



4장 자료의 종류에는 어떤 것들이 있나요?

본 강의 자료는 생능출판사 두근두근 파이썬(천인국 저)의 강의 자료를 기반으로 제작되었습니다.

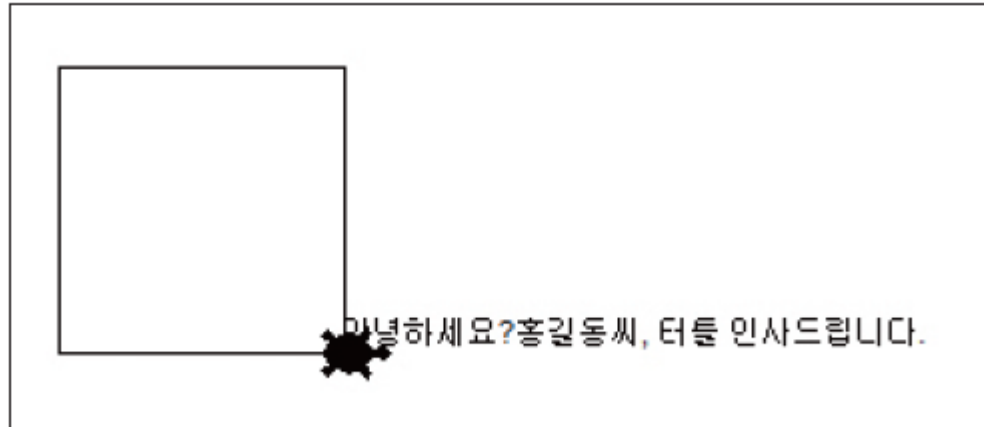
한국항공대학교 안준선

이번 장에서 배울 내용

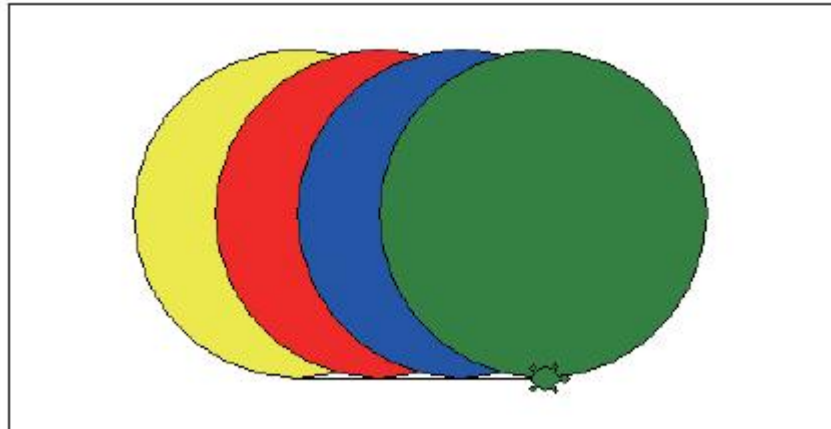
- ▶ 파이썬의 기본적인 자료형(**types**)
- ▶ 문자열 내 맘대로 사용하기
 - ▣ % 연산자
 - ▣ 파이썬의 강력한 기능 : **인덱스 []**
- ▶ 리스트 자료형

이번 장에서 만들 프로그램

(1) 터틀 그래픽의 거북이와 인사하는 프로그램을 작성해 보자.



(2) 여러 개의 색상을 **리스트**에 저장하였다가 하나씩 꺼내서 원들을 그려 보자



파이썬에서 사용할 수 있는 자료형

분류	종류	값
진리값	boolean	True, False
숫자	int	... -2, -1, 0, 1, 2, 3, 4, ...
	float	3.14, 2.33e12
	complex	3+2j, 5-2j
문자열	string	"abc", 'Hello', "%", "!!!"
열(sequences)	list, ...	[1,2,3,4]
...		



파이썬과 자료형

- ▶ 변수에 어떤 종류의 자료도 저장할 수 있다.
(파이썬에서 변수는 가리키기 때문에 가능)

```
x = 10  
print("x =", x)  
x = 3.14  
print("x =", x)  
x = "Hello World!"  
print("x =", x)
```

```
x = 10  
x = 3.14  
x = Hello World!
```

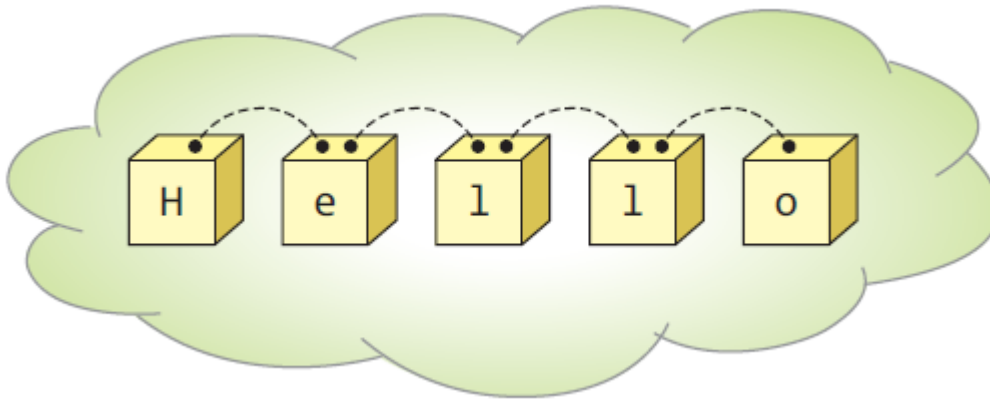
문자열

- ▶ 컴퓨터에게는 숫자가 중요하지만 인간에게는 텍스트 (text)가 중요하다.
- ▶ (예) 문자 메시지, 인터넷 도메인 이름
- ▶ 컴퓨터를 이용한 텍스트의 처리도 무척 중요하다.



문자열

- ▶ 문자열(string)은 문자들의 나열(sequence of characters)이다.



문자열을 만드는 방법

- ▶ 큰따옴표
- ▶ 작은 따옴표

```
>>> "Hello"  
'Hello'
```

```
>>> msg = "Hello"  
>>> msg  
'Hello'  
>>> print(msg)  
Hello
```


문법적인 오류

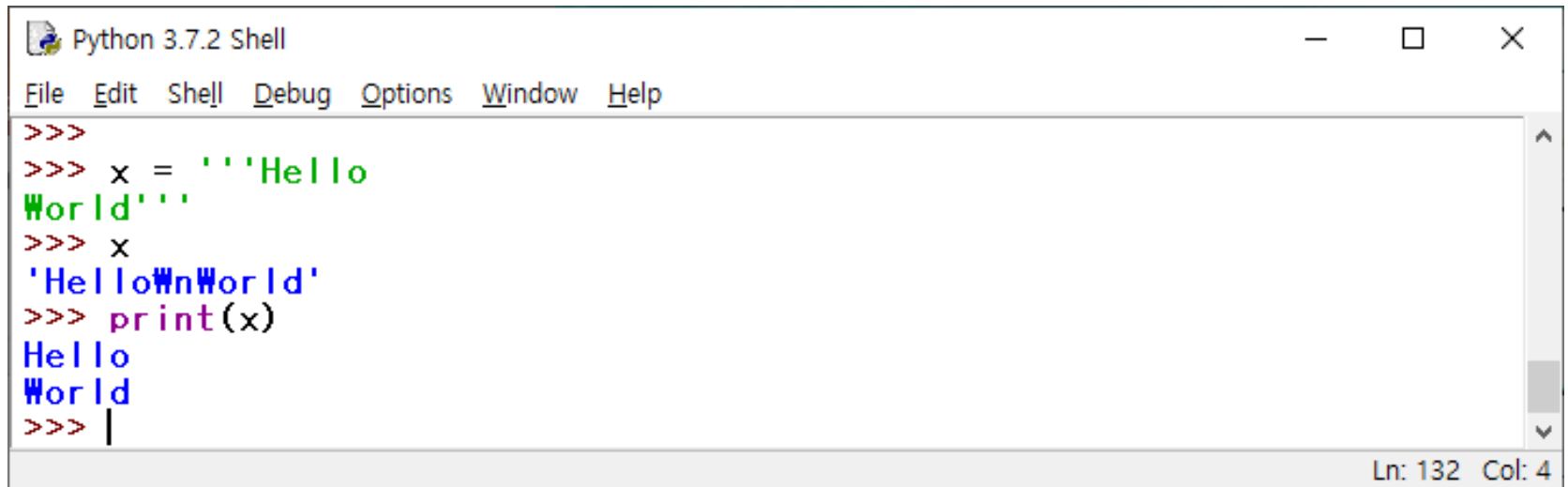
- ▶ 큰따옴표(“)로 시작했다가 작은따옴표(‘)로 끝내면 문법적인 오류이다.

```
>>> msg = "Hello'  
SyntaxError: EOL while scanning string literal
```



여러줄짜리 문자열 만들기

- ▶ 작은 따옴표 세 개(''')를 사용하여 여러 줄 문자를 만들 수 있다.

A screenshot of a Python 3.7.2 Shell window. The window has a title bar with the text 'Python 3.7.2 Shell' and standard window controls (minimize, maximize, close). Below the title bar is a menu bar with the following items: File, Edit, Shell, Debug, Options, Window, and Help. The main area of the window is a text editor showing a Python script. The script consists of four lines of code: a prompt followed by a blank line, a line defining a variable 'x' as a multi-line string 'Hello\nWorld', a line printing the variable 'x', and a line defining a variable 'x' as a multi-line string 'Hello\nWorld'. The output of the script is displayed below the code: 'Hello\nWorld'. The status bar at the bottom right of the window shows 'Ln: 132 Col: 4'.

```
>>>
>>> x = '''Hello
World'''
>>> x
'Hello\nWorld'
>>> print(x)
Hello
World
>>> |
```

Ln: 132 Col: 4

100과 "100"의 차이

- ▶ 100 -> 정수
- ▶ "100", '100' -> 문자열

```
>>> print(100+200)
300
>>> print("100"+"200")
100200
```

100+200을 하면 (정수+정수) 형태가 되어서 덧셈이 가능하다. 하지만 "100"+"200"은 텍스트와 텍스트끼리 합하는 것이기 때문에 그냥 2개의 텍스트가 붙어 버린다.

문자열 -> 숫자

- ▶ `int()`: 문자열을 정수로 변환
- ▶ `float()`: 문자열을 실수로 변환

```
t = input("정수를 입력하시오: ")  
x = int(t)  
t = input("정수를 입력하시오: ")  
y = int(t)  
print(x+y)
```

```
정수를 입력하시오: 100  
정수를 입력하시오: 200  
300
```

숫자->문자열

- ▶ 다음 코드에 오류가 발생하는 이유는?

```
>>> print('나는 현재 ' + 21 + '살이다.')
```

Traceback (most recent call last):

File "<pyshell#1>", line 1, in <module>

```
print('나는 현재 ' + 21 + '살이다.')
```

TypeError: Can't convert 'int' object to str implicitly

문자열과 숫자를 합칠 수 없는 의미입니다.

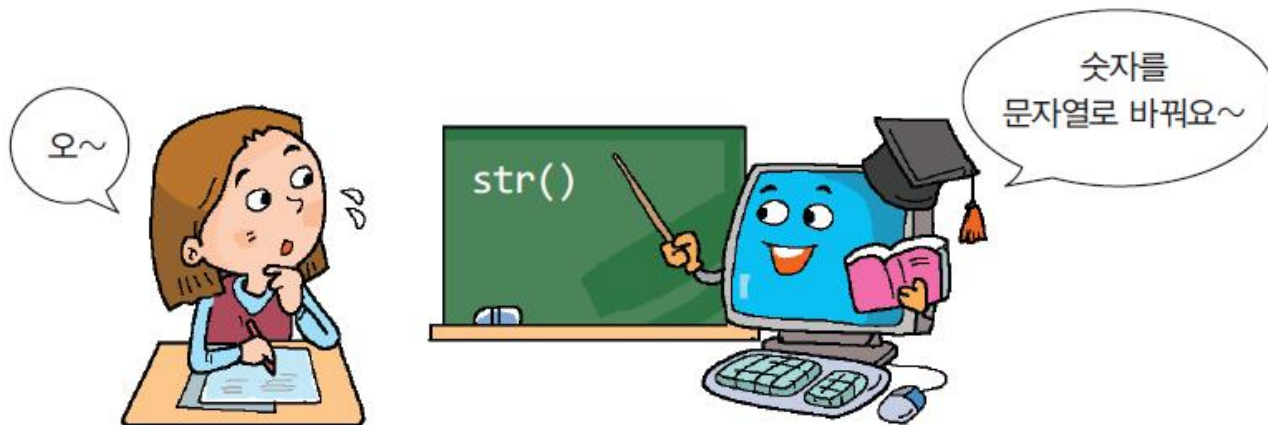


숫자->문자열

▶ str() 함수 사용

```
>>> print('나는 현재 ' + str(21) + '살이다.')  
나는 현재 21살이다.
```

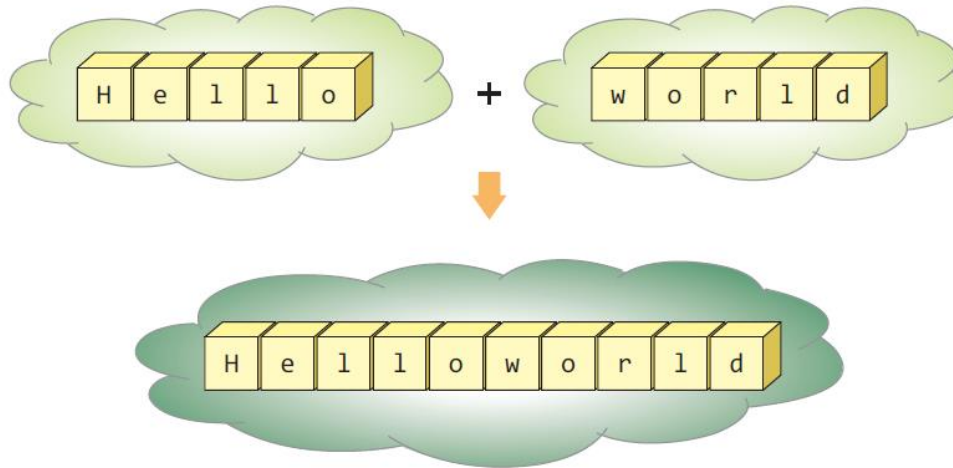
```
>>> print('원주율은 ' + str(3.14) + '입니다.')  
원주율은 3.14입니다.
```



문자열 접합

- ▶ 2개의 문자열을 합치려면 \rightarrow + 연산자

```
>>> 'Hello ' + 'World!'  
'Hello World!'
```



- ▶ 그냥 연속해서 써도 된다.

```
>>> 'Hello ' 'World' '!!!'  
'Hello World!!!'
```

문자열 반복

- ▶ 문자열을 반복하려면 -> * 연산자

```
>>> message = " Congratulations!"  
>>> print(message*3)  
Congratulations!Congratulations!Congratulations!
```

```
>>> print("="*50)  
=====
```


문자열에 변수값 포함

- ▶ 문자열에 변수의 값을 삽입하여 출력하고 싶으면 % 연산자 사용

```
>>> price = 10000
>>> print("상품의 가격은 %s원입니다." % price)
상품의 가격은 10000원입니다.
```

문자열에 변수값 포함

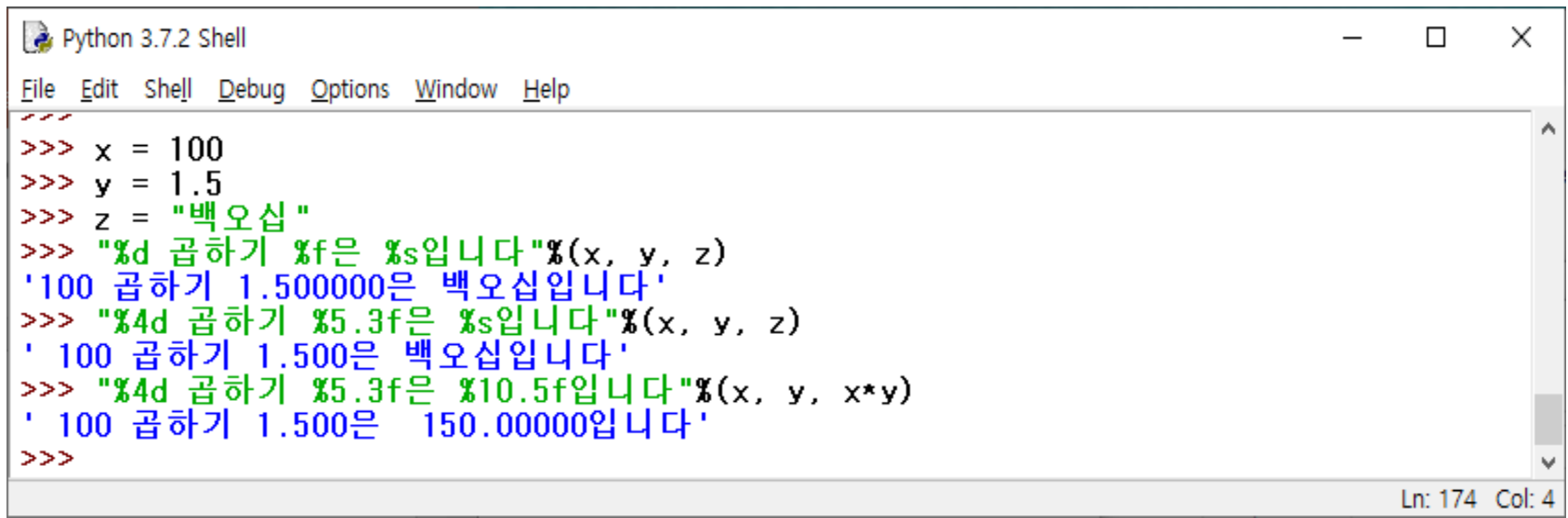
- ▶ 문자열에 변수의 값을 삽입하여 출력하고 싶으면 % 연산자 사용

```
>>> price = 10000  
>>> print("상품의 가격은 %s원입니다." % price)  
상품의 가격은 10000원입니다.
```

문자열 삽입 연산자 %

<문자열(%포함)> % (삽입값수식, 삽입값수식, 삽입값수식...)

- 왼쪽 문자열 안 위치 표시 종류
 - %s : 문자열
 - %3d : 3자리 정수
 - %5.2f : 부동소숫점 수(5칸에 소수점 이하 2개)
- 오른쪽 변수값 : 각 순서에 맞게 (,,,)로 여러 개 줄 수 있다.



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
>>> x = 100
>>> y = 1.5
>>> z = "백오십"
>>> "%d 곱하기 %f은 %s입니다"%(x, y, z)
'100 곱하기 1.500000은 백오십입니다'
>>> "%4d 곱하기 %5.3f은 %s입니다"%(x, y, z)
' 100 곱하기 1.500은 백오십입니다'
>>> "%4d 곱하기 %5.3f은 %10.5f입니다"%(x, y, x*y)
' 100 곱하기 1.500은 150.000000입니다'
>>>
```

Ln: 174 Col: 4

Lab: 거북이와 인사해보자

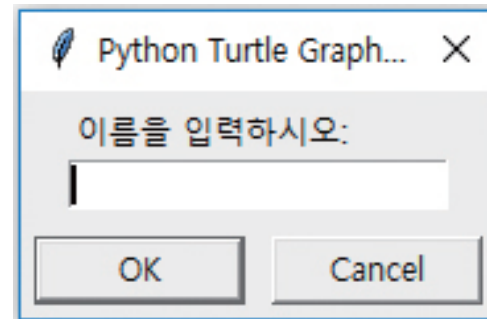


- ▶ 터틀 그래픽에서 사용자의 이름을 받아서 다음과 같이 출력해보자.



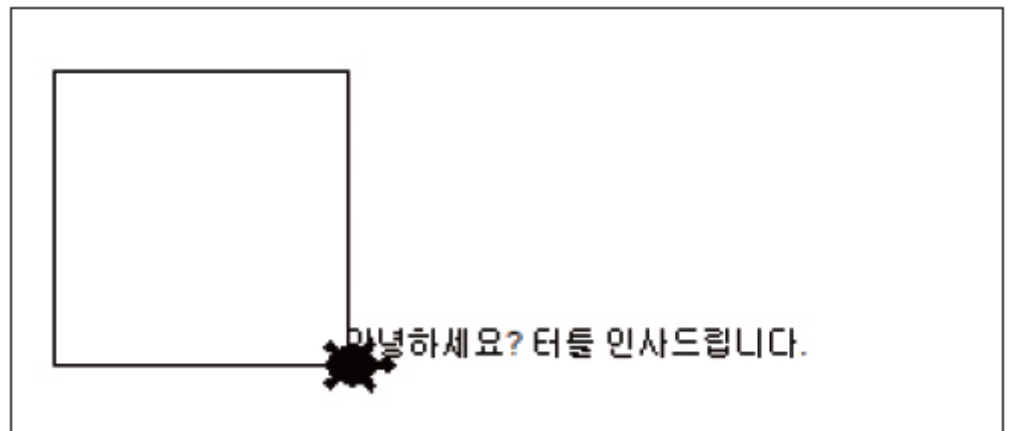
▶ 터틀 그래픽에서 문자열을 입력받는 방법

```
s = turtle.textinput("", "이름을 입력하시오: ")
```



▶ 터틀 그래픽에서 문자열을 출력하는 방법

```
t.write("안녕하세요? 터틀 인사드립니다.")
```



Solution

```
import turtle
t = turtle.Turtle()
t.shape("turtle")
s = turtle.textinput("", "이름을 입력하시오: ")
t.write("안녕하세요?" + s + "씨, 터틀 인사드립니다.")
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
t.left(90)
t.forward(100)
```

안녕하세요?
○○○ 씨
터틀 인사드립니다.



도전문제

사각형의 각 변에 "안녕하세요? 홍길동씨, 터틀 인사드립니다."을 출력해보자.

개별 문자 추출

- ▶ 문자열에서 개별 문자들을 추출하려면 -> 인덱스 연산 [:]을 사용한다.
 - ▣ “Monty Python”[6:10] → “Pyth”
 - ▣ “Monty Python”[-12:-7] → “onty”

						[6:10]					
0	1	2	3	4	5	6	7	8	9	10	11
M	o	n	t	y		P	y	t	h	o	n
-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1
[-12:-7]											

- ▶ [:]의 숫자는 생략 가능하다.(끝 표현)
- ▶ [::]로 추가로 간격도 줄 수 있다.

```
Python 3.7.2 S... - □ ×
File Edit Shell Debug Options
Window Help

>>> s = "Monty Python"
>>> s[6:10]
'Pyth'
>>> s[-12:-7]
'Monty'
>>> s[3:-6]
'ty '
>>> s[5:]
' Python'
>>> s[:-2]
'Monty Pyth'
>>> s[:]
'Monty Python'
>>>
```

Ln: 183 Col: 0

특수 문자열

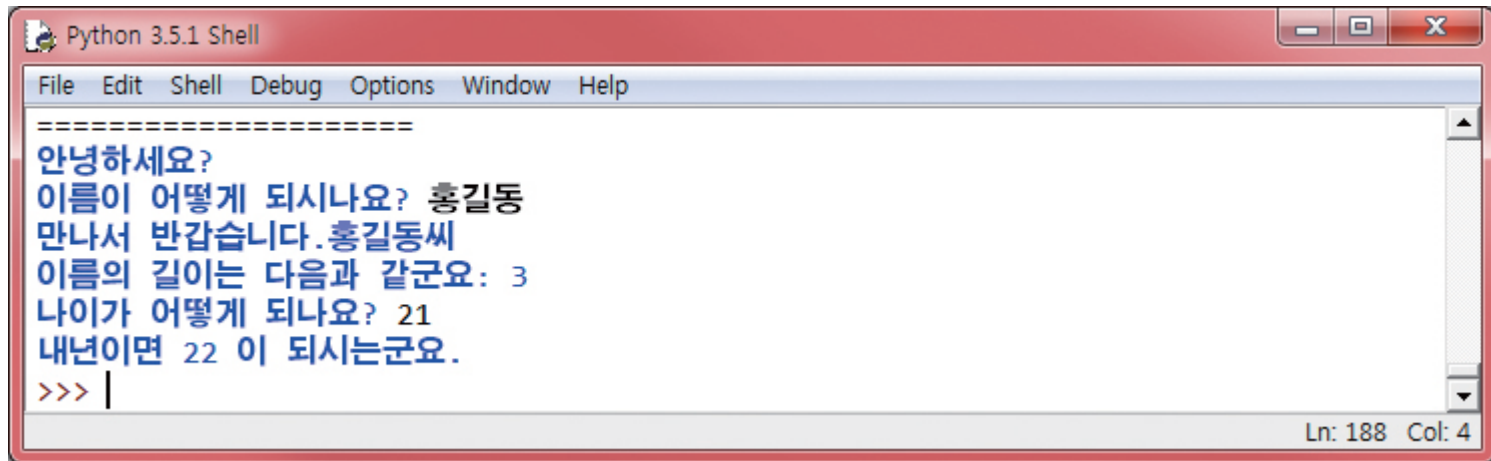
특수 문자열	의미
\n	줄 바꿈 문자
\t	탭 문자
\\	역슬래시 자체
\"	큰따옴표 자체
\'	작은따옴표 자체

```
>>> print("말 한마디로\n천 냥 빚을 갚는다")  
말 한마디로  
천 냥 빚을 갚는다
```

Lab: 친근하게 대화하는 프로그램



- ▶ 변수를 사용하여 사용자의 이름과 나이를 문자열 형태로 기억했다가 출력할 때 사용하는 프로그램을 작성해 보자.



```
Python 3.5.1 Shell
File Edit Shell Debug Options Window Help
=====
안녕하세요?
이름이 어떻게 되시나요? 홍길동
만나서 반갑습니다. 홍길동씨
이름의 길이는 다음과 같군요: 3
나이가 어떻게 되나요? 21
내년이면 22 이 되시는군요.
>>> |
```

Ln: 188 Col: 4

- ▶ 문자열의 길이를 계산할 때는 len(s)를 사용한다.



Run Python

Solution

```
print('안녕하세요?')
name = input('이름이 어떻게 되시나요? ')

print('만나서 반갑습니다.' + name + "씨")
print('이름의 길이는 다음과 같군요:', end=' ')
print(len(name))

age = int(input("나이가 어떻게 되나요? "))
print("내년이면", str(age+1), "이 되시는군요.")
```



도전문제

사용자에게 다른 정보도 물어보고 친근하게 다시 답변해보자. 예를 들어서 취미에 관하여 다음과 같이 질문할 수도 있다.

“취미가 무엇인가요?” “영화 감상”

“네 저도 영화 감상 좋아합니다.”

Lab: 연, 월, 일을 합하여 출력하기



- ▶ 문자열을 저장하는 변수를 사용하여 사용자가 입력하는 오늘의 연도, 월, 일을 모두 합하여 화면에 출력하는 프로그램을 작성해 보자.

오늘의 연도를 입력하시오: 2016

오늘의 월을 입력하시오: 12

오늘의 일을 입력하시오: 25

오늘은 2016년 12월 25일입니다.

Solution

```
year = input("오늘의 연도를 입력하시오: ")  
month = input("오늘의 월을 입력하시오: ")  
date = input("오늘의 일을 입력하시오: ")  
print("오늘은", year+"년", month+"월", date+"일입니다.")
```



도전문제

사용자로부터 현재 날짜를 받아서 “월/일/연도” 순으로 출력해보자.

Lab: 2050년에는 몇 살이 될



- ▶ 자신이 2050년에 몇 살이 될 것인지를 계산하는 프로그램을 작성해 보자.

올해는 2016입니다.
몇 살이신지요? 21
2050년에는 55살 이시군요.

```
import time
now = time.time()
thisYear = int(1970 + now//((365*24*3600)))
print("올해는 " + str(thisYear)+"입니다.")
```

Solution

```
import time

now = time.time()
thisYear = int(1970 + now//(365*24*3600))
print("올해는 " + str(thisYear)+"입니다.")

age = int(input("몇 살이신지요? "))
print("2050년에는 "+str(age + 2050-thisYear)+"살 이시군요.")
```



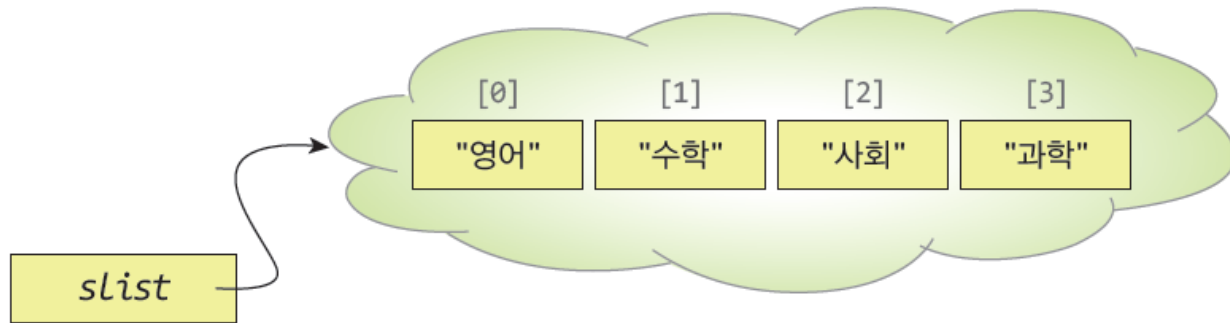
도전문제

str()을 사용하지 않고 print("올해는 ", thisYear, "입니다.")와 같이 쉼표를 사용하여 변수와 문자열을
동시 출력할 수 있는가? 위의 프로그램을 이런 식으로 변경해보자. 어떤 방법이 편리한가?

리스트

- ▶ 리스트(list): 여러 개의 자료들을 모아서 하나의 묶음으로 저장하는 것

```
slist = [ '영어', '수학', '사회', '과학' ]
```



리스트에 항목을 동적으로 추가

- ▶ 공백 리스트를 생성한 후에 코드로 리스트에 값을 추가하는 것

```
list = []  
list.append(1)  
list.append(2)  
list.append(6)  
list.append(3)  
  
print(list)
```

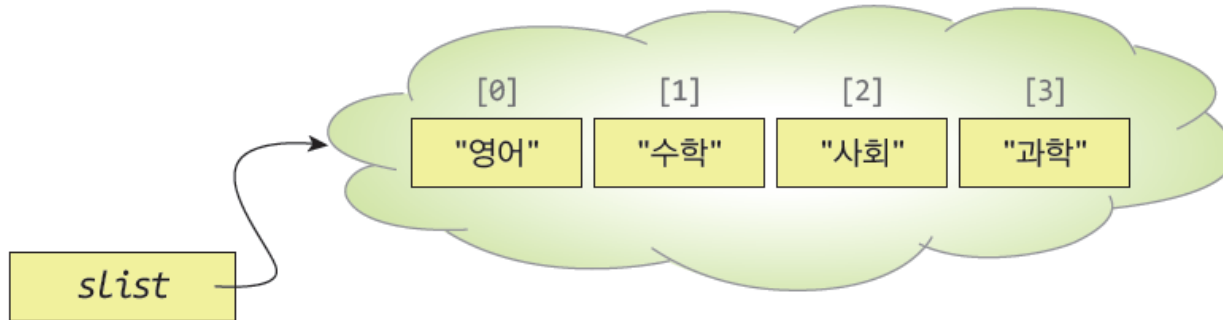
```
[1, 2, 6, 3]
```

리스트 원소 접근하기

- ▶ 리스트의 원소를 바꿀 수 있다!

```
slist = [ '영어', '수학', '사회', '과학' ]  
print(slist[0])
```

영어



리스트 원소 변경하기

- ▶ 리스트는 원소를 변경할 수 있다!
 - ▣ 리스트 원소는 지정문 왼쪽에도 올 수 있다.
 - ▣ 이명(Alias, 한 리스트를 두 이름으로...)에 주의!

```
>>> grades
['a', 'b', 'c', 'd', 'f']
>>> grades[0] = 'a+'
>>> grades
['a+', 'b', 'c', 'd', 'f']
>>> grades[2] = "c0"
>>> grades
['a+', 'b', 'c0', 'd', 'f']
```

```
>>> numbers = [1,2,3,4,5]
>>> integers = numbers
>>> numbers[1] = 20
>>> numbers
[1, 20, 3, 4, 5]
>>> integers
[1, 20, 3, 4, 5]
```

- ▣ 문자열은 변경할 수 없다.

```
>>> name = "joonseon Ahn"
>>> name[0] = "J"
Traceback (most recent call last):
  File "<pyshell#31>", line 1, in <module>
    name[0] = "J"
TypeError: 'str' object does not support item assignment
```

Lab: 친구들의 리스트 생성하기



- ▶ 제일 친한 친구 5명의 이름을 리스트에 저장했다가 출력하는 프로그램을 작성하자.

```
Python 3.5.2 Shell
File Edit Shell Debug Options Window Help
친구의 이름을 입력하시오: 홍길동
친구의 이름을 입력하시오: 강감찬
친구의 이름을 입력하시오: 이순신
친구의 이름을 입력하시오: 권율
친구의 이름을 입력하시오: 정약용
['홍길동', '강감찬', '이순신', '권율', '정약용']
>>> |
```

Ln: 23 Col: 4



Run Python

Solution

```
friend_list = [ ]

friend = input("친구의 이름을 입력하시오: ")
friend_list.append(friend)

friend = input("친구의 이름을 입력하시오: ")
friend_list.append(friend)

friend = input("친구의 이름을 입력하시오: ")
friend_list.append(friend)

friend = input("친구의 이름을 입력하시오: ")
friend_list.append(friend)

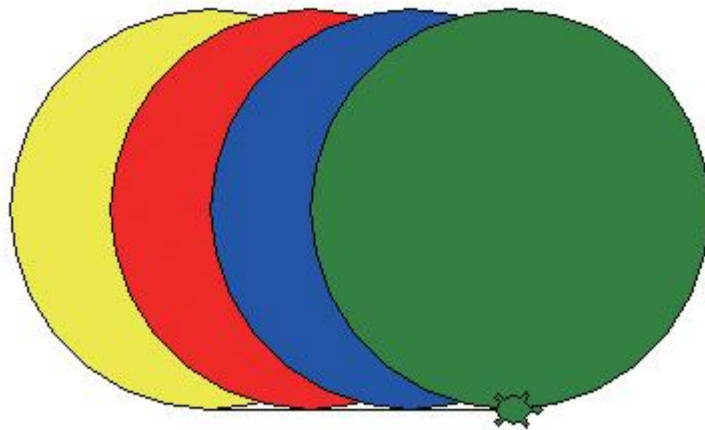
friend = input("친구의 이름을 입력하시오: ")
friend_list.append(friend)

print(friend_list)
```

Lab: 리스트에 저장된 색상으로 그리기



- ▶ 리스트에 색상을 문자열로 저장하였다가 하나씩 꺼내서 거북이의 채우기 색상으로 설정하고 원을 그려 보자.



Solution

```
import turtle
t = turtle.Turtle()
t.shape("turtle")

# 리스트를 사용하여 색상을 문자열로 저장한다.
color_list = [ "yellow", "red", "blue", "green" ]

t.fillcolor(color_list[0]) # 채우기 색상을 설정한다.
t.begin_fill() # 채우기를 시작한다.
t.circle(100) # 속이 채워진 원이 그려진다.
t.end_fill() # 채우기를 종료한다.

t.forward(50)
t.fillcolor(color_list[1]) # 채우기 색상을 설정한다.
t.begin_fill() # 채우기를 시작한다.
t.circle(100) # 속이 채워진 원이 그려진다.
t.end_fill() # 채우기를 종료한다.

...
```

이번 장에서 배운 것

- ✓ 파이썬에서 기본적인 자료형은 진리값, 정수, 실수, 문자열이다.
- ✓ 문자열은 큰따옴표("...")나 작은 따옴표('...')를 사용할 수 있다.
- ✓ 문자열을 정수로 변경하려면 `int()`를 사용한다.
- ✓ 문자열을 실수로 변경하려면 `float()`를 사용한다.
- ✓ 정수나 실수를 문자열로 변경하려면 `str()`을 사용한다.
- ✓ 여러 줄의 문자열은 `"""`로 표현 가능하다.
- ✓ `\n`은 줄 바꿈을 나타내는 특수 문자열이다.
- ✓ 리스트는 자료들을 모아서 저장할 수 있다.
- ✓ `%` 연산으로 문자열에 다른 값들을 적절히 삽입할 수 있다.
- ✓ 리스트와 문자열의 부분을 인덱스 연산 `[]`으로 자유롭게 읽거나 쓸 수 있다.
- ✓ 리스트는 변경이 가능하다. 그러나 문자열은 불변이다!



Q & A



질문은 동영상 댓글
또는 Teams 게시판에!

한국항공대학교 KOREA AEROSPACE UNIVERSITY Stream

홈 검색 내 콘텐츠 만들기 검색

컴퓨팅적 사고와 문제해결

한국항공대학교
안준선

0:00:03 / 1:11:15

세부 정보 사람

컴퓨팅적 사고와 문제해결 **chapt1**
안준선 항공전자정보공학부(교수) 이(가) 2020-03-12에 게시함 제한 46 0

컴퓨팅적 사고와 문제해결 1주차 이론수업 자료입니다.

설정 보기

댓글 0개

새로운 댓글 게시