

11장

사용자 상호작용

11장 사용자 상호작용

11.1 출력 보여 주기

11.2 사용자 입력 받기

11.3 요약

11.1 출력 보여 주기



11.1.1 식 출력하기

» print()

```
print("hello!") ---- 문자열
print(3 * 2 * (17 - 2.1)) ---- 수학식
print("abc" + "def") ---- 두 문자열 연결하기
word = "art" ---- 변수 만들기
print(word.replace("r", "n")) ---- "r"을 "n"으로 바꾸기
```

코드 11-1 식 출력하기

- print 괄호 안에 str 타입의 객체만 넣어야 하는 것은 아니다.
- 예를 들어 print(3 * 2 * (17 - 2.1))는 float 타입의 값을 계산한다.
- print 명령은 괄호 안에 어떤 타입의 객체가 들어가도 잘 작동한다.



11.1.2 여러 값 출력하기

» print 뒤의 괄호에 여러 객체를 넣을 수도 있다.

- 여러 다른 객체를 넣고 싶은 경우에는 각각을 콤마로 분리하면 된다.
- 파이썬 인터프리터는 각 객체의 값을 출력하면서 자동으로 사이에 공백을 넣어준다.
- 공백이 출력되지 않게 하려면 모든 객체를 문자열로 변환해 하나로 연결한 다음에 그 연결된 문자열을 print에 넘겨야 한다.

```
a = 1
b = 2
c = a/b
print(a,"/",b,"=",c)
add = str(a)+"/"+str(b)+"="+str(c)
print(add)
```

변수 초기화

나누기 계산

콤마를 사용해 정수(변수 a, b, c)와 문자열("/", "=")을 분리

str을 사용해 정수를 문자열로 변환한 후 + 연산자로 "/", "=" 문자열과 연결

문자열 출력

코드 11-2
여러 값 출력하기

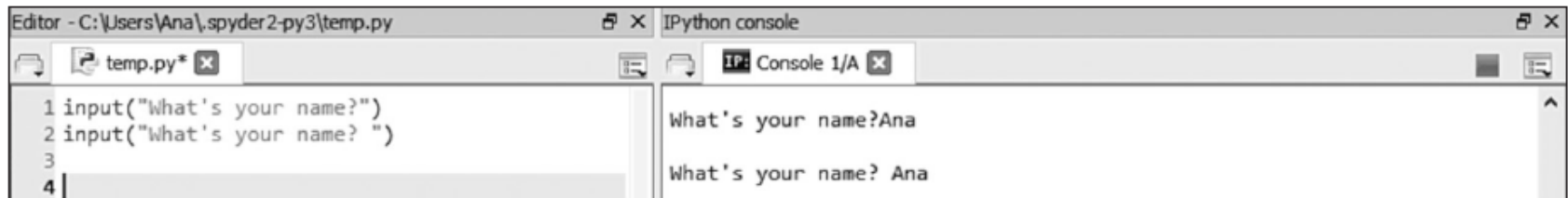
- 첫째 줄은 출력된 모든 객체 사이에 공백이 들어가 있지만, 둘째 줄은 공백이 없다.

» 셀프 체크 11.2

11.2 사용자 입력 받기

11.2.1 프롬프트를 표시해서 입력 요청하기

» input() 명령



```

Editor - C:\Users\Ana\spyder2-py3\temp.py
temp.py*
1 input("What's your name?")
2 input("What's your name? ")
3
4

IPython console
What's your name?Ana
What's your name? Ana
  
```

그림 11-1

프롬프트를 표시해서 사용자에게 입력을 요청하는 방법

- 그림 11-1을 보면 사용자가 입력한 내용이 프롬프트 바로 뒤에 표시된다.
- 그래서 프롬프트 문자열 맨 뒤에 공백을 넣어서 사용자 입력과 프롬프트를 구분할 수 있게 해주는 편이 낫다.

11.2.2 입력 읽기



» skip ...



11.2.3 입력을 변수에 저장하기

- » 사용자가 입력한 내용은 문자열로 변환된다.
- » 사용자 입력을 변수에 대입하면 입력 받은 값을 변수에 연결할 수 있다.

```
user_place = input("어디에 사십니까? ") ---- 사용자 입력을 받고 그 값을 user_place 변수에 저장
text = user_place.capitalize() + "!" ---- 사용자 입력에서 첫 글자를 대문자로 만든 문자열과
print(text)                                느낌표 하나로 이뤄진 문자열을 연결
print("거기는 참 좋은 곳이지요!") ---- 사용자 입력에 따라 적합한 출력을 표시
```

코드 11-4
사용자 입력 저장하기

- 사용자 입력을 문자열로 변환한 다음부터는 원하는 연산은 무엇이든 그 문자열에 적용할 수 있다.
- 예를 들어 문자열을 대문자나 소문자로 바꾸거나, 부분 문자열의 인덱스를 구하거나, 사용자가 입력한 문자열 안에 어떤 특정 문자열이 있는지 검색할 수도 있다.

11.2.4 사용자 입력을 다른 타입으로 변환하기

» 사용자가 입력한 모든 내용은 문자열로 저장된다. 하지만 숫자를 다루는 프로그램을 작성한다면 문자열을 입력 받아도 그리 편리하지 않다.

```
user_input = input("제곱을 계산할 수를 입력하시오: ") ---- 사용자 입력을 받아서 저장
num = int(user_input) ---- 사용자 입력을 정수로 변환
print(num*num) ---- 정수 값의 제곱을 계산해 출력. 이 프로그램의 첫 두 줄을
                        num = int(input("제곱을 계산할 수를 입력하시오: "))
                        라는 한 줄로 바꿀 수도 있음
```

코드 11-5

사용자 입력을 가지고 계산하기

- 사용자가 정수가 아닌 값을 입력하면 이 프로그램은 즉시 오류 메시지를 내면서 끝난다. 파이썬에는 정수가 아닌 값이 들어 있는 문자열을 수로 변환하는 방법이 없기 때문이다.
- 사용자가 제대로 된 수(정수)를 입력하는 경우라도, 그 입력이 정수처럼 보일지는 몰라도 실제로는 숫자들로만 이뤄진 문자열임을 기억해야 한다.
- 사용자가 5를 입력하더라도 파이썬은 이를 "5"라는 문자열로 취급한다.
→ 따라서 수를 얻으려면 문자열을 수로 타입 변환(casting, 캐스팅)해야 한다.
- 타입을 변환하려면 문자열 객체 앞에 타입 이름인 int를 표시하되, 괄호로 둘러싸서 표시하면 된다



11.2.5 더 많은 입력 요청하기

» 사용자에게 입력을 여러 번 받는 프로그램을 짤 수도 있다.

```
num1 = float(input("첫 번째 수를 입력하시오: ")) ---- 수를 입력 받아서 실수(float)로 바꿈
num2 = float(input("두 번째 수를 입력하시오: ")) ---- 다른 수를 입력 받아서 실수(float)로 바꿈
print(num1, "*", num2, "=", num1*num2 ) ---- 사용자가 입력한 두 실수와 결과를 예쁘게 출력
```

코드 11-6

둘 이상의 사용자 입력을 사용해 계산하기

- 이 프로그램은 계산 결과만 출력하는 대신에 어떻게 그 결과를 계산했는지도 표시한다.
- 사용자가 4.1과 2.2를 입력하면 프로그램은 $4.1 * 2.2 = 9.02$ 를 출력한다.



11.3 요약

- » (Q11.1) 사용자에게 수를 두 개 입력 받는 프로그램을 작성하라. 두 수를 각각 b와 e라는 변수에 저장하라. 프로그램은 be을 적절한 메시지와 함께 표시해야 한다(참고: 제공은 ** 연산자를 사용한다).
- » (Q11.2) 사용자 이름과 나이를 묻는 프로그램을 작성하라. 각 값을 저장하는 변수에 적당한 이름을 붙여야 한다. 사용자가 25년 후 몇 살이 될지 계산하라. 예를 들어 사용자가 현석과 44를 입력하면 프로그램은 "안녕 현석! 25년 후 당신은 69살이야!"라고 표시해야 한다.