

# 정보데이터와 출력

오 동 인

경희대학교 의과대학 의공학교실

# 문자 데이터 출력하기

- 문자열
  - 큰따옴표(" ")나 작은따옴표(' ')로 감싼 문자의 집합



- 올바른 예

"ABC" "abc" "123" "12.3" "@ # @" "파이썬" "Python"

- 틀린 예

```
>>> x = "AB
SyntaxError: EOL while scanning string literal
>>> x = "AB'
SyntaxError: EOL while scanning string literal
>>> x = 'AB
SyntaxError: EOL while scanning string literal
>>> x = 'AB"
SyntaxError: EOL while scanning string literal
>>> x = ""AB""
SyntaxError: invalid syntax
>>> x = """"AB""""
SyntaxError: EOL while scanning string literal
```

# 문자열 만드는 방법

- 큰따옴표로 양쪽 둘러싸기
  - "ABC", "A's C"
- 작은따옴표로 양쪽 둘러싸기
  - 'ABC', 'A"BC'
- 큰따옴표 3개를 연속으로 써서 양쪽 둘러싸기
  - """ABC"""
- 작은따옴표 3개를 연속으로 써서 양쪽 둘러싸기
  - '''ABC'''
- 문자열 출력
  - print() 함수 인수에 출력하려는 문자열 값을 전달하여 실행

>>> print('Python')

출력 결과

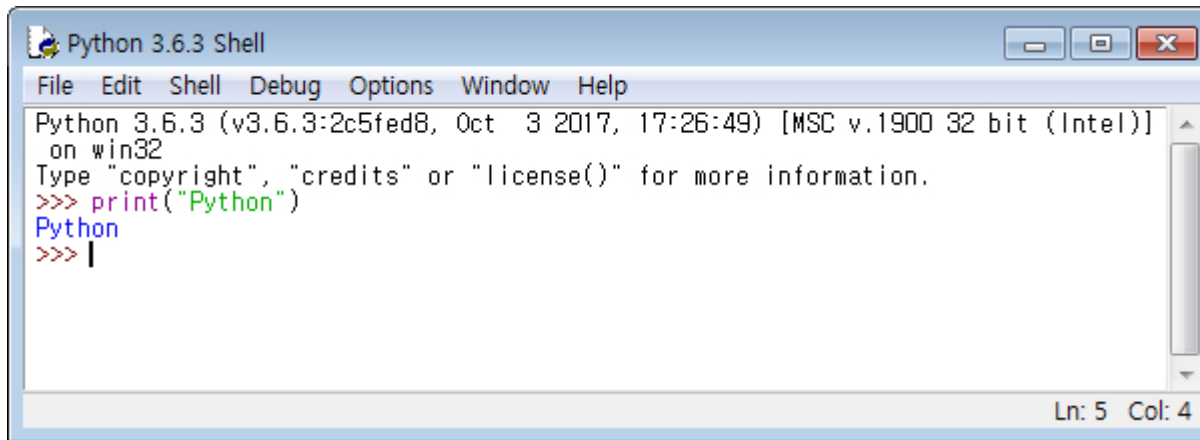
인수

"Python"  
문자열

```
print(인수1, 인수2, 인수3)
print(값1, 값2, sep='문자 또는 문자열')
print(값1, end='문자 또는 문자열')
\n은 값을 다음 줄에 출력하게 만드는 제어 문자
\t는 tab, \\ ('\ ' 출력)
```

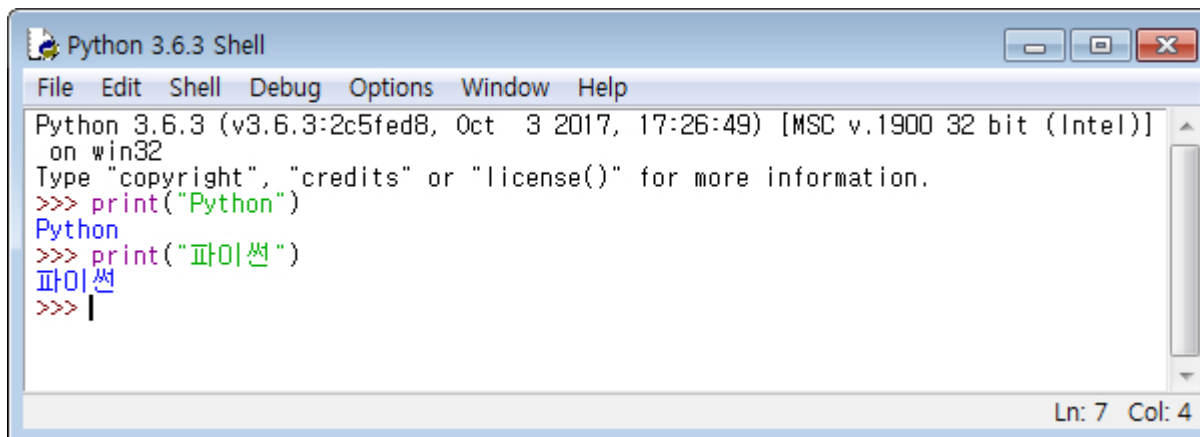
# 문자 데이터 출력하기

- Ex) 다음 문자열을 출력하시오.



```
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Python")
Python
>>> |
```

Ln: 5 Col: 4



```
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print("Python")
Python
>>> print("파이썬")
파이썬
>>> |
```

Ln: 7 Col: 4

`print("Python")`

`print("파이썬")`

`print("안녕 Python")`

`print("안녕"+"파이썬")`

`print("안녕 "+"파이썬")`

`print("안녕"+" "+"파이썬")`

`print("10"+"20")`

`print("10"+20)`

`print("파이썬 "*10)`

# 숫자 데이터 출력하기

- 숫자
  - 소수점의 유무에 따라 정수, 실수로 구분

```
1  0  -1  100  1234567890
1.2 -1.2  0.123456789
```

- 숫자 출력
  - print() 함수 인수에 콤마로 구분된 값을 전달하여 실행

`>>> print(1, 0, -1, 100, 1234567890)`  
 인수      정수들을 콤마(,)로 구분

`1 0 -1 100 1234567890`  
 출력 결과

# 숫자 데이터 출력하기

- Ex) 다음 정수와 실수를 각각 출력하시오.

# 숫자 출력

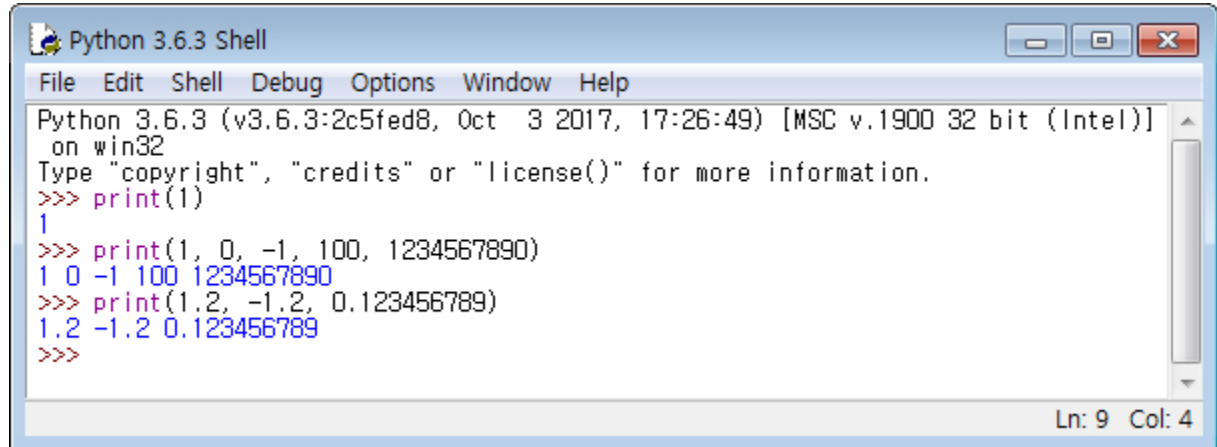
```
print(1)
```

```
print(1, 0, -1, 100, 1234567890)
```

```
print(1.2, -1.2, 0.123456789)
```

```
print(12345678901234567890, -12345678901234567890)
```

```
print(0.1234567890123456789)
```



The screenshot shows a Python 3.6.3 Shell window with the following content:

```
Python 3.6.3 (v3.6.3:2c5fed8, Oct 3 2017, 17:26:49) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print(1)
1
>>> print(1, 0, -1, 100, 1234567890)
1 0 -1 100 1234567890
>>> print(1.2, -1.2, 0.123456789)
1.2 -1.2 0.123456789
>>>
```

The status bar at the bottom right indicates "Ln: 9 Col: 4".

# 숫자 데이터 출력하기

- 수식

- 연산자, 피연산자로 구성
- 산술 연산, 관계 연산, 논리 연산, 비트 연산 등 처리
- 산술연산자와 수식

연산	연산자	수식	결과
덧셈	+	$6 + 4$	10
뺄셈	-	$6 - 4$	2
곱셈	*	$6 * 4$	24
나눗셈	/	$6 / 4$	1.5

• 수식의 계산 결과 출력

- print() 함수 인수에 수식을 전달하면 계산 결과 출력

`>>> print( 6 + 4, 6 - 4, 6 * 4, 6 / 4 )`  
 인수 수식  
 ↓  
 10 2 24 1.5  
 출력 결과

# 연산자

연산자	기능	문법	설명
+	덧셈	$a + b$	두 값을 더함
-	뺄셈	$a - b$	a에서 b를 뺌
*	곱셈	$a * b$	두 값을 곱함
/	나눗셈	$a / b$	a에서 b를 나누며 결과는 실수
//	버림 나눗셈 (floor division)	$a // b$	a에서 b를 나누며 소수점 이하는 버림
%	나머지	$a \% b$	a에서 b를 나누었을 때 나머지를 구함
**	거듭제곱	$a ** b$	a를 b번 곱함
@	행렬 곱셈	$a @ b$	행렬 a와 b를 곱함
+	양수 부호	$+a$	a에 양수 부호를 붙임
-	음수 부호	$-a$	a에 음수 부호를 붙임
+=	덧셈 후 할당	$a += b$	a와 b를 더한 후 결과를 a에 할당
-=	뺄셈 후 할당	$a -= b$	a에서 b를 뺀 후 결과를 a에 할당
*=	곱셈 후 할당	$a *= b$	a와 b를 곱한 후 결과를 a에 할당
/=	나눗셈 후 할당	$a /= b$	a에서 b를 나눈 후 결과를 a에 할당(결과는 실수)
//=	버림 나눗셈 후 할당	$a //= b$	a에서 b를 나눈 후 결과를 a에 할당(소수점 이하는 버림)
%=	나머지 연산 후 할당	$a \% = b$	a에서 b를 나누었을 때 나머지를 구하여 a에 할당
**=	거듭제곱 후 할당	$a ** = b$	a를 b번 곱한 후 결과를 a에 할당
@=	행렬 곱셈 후 할당	$a @ = b$	행렬 a와 b를 곱한 후 결과를 a에 할당



# 숫자 데이터 출력하기

- Ex) 다음 연산을 출력하시오.

# 연산 출력

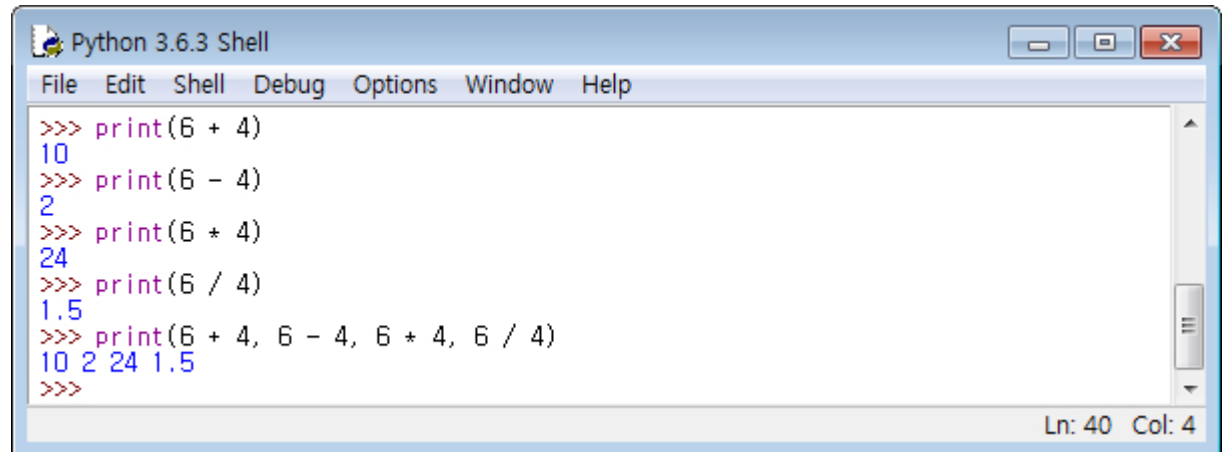
```
print(6+4)
print(6-4)
print(6*4)
print(6/4)
print(6+4, 6-4, 6*4, 6/4)
```

```
print(6//4)
print(6%4)
```

```
print(1+2+3+4+5)
```

```
print((5+10+15)/3)
```

```
print("6+4 =", 6+4)
```



```
Python 3.6.3 Shell
File Edit Shell Debug Options Window Help
>>> print(6 + 4)
10
>>> print(6 - 4)
2
>>> print(6 * 4)
24
>>> print(6 / 4)
1.5
>>> print(6 + 4, 6 - 4, 6 * 4, 6 / 4)
10 2 24 1.5
>>>
```

# EOD