Simple Input/Output

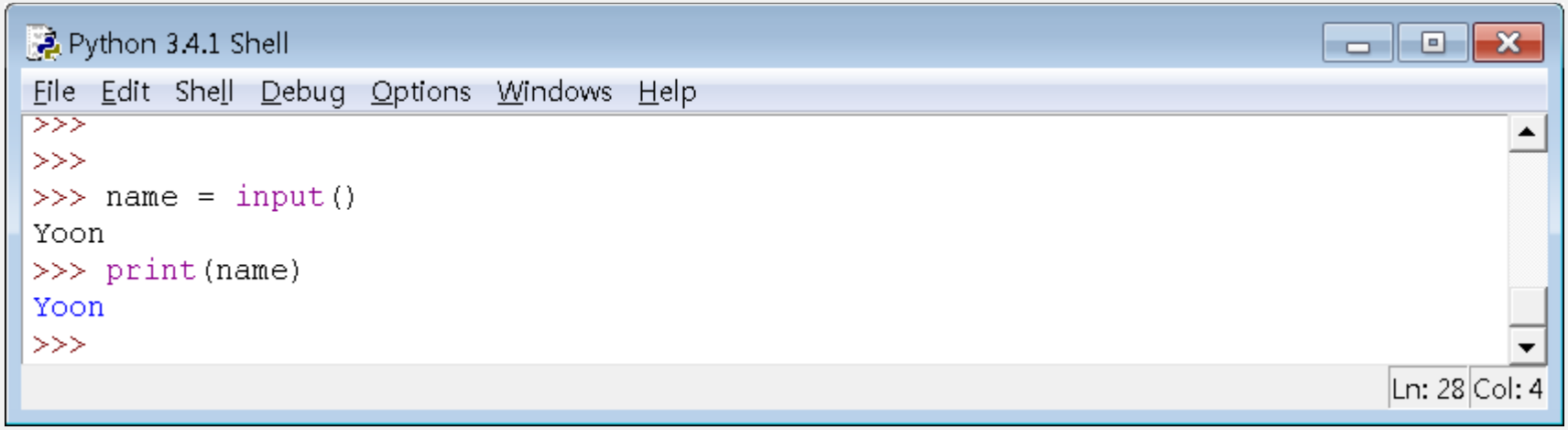
국민대 소프트웨어학부

수업목표

- `input()`
- print
- Inputting numbers
- Input from the Web

input()

- 사용자 입력 함수
  - 함수: int(), float(), type(), …
- 사용자로부터 스트링 입력



input()

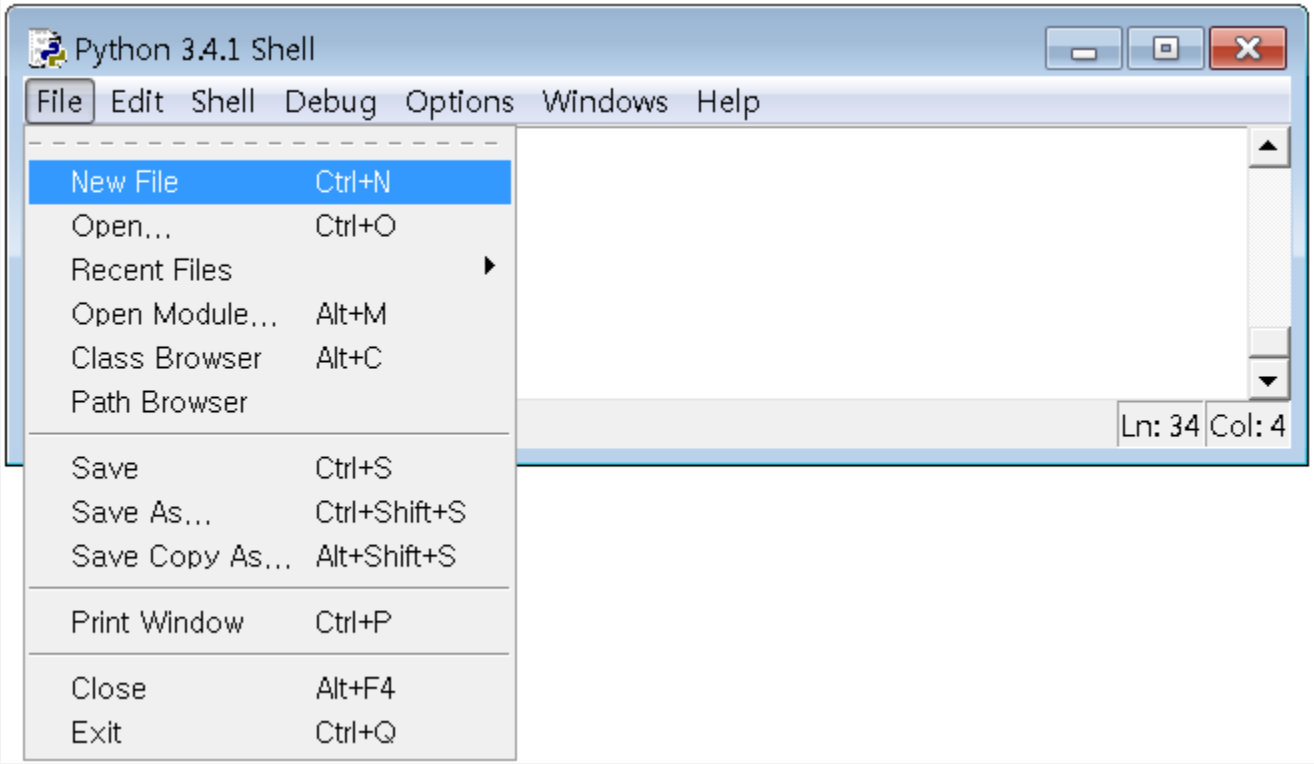
- input 함수
  - 명령 프롬프트에서 사용자로부터 입력 받는 함수
  - 사용자가 무엇을 입력해도 함수의 결과 값은 문자열(string)자료형

```
<입력 받을 변수> = input(<프롬프트 문자열>)
```

```
# 입력을 받습니다.
string = input("입력> ")

# 출력 합니다.
print(string)
```

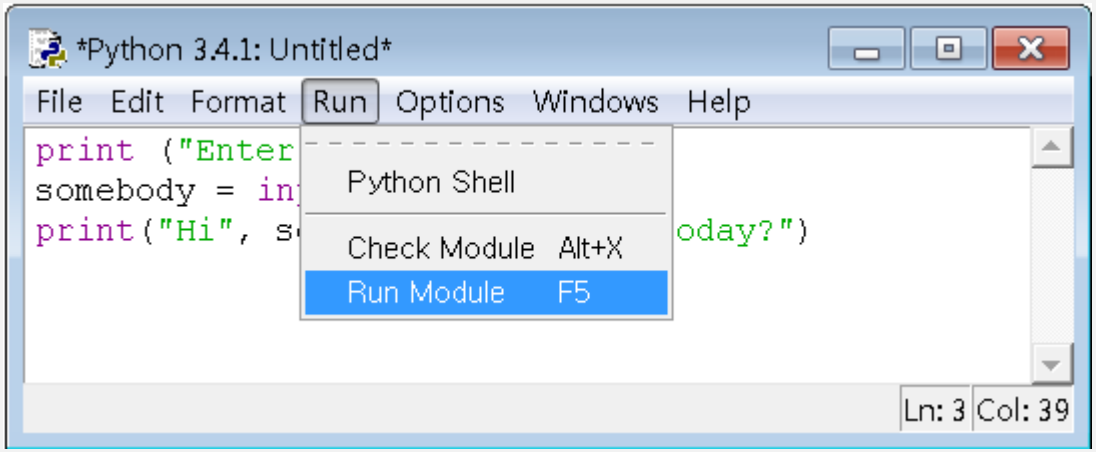
input()

- 프로그램 코드 파일 생성 및 저장
  - File → New File 또는 ctrl n
  - 코드 타이핑
  - 저장!



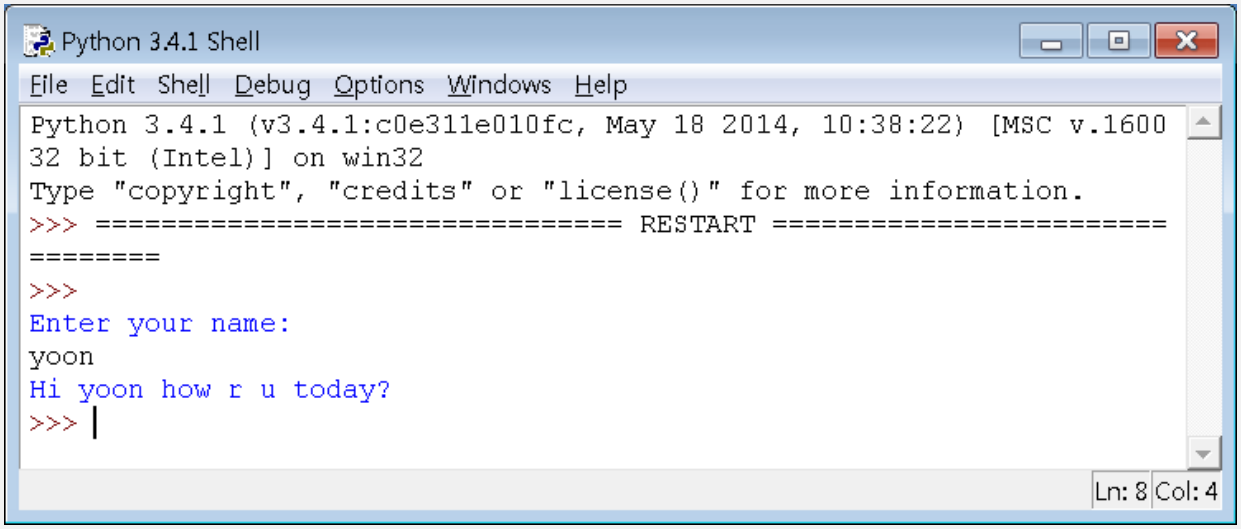
input()

- 프로그램 코드 파일 실행
  - Run → Run Module 또는 F5



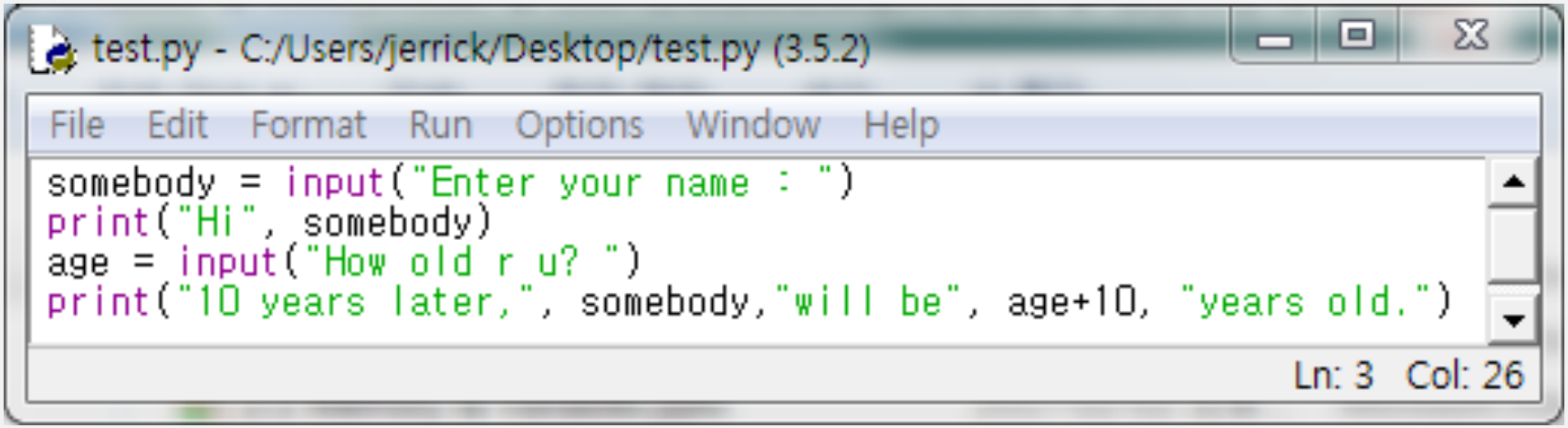
input()

- 프로그램 코드 파일 실행
  - Run → Run Module 또는 F5



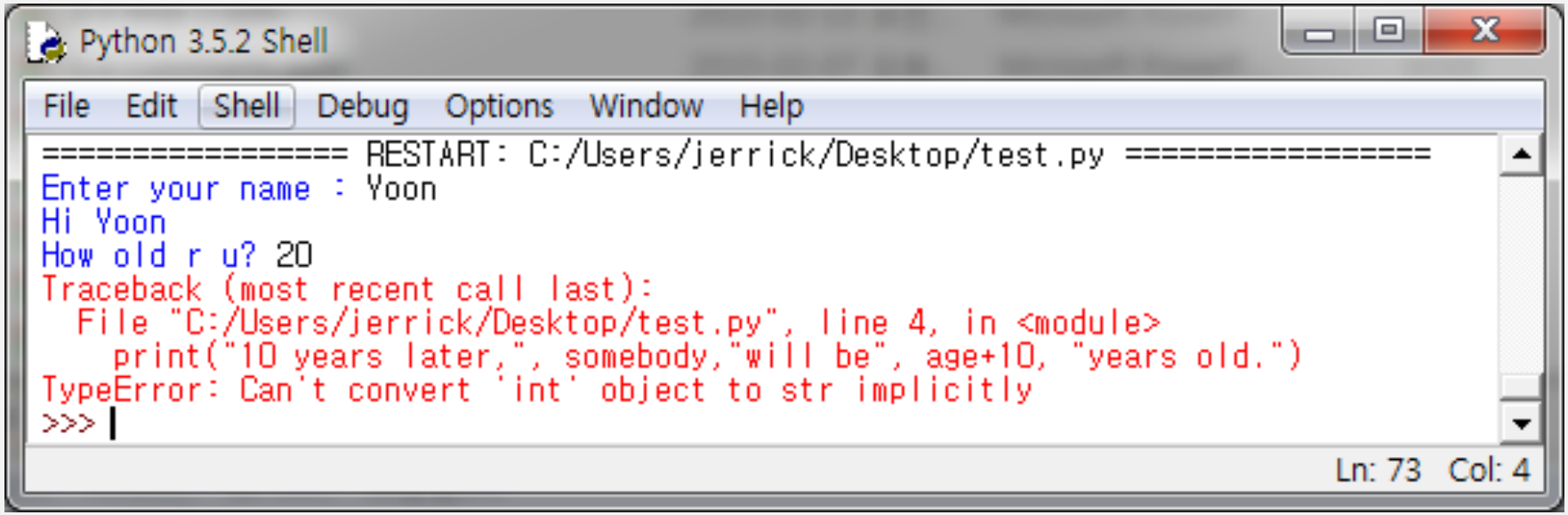
input()

- 사용자 입력 스트링의 숫자 변환
  - 스트링 + 숫자 → 에러 발생



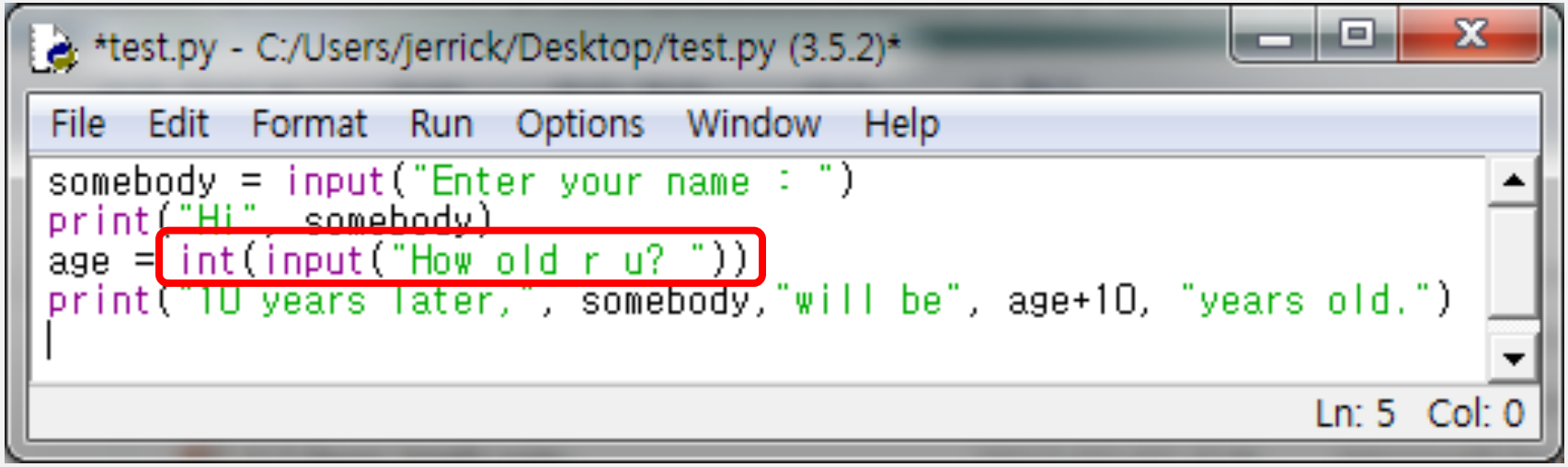
input()

- 사용자 입력 스트링의 숫자 변환
  - 스트링 + 숫자 → 에러 발생



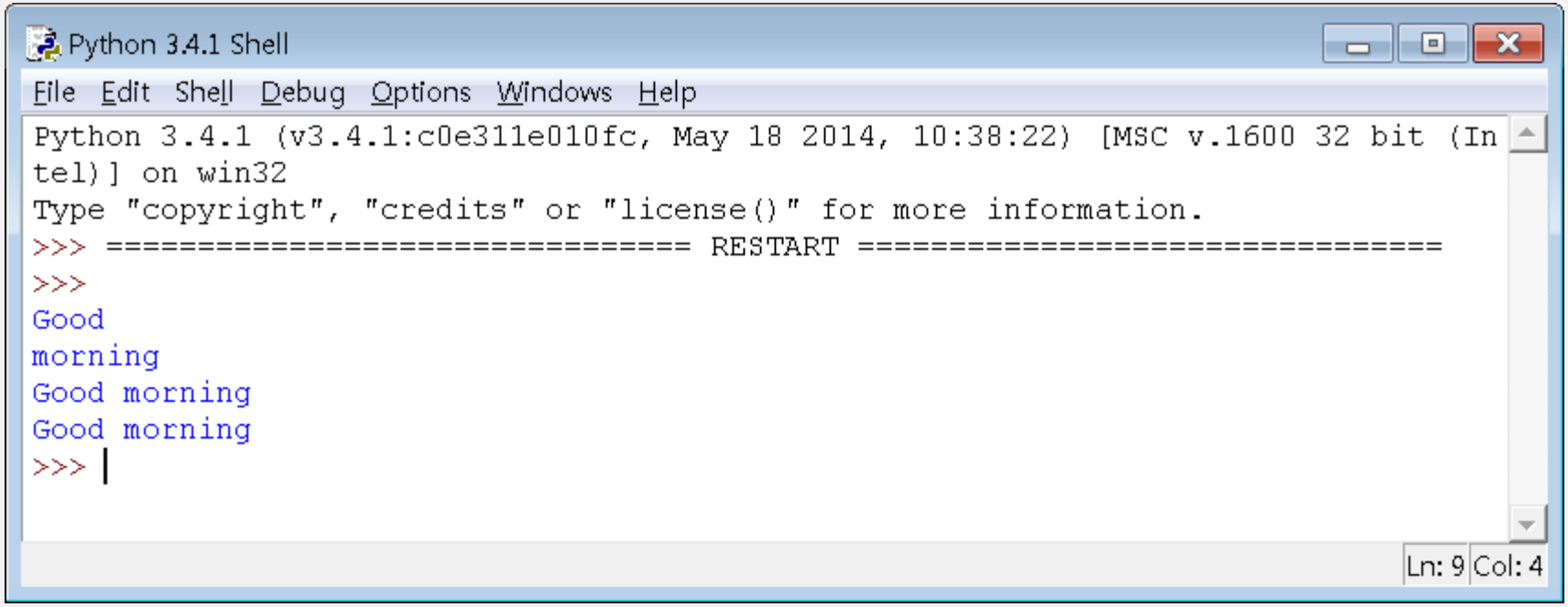
input()

- 사용자 입력 스트링의 숫자 변환
  - int(), float() 사용



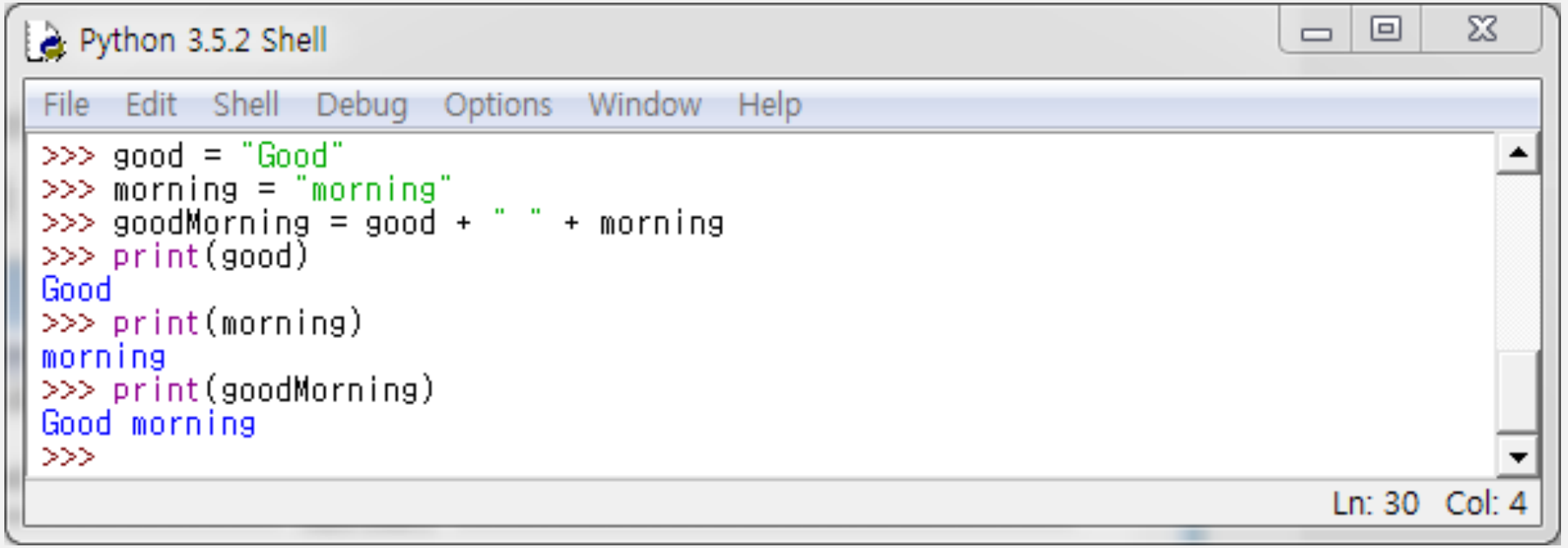
print

- '+' 연산
  - 스트링 연결 (string concatenation)
  - 숫자 합산 (add)



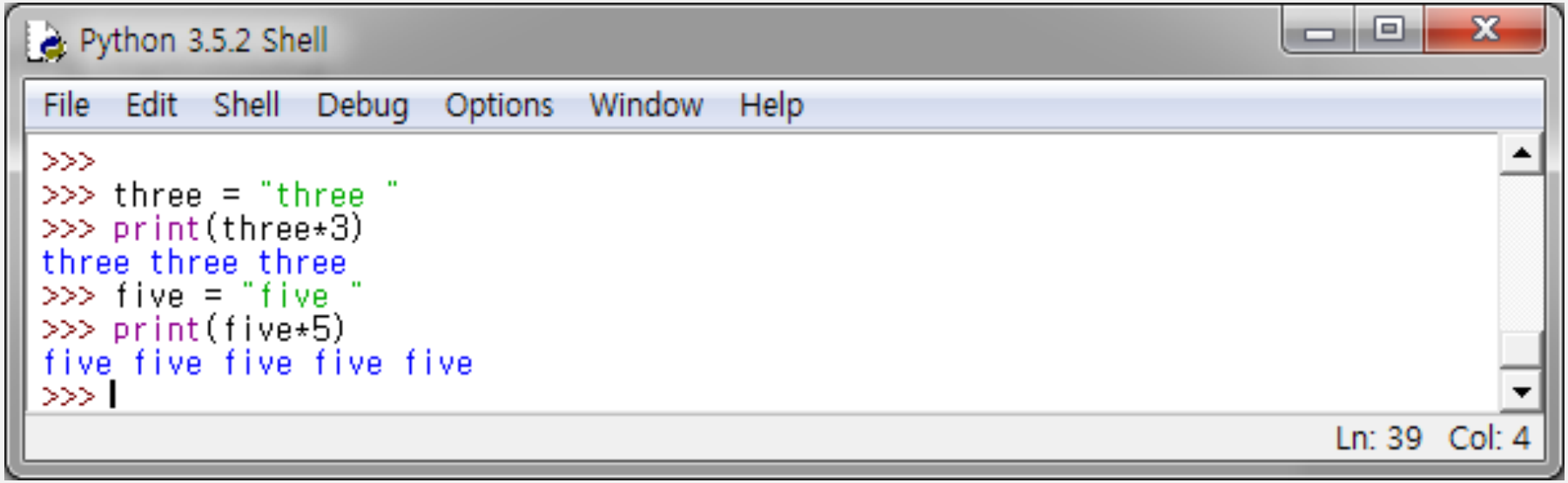
print

- '+' 연산
  - 스트링 연결 (string concatenation)
  - 숫자 합산 (add)



print

- '*' 연산
  - 스트링 반복 (string repetition)
  - 숫자 곱하기 (multiplication)



print

- 문자열 길이 구하기
  - len(<문자열>) : 문자열의 길이를 반환한다.

```
# len( <객체> ) : <객체>의 길이를 반환하는 함수
print(len("안녕하세요"))
```

print

- 스트링 출력
  - 특수문자
    - '\n' : 새로운 라인 (new line)
    - '\t' : 탭(tab)

Python 3.4.1: 2.4example5.py - C:/Users/mkyoon/Desktop/2.4example5.py

File Edit Format Run Options Windows Help

```
print("Good\n" + "morning")
print ("name address")
print ("name\taddress")
```

Ln: 1 Col: 18

Python 3.4.1 Shell

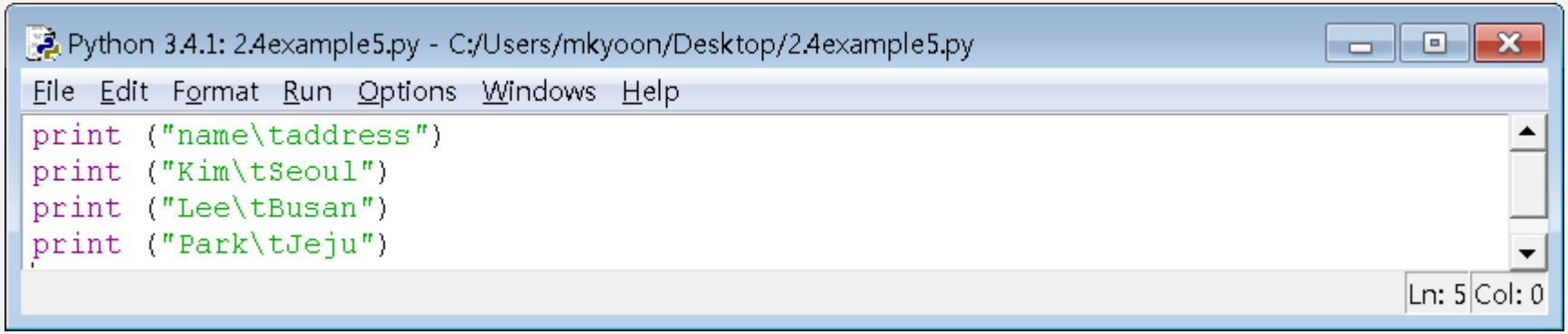
File Edit Shell Debug Options Windows Help

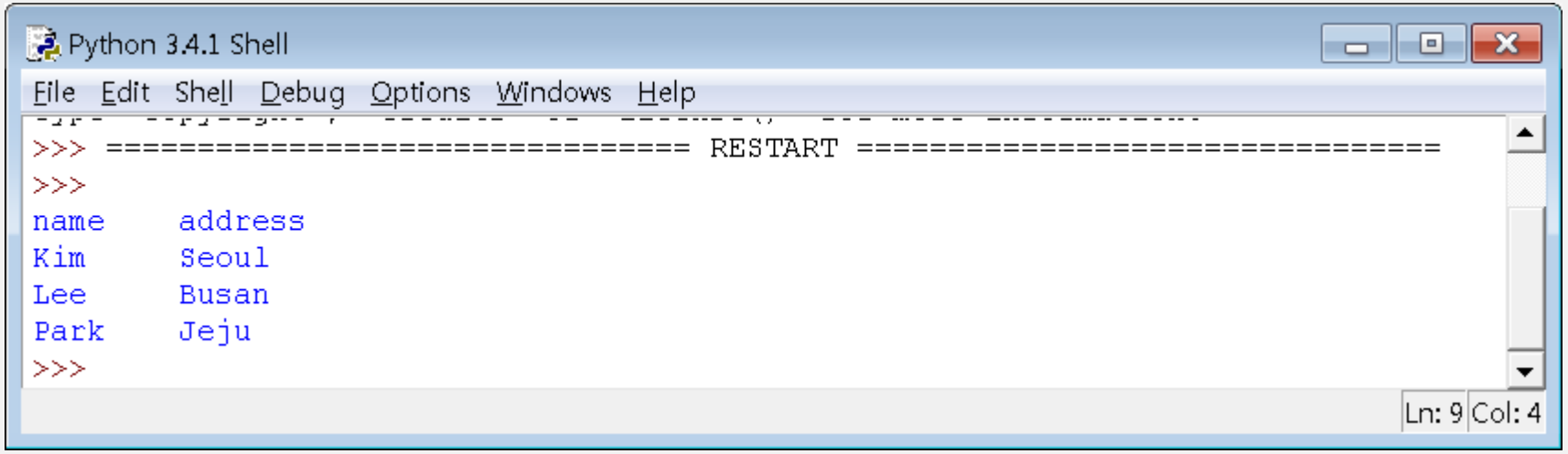
```
>>> ================================ RESTART ================================
>>>
Good
morning
name address
name    address
>>>
```

Ln: 9 Col: 4

print

- 스트링 출력
  - 특수문자
    - '\n' : 새로운 라인 (new line)
    - '\t' : 탭(tab)





print

- 스트링 출력
  - 이스케이프 문자

확장문자	내용
\’	단일 인용문자
\”	이중 인용문자
\\	역슬래시(back slash)문자
\a	Beep 소리문자
\b	백스페이스
\f	Form-feed
\n	개행문자(줄바꿈)
\r	Carrige return
\t	Tab (수평)
\v	Tab (수직)
\000	Null

print

- 스트링 출력
  - 이스케이프 문자

```
print("성민이는 ₩"배가고프다₩" 라고 말했습니다.")
print("햄버거₩t피자₩t치킨")
print("₩₩ ₩₩ ₩₩ ₩₩ ₩₩")
```

print

- 여러 줄 문자열
  - 큰따옴표(또는 작은따옴표)를 3번 반복해서 사용하면 여러 줄 문자열을 표현 할 수 있다.
    - ₩ 를 하고 줄을 바꾸면, 줄 연속을 의미한다.

```
multiline = """₩
Don₩'t let the noise of other₩'s opinions drown out your inner voice.
And most important, have the courage to follow your heart and intuition.
They somehow already know what you truly want to become.
Everything else is secondary.
"""

print(multiline)
```

https://news.stanford.edu/2005/06/14/jobs-061505/

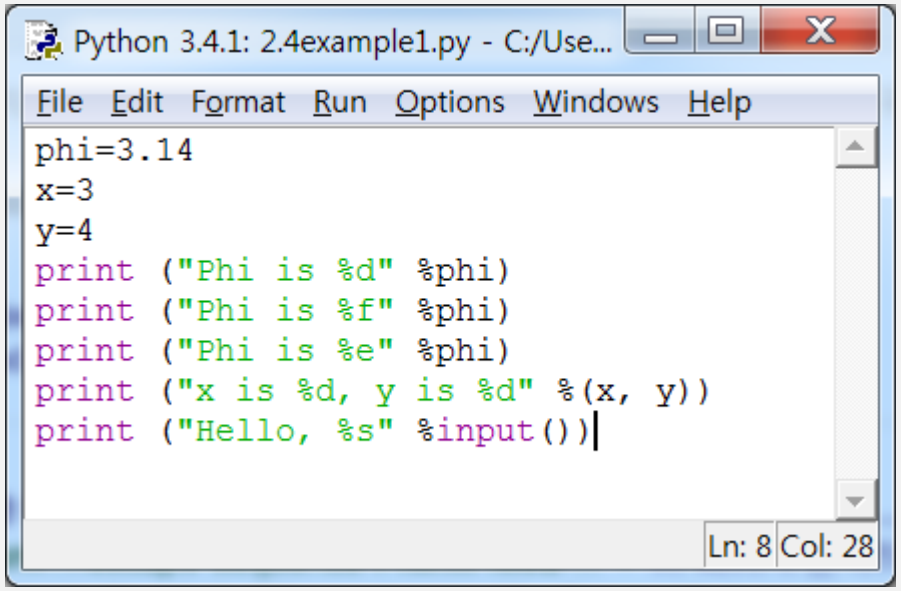
print

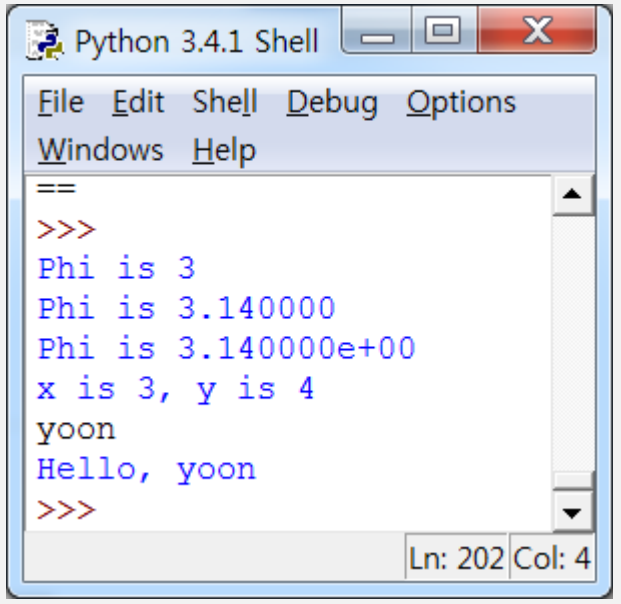
- 스트링 출력 포맷

포맷	의미
%s	String(문자열)
%c	Character(문자)
%d	Decimal(10진수 정수)
%f	Floating-point(부동소수)
%o	Octal(8진수 정수)
%x	heXa(16진수 정수. 소문자표시)
%X	heXa(16진수 정수. 대문자표시)
%e	과학적 수치(소문자표시)
%E	과학적 수치(대문자표시)

print

- 스트링 출력 포맷





print

- 스트링 출력 포맷

```
# 문자열
print("I have %s apples." %"five")

# 문자 1개
print("My grade is %c." %'A')

# 정수
print("I have %d apples." %3)

# 부동소수 및 %
print("Error is %f%%" %3.234)

# 8진수, 16진수
print("%o %x %X" %(9, 10, 255))
```

print

- format 메소드
  - '{}' 형태의 기호를 포함한 문자열 뒤에 '.format()' 을 붙여서 사용

```
# 문자열
print("I have {} apples.".format("five"))

# 문자 1개
print("My grade is {}.".format('A'))

# 정수
print("I have {} apples.".format(3))

# 부동소수 및 %
print("Error is {}%%".format(3.234))

# 8진수, 16진수
print("{} {} {}".format(9, 10, 255))
```

https://docs.python.org/3/library/stdtypes.html?highlight=format#str.format

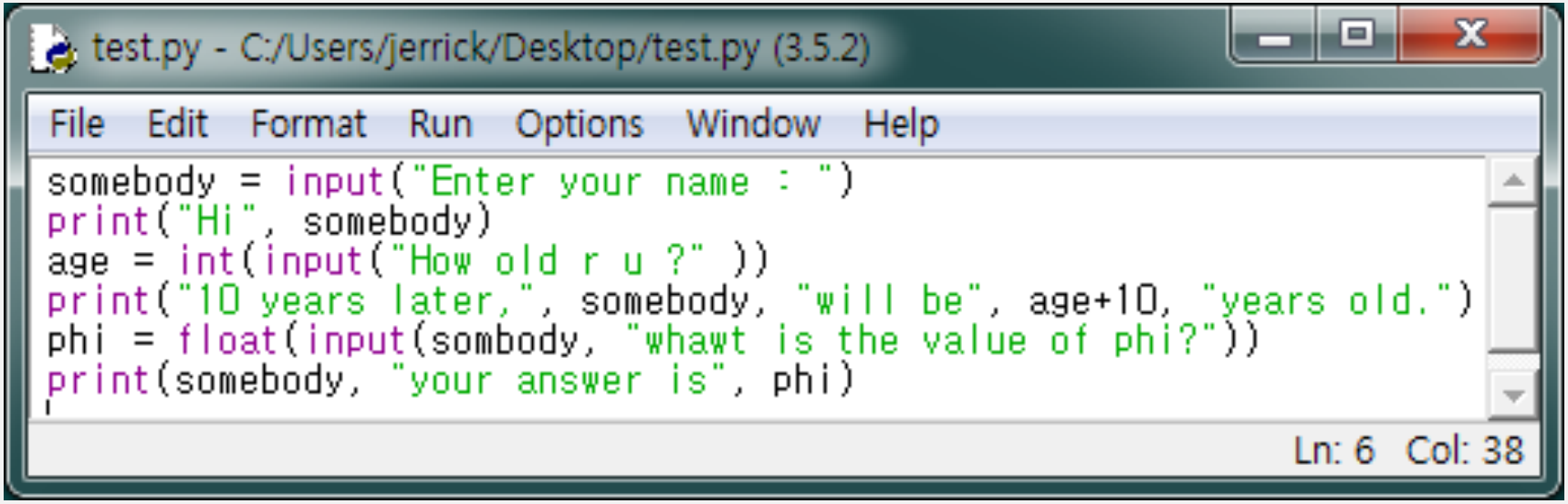
print

- 선택 연산자([<숫자>])
- 범위 선택 연산자([<숫자>:<숫자>]

```
# 연결 연산자(Concatenation)
print("안녕하세요" + "...!")

# 반복 연산자
print("안녕하세요" * 3)

# 선택 연산자(Indexing)
print("안녕하세요"[0])
print("안녕하세요"[1])
print("안녕하세요"[2])
print("안녕하세요"[3])
print("안녕하세요"[4])

# 범위 선택 연산자(Slicing)
print("안녕하세요"[0:2])
print("안녕하세요"[1:3])
print("안녕하세요"[2:4])
print("안녕하세요"[:3])
print("안녕하세요"[3:])
```

Inputting numbers

- String vs int vs float
  - 문자열 age를 정수로 변환하여 저장.
  - 문자열 phi를 실수로 변환하여 저장.



Inputting numbers

- 문자열을 숫자로 바꾸기
  - int 함수 또는 float 함수를 이용해서 바꾼다.
- 숫자를 문자열로 바꾸기
  - str 함수를 이용한다.

```
# 문자열 끼리 덧셈을 합니다.
string1 = str(123)
string2 = str(456)
print("string1 + string2:", string1 + string2)

# 정수 끼리 덧셈을 합니다.
number1 = int("123")
number2 = int("456")
print("number1 + number2:", number1 + number2)
```

Inputting numbers

- 실습
  - 사용자로부터 두 자연수 A, B를 입력 받고, 두 자연수의 사칙연산 결과를 출력하는 코드를 작성 하세요.

```
A : 5
B : 2
A+B : 7
A-B : 3
A*B : 10
A/B : 2.5
```

Inputting numbers

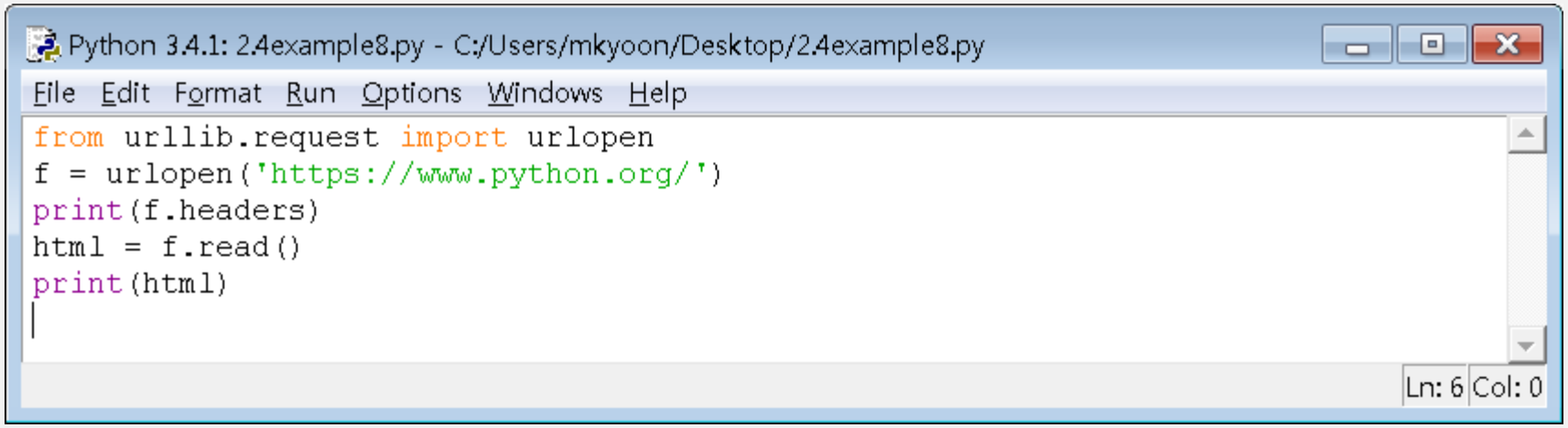
- 실습
  - 사용자로부터 두 자연수 A, B를 입력 받고, 두 자연수의 사칙연산 결과를 출력하는 코드를 작성 하세요.

```
A : 5
B : 2
A+B : 7
A-B : 3
A*B : 10
A/B : 2.5
```

```
# 사용자로부터 두 자연수를 입력 받는다.
A = int(input("A : "))
B = int(input("B : "))

# 사칙 연산의 결과를 출력 합니다.
print("A+B :", A+B)
print("A-B :", A-B)
print("A*B :", A*B)
print("A/B :", A/B)
```

Input from the web

- WWW(World Wide Web)으로부터의 입력
  - Html(Hyper-text markup language) 파일
  - “from urllib.request import urlopen”
    - from, import: 모듈 사용 명령어
    - urllib.request, urlopen: 사용 모듈
      - url: uniform resource locator



실습

- 난수 생성 및 출력
  - random 모듈
    - import random 사용 (모듈에 대해서는 뒤에서 설명)
    - random.randint(a,b) : a이상 b이하인 숫자 임의 생성

Python 3.4.1: 2.4example2.py - C:/Users/mkyoon/Desktop...

File Edit Format Run Options Windows Help

```
import random

i=1
print("Random #%d: %d" %(i, random.randint(1,6)))
i+=1
print("Random #%d: %d" %(i, random.randint(1,6)))
i+=1
print("Random #%d: %d" %(i, random.randint(1,6)))
i+=1
print("Random #%d: %d" %(i, random.randint(1,6)))
i+=1
print("Random #%d: %d" %(i, random.randint(1,6)))
```

Ln: 1 Col: 0

Python 3.4.1 Shell

File Edit Shell Debug Options Windows Help

```
==========
==
>>>
Random #1: 5
Random #2: 2
Random #3: 5
Random #4: 3
Random #5: 3
>>>
```

Ln: 264 Col: 4

숙제

- 로또
  - 1~45까지 숫자 중 임의로 5개의 숫자를 출력하는 프로그램을 작성하시오. 단, 모든 숫자가 선택될 확률은 동일해야 하며, 뽑히는 순서는 상관없다.