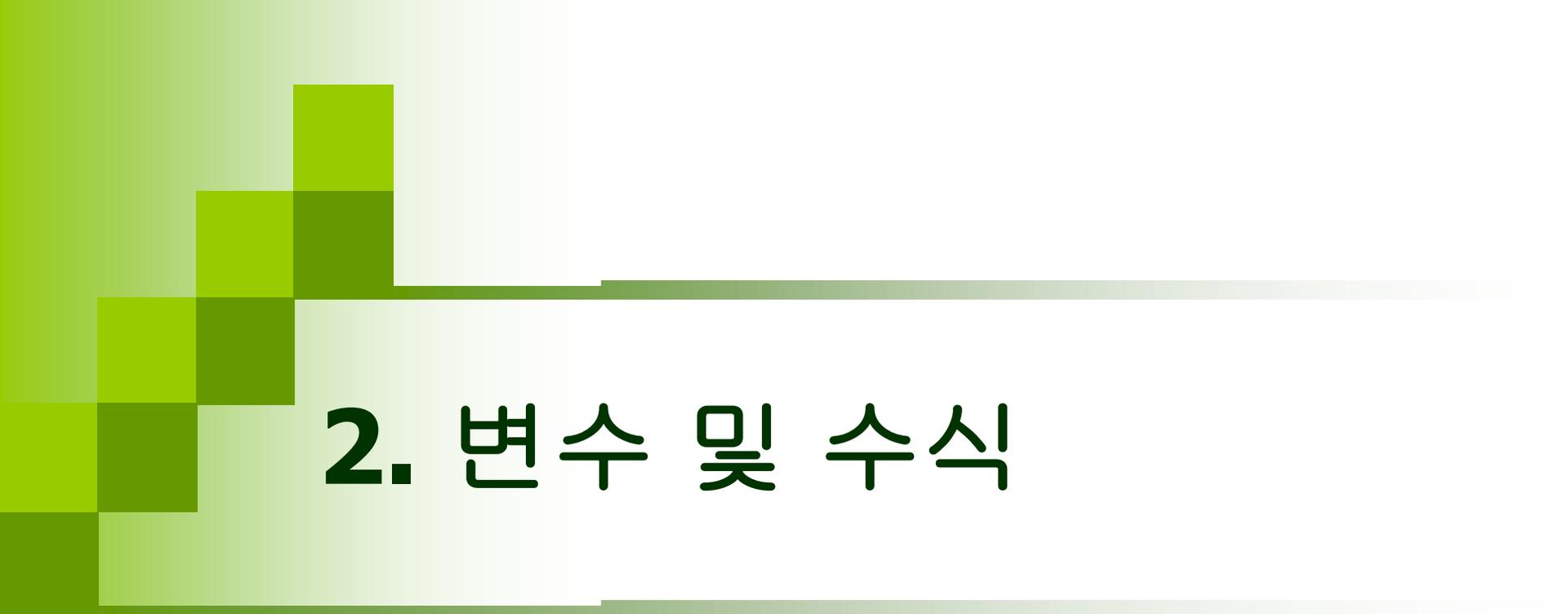
거북그래픽을 사용하여 정사각형 그리기

```
import turtle

t = turtle.Pen()
t.shape("turtle")

t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
```





2. 변수 및 수식



2.1 수식의 사용

수식의 사용

- 파이썬 언어를 가장 쉽게 사용하는 방법
 - 수식 계산기처럼 사용
 - 어떠한 수식이든 입력하면 그 값을 계산하여 바로 출력

프로그램 4.1 수식 예제



```
>>> 19 + 3
22
>>> (15 - 2 * 3) / 3
3
>>> 13 / 4
3.25
>>> 13 % 4
1
>>> 13.0 / 4
3.25
>>> 2.3 * 12.1
27.83
```

산술 연산

- 파이썬은 산술 연산으로 정수 및 실수에 대해서
가감승제, 나머지, 거듭제곱 등의 연산을 할 수 있다.

표 4.1 산술 연산

연산	연산자	정수 연산 예	실수 연산 예
덧셈	$a + b$	$19 + 3 = 22$	$3.5 + 12.6 = 16.1$
뺄셈	$a - b$	$12 - 4 = 8$	$11.5 - 7.3 = 4.2$
곱셈	$a * b$	$18 * 4 = 72$	$2.3 * 12.1 = 27.83$
나눗셈	$a // b$ a / b	$13 // 4 = 3$	$13 / 4 = 3.25$ $13.0 / 4.0 = 3.25$
나머지	$a \% b$	$13 \% 4 = 1$	$13.0 \% 4.0 = 1.0$ $13.1 \% 4.1 = 0.8000000$
거듭제곱	$a ** b$	$2 ** 5 = 32$	$2.5 ** 5 = 97.65625$

문자열

- 문자열은 작은따옴표(') 혹은 큰따옴표(")로 표시한다.

프로그램 4.2 문자열 예제



```
>>> 'Merry Christmas '  
'Merry Christmas '  
>>> "Merry Christmas"  
'Merry Christmas '  
>>> "메리 크리스마스 "  
'메리 크리스마스 '  
>>> "Today's date"  
"Today's date"  
>>> 'Today's date '  
SyntaxError: invalid syntax  
>>> 'Today\'s date '  
"Today's date"
```

문자열 연산

- 문자열 접합(+)
 - 두 문자열을 접합 즉 붙여서 새로운 문자열을 만들어 준다.
- 문자열 곱하기(*)
 - 대상 문자열을 반복한 새로운 문자열을 만들어 준다.

프로그램 4.3 문자열 연산



```
>>> 'Merry Christmas' + ' and Happy New Year!'
'Merry Christmas and Happy New Year!'
>>> 'Merry Christmas'*2
'Merry ChristmasMerry Christmas'
```

리스트

- 값들의 리스트사용 가능
 - 예를 들어, 문자열들의 리스트, 숫자들의 리스트, ...
 - 리스트에 접합('+')과 곱하기('*') 연산 가능

프로그램 4.4 리스트 예제

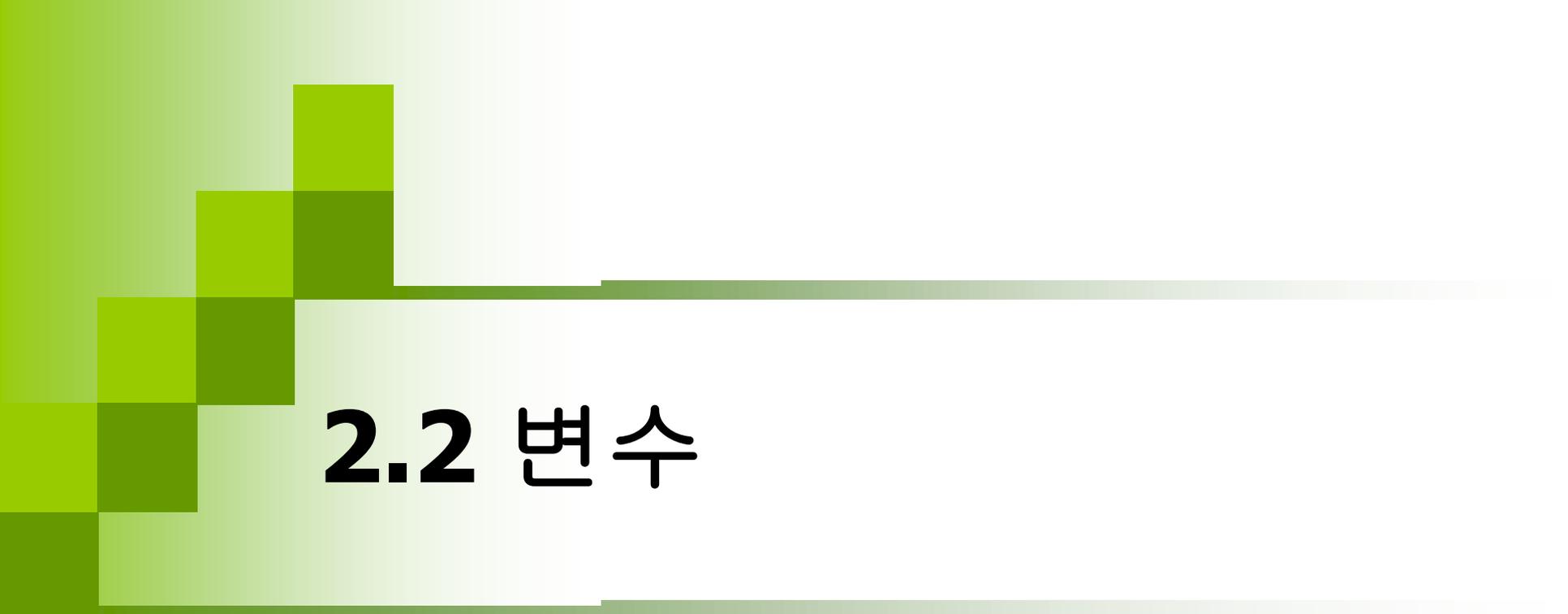


```
>>> ['seoul', 'tokyo', 'paris']
['seoul', 'tokyo', 'paris']
>>> ['seoul', 'tokyo', 'paris'] * 2
['seoul', 'tokyo', 'paris', 'seoul', 'tokyo', 'paris']
>>> ['seoul', 'tokyo', 'paris'] + ['new york', 'hongkong']
['seoul', 'tokyo', 'paris', 'new york', 'hongkong']
>>> [1,3,5]
[1,3,5]
>>> [1,3,5] * 2
[1,3,5,1,3,5]
>>> [1,3,5] + [7,9]
[1,3,5,7,9]
```

print 문

- print 문을 이용한 출력
- 여러 개의 값 출력
 - 콤마(,)로 구분하고 각각은 빈칸으로 구분되어 출력된다

```
>>> print(19 + 2)
21
>>> print((15 - 2 * 3) / 3)
3
>>> print(10, 20, 30)
10 20 30
>>> print('Merry Christmas' + ' and Happy New Year!')
Merry Christmas and Happy New Year!
>>> print('Merry Christmas'*2)
Merry ChristmasMerry Christmas
>>> print('Merry Christmas', 'Happy New Year!')
Merry Christmas Happy New Year!
>>> print("메리 크리스마스")
메리 크리스마스
>>> print(['seoul', 'tokyo', 'paris'])
['seoul', 'tokyo', 'paris']
>>> print(['서울', '도쿄', '파리'])
['서울', '도쿄', '파리']
>>> print(['서울', '도쿄', '파리'] + ['뉴욕', '홍콩'])
['서울', '도쿄', '파리', '뉴욕', '홍콩']
>>> print([1,3,5])
[1,3,5]
>>> print([1,3,5] * 2)
[1,3,5,1,3,5]
```



2.2 변수

변수

- 변수는 숫자, 문자 혹은 리스트 등의 값을 저장하기 위한 기억장소이다.
 - 변수는 따로 자료형을 선언할 필요가 없다.
- 변수의 사용
 - 변수의 값을 읽거나, 새로운 값을 저장하여 변경 가능
 - 새로운 값을 저장하면 기존값은 없어지고 새로운 값만 저장됨.
- 변수의 사용 예
 - `score = 70`
 - `print(score)`
 - `score = 80`
 - `score = 45 * 2`

대입문

- 변수에 값을 저장할 때 대입문을 사용한다

변수 = 수식

- 먼저 수식을 계산하고 그 결과값을 변수에 저장한다.

프로그램 4.6 두 점수 출력



```
>>> # 두 점수 출력 예제
>>> score = 70
>>> print("첫 번째 점수:", score)
첫 번째 점수: 70
>>> score = 80
>>> print("두 번째 점수:", score)
두 번째 점수: 80
>>> score = 45 * 2
>>> print("세 번째 점수:", score)
세 번째 점수: 90
```

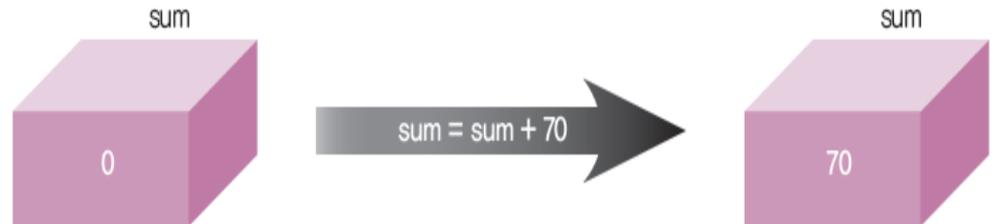
변수의 활용

- `sum = 0`
- `sum = sum + 70`
 - 변수 `sum`의 값에 70을 더해서 새로운 값을 계산한 후에
 - 이 계산된 값을 다시 변수 `sum`에 저장

프로그램 4.7 두 점수의 합 계산 1



```
>>> sum = 0
>>> sum = sum + 70
>>> print("첫 번째 합:", sum)
첫 번째 합: 70
>>> sum = sum + 90
>>> print("두 번째 합:", sum)
두 번째 합: 160
```



변수와 사용자 입력

■ 사용자 입력

- `input` 함수는 사용자가 문자열을 입력하고 엔터키를 치면 입력된 한 줄의 문자열을 읽어서 준다.
- 예 `name = input("이름 입력: ")`
 - 문자열 "이름 입력: "을 출력하고
 - 사용자로부터 입력을 받아,
 - 이를 변수 `name`에 저장한다.

프로그램 4.12 사용자에게 인사



```
>>> name = input("이름 입력: ")
이름 입력: 홍길동
>>> print(name, "님 안녕하세요")
홍길동 님 안녕하세요
```

정수 입력

- 입력된 문자열을 `int` 함수를 이용하여 정수로 변환한다.

```
score = int(input("첫 번째 점수: "))
```

프로그램 4.13 두 점수의 합 계산 3



```
>>> sum = 0
>>> score = int(input("첫 번째 점수: "))
첫 번째 점수: 88
>>> sum = sum + score
>>> score = int(input("두 번째 점수: "))
두 번째 점수: 92
>>> sum = sum + score
>>> print("두 점수의 합:", sum)
두 점수의 합: 180
```

실수 입력 예: 마일 킬로미터 변환

■ 실수 입력

- `input` 함수를 사용하여 마일 수를 입력받아
- 이를 `float` 함수를 이용하여 실수로 변환하여
- 이를 변수 `mile`에 저장

프로그램 4.14 마일 킬로미터 변환

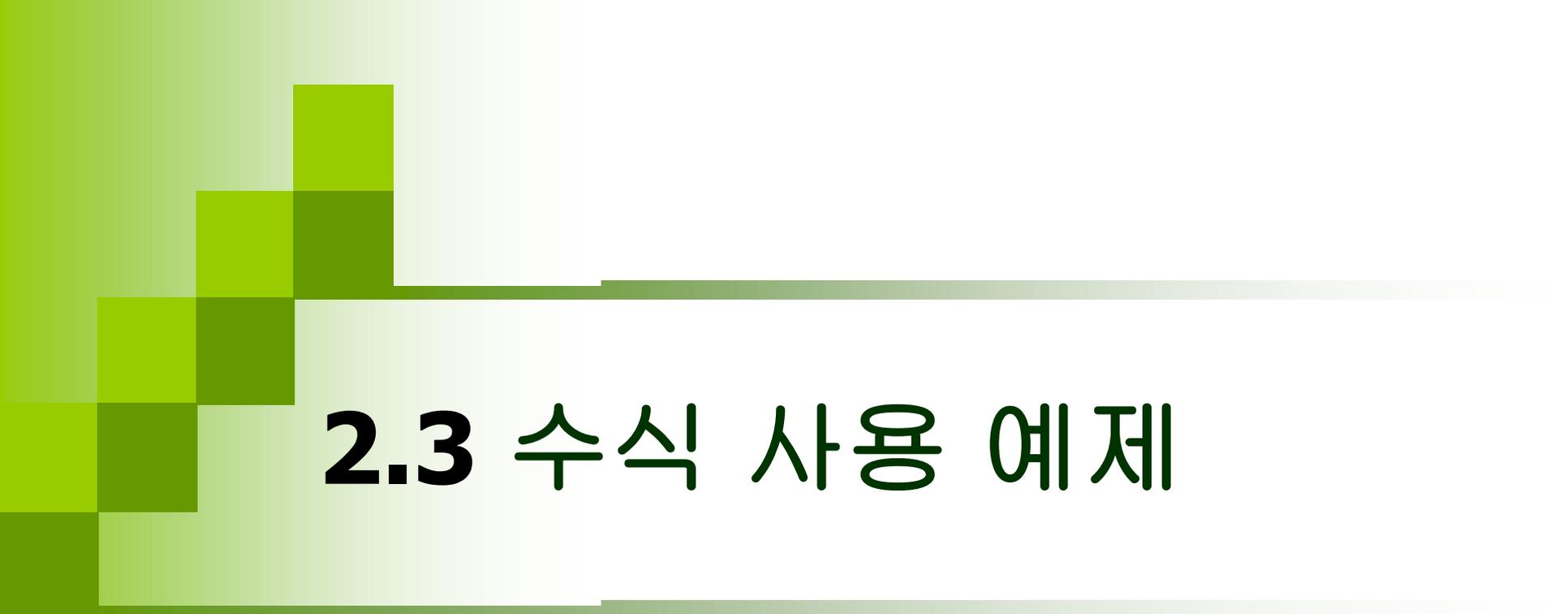


마일을 킬로미터로 변환하는 프로그램

```
mile = float(input("마일 입력: "))
kilo = 1.6 * mile
print(mile, "마일은", kilo, "킬로미터입니다.")
```

실행 결과

```
마일 입력: 61.5
61.5 마일은 98.4 킬로미터입니다.
```



2.3 수식 사용 예제

예제: 은행 계좌 입금

- 먼저 계좌 소유자 이름을 입력받고
- 사용자로부터 입금액을 입력받아
- 이를 잔액에 더한다.

프로그램 4.15 은행 계좌 입금



```
>>> balance = 0
>>> name = input("계좌 소유자 이름: ")
계좌 소유자 이름: 홍길동
>>> amount = int(input("입금액 입력: "))
입금액 입력: 50000
>>> balance = balance + amount
>>> print(name, "님의 현재 잔액:", balance)
홍길동 님의 현재 잔액: 50000
```

예제: 온도 변환

- 사용자로부터 화씨 온도를 입력받아
- 이를 섭씨 온도로 변환하여 출력한다.
- 온도 변환 공식 $^{\circ}\text{C} = 5/9 * (^{\circ}\text{F} - 32)$

프로그램 4.16 화씨 온도-섭씨 온도 변환



화씨 온도를 섭씨 온도로 계산하는 프로그램

```
fahr = float(input("화씨 온도 입력: "))  
celsius = 5 * (fahr - 32) / 9  
print("섭씨 온도:", celsius)
```

실행 결과

```
화씨 온도 입력: 68.0  
섭씨 온도: 20.0
```

math 모듈: 표 4.3

프로그램 4.19 삼각형 면적 계산



헤론의 공식을 이용한 삼각형 면적 계산 프로그램

```
import math
print("삼각형 세 변의 길이를 입력하세요")
a = int(input())
b = int(input())
c = int(input())
s = (a + b + c)/2
area = math.sqrt(s * (s-a) * (s-b) * (s-c))
print("삼각형 면적: ", area)
```

실행 결과

삼각형 세 변의 길이를 입력하세요

3

4

5

삼각형 면적: 6.0

- Math 모듈이 제공하는 수학 상수 및 함수 사용

```
import math
```



프로그래밍 실습

▶ 프로그래밍 실습

1. 은행계좌 예제 프로그램을 참고하여 출금하는 프로그램을 작성하시오. 사용자로부터 출금할 금액을 입력 받아 이를 현재 잔액에서 뺌으로써 출금할 수 있다.
2. 원-달러 환전 프로그램을 작성하시오. 입력으로 환율과 환전할 금액(원)을 받아 달러로 환전한 후에 그 금액을 출력한다.



3. 판단과 선택

if 문

■ if 문 형식

if 조건식:
문장1

- 조건식은 결과가 참/거짓인 식
 - 관계연산자: $>$, $<$, $>=$, $<=$, $==$, $!=$
 - 논리연산자: **and**, **or**, **not**
- 문장1은 들여쓰기 해야 하며,
여러 개의 문장이 있을 경우 들여쓰기에 일관성을 요함
- 조건이 참일 경우 문장1을 실행
거짓일 경우에는 문장1을 건너뛴

if 문

```
grade = float(input('성적 입력: '))  
if grade >= 4.0:  
    print('장학금을 받을 수 있습니다.')  
    print('축하합니다.')
```

```
>>>  
성적 입력: 4.1  
장학금을 받을 수 있습니다.  
축하합니다.
```

```
>>>  
성적 입력: 3.7  
>>>
```

if-else 문

■ if-else 문 형식

if 조건식:

문장1

else:

문장2

- 조건이 참일 경우 문장1을 실행
거짓일 경우에는 문장2를 실행
- 문장 1의 들여쓰기와 문장2의 들여쓰기는
일치하는 것이 바람직하다.

if-else 문

```
grade = float(input('성적 입력: '))
if grade >= 4.0:
    print('장학금을 받을 수 있습니다.')
    print('축하합니다.')
else:
    print('다음 학기에 ...')
print('안녕히 가세요.')
```

```
>>>
성적 입력: 3.8
다음 학기에 ...
안녕히 가세요.
```

elif (else if)

- 여러 개의 대등한 조건 중에서 선택할 때

if 조건식1:

문장1

elif 조건식2:

문장2

elif 조건식3:

문장3

...

else:

문장n

```
score = int(input("점수를 입력하세요: "))
if score < 0 or score > 100:
    print("잘못된 점수입니다")
elif score >= 90:
    print("A 학점")
elif score >= 80:
    print("B 학점")
elif score >= 70:
    print("C 학점 ")
elif score >= 60:
    print("D 학점")
else:
    print("F 학점")
```

- 참인 최초의 조건식k에 대해 문장k를 실행
참인 조건식이 없으면 문장n을 실행

▶ 프로그래밍 실습

- 근의 공식을 사용하여 형태의 2차 방정식의 근들을 구하여 출력하는 파이썬 프로그램을 작성하시오.
(`math` 모듈의 `sqrt()` 함수를 사용하여 제곱근 계산)

$$x^2 + bx + c = 0$$

```
>>>  
b = 2  
c = -4  
x1 = 1.2360679774997898  
x2 = -3.23606797749979
```