

파이썬

6강. 변수와 자료형(표준입출력장치)

1. 표준입출력장치

- 일반적으로 컴퓨터 시스템에서 표준입력장치는 키보드를 의미하고, 표준출력장치는 모니터를 의미한다.
- 프로그래밍에서 키보드로 값을 입력받는 방법과 콘솔로 결과를 출력하는 방법에 대해 서 알아본다.

2. 표준입력장치

- 표준입력장치로 대표되는 키보드로 부터 임의의 값을 입력받아서 프로 그래밍을 작성할 수 있다.
- 프로 그램을 단위 테스트할 목적으로 소량의 데이터를 키보드로 입력 받아서 로직(logic)의 처리과정을 확인 하는 경우에 주로 이용된다.
- 키보드로 입력받는 input()함수의 형식은 다음과 같다.

형식 ↵ ↵

`input(prompt)`

`builtins ↵`

- input은 builtins 모듈에서 제공되는 내장함수이다. 프롬프트(prompt)는 키보드로 입력할 경우 콘솔 에 출력되는 메시지 역할을 한다. 예를 들면 "숫자 입력 : " 으로 프롬프트를 지정할 경우 통합개발도구 의 콘솔에는 '숫자 입력 : >?' 형식으로 나타난다.
- 사용자는 ? 뒤에 키보드로 값을 입력하면 된다. 한편 키보드로 입력한 모든 값은 문자열로 처리된다는 점을 꼭 기억할 필요가 있다. 만약 숫자 를 입력할 경 우에는 숫자형으로 변경해주어야 한다.

2. 표준입력장치

chapter02.lecture.step03_std_input_output.py ↵

Python Console ↵

(1) 문자형 숫자 입력 ↵

num = input("숫자 입력 : ") ↵

print('num type : ', type(num)) # <class 'str'> ↵

print('num = ', num) ↵

print('num = ', num*2) ↵

숫자 입력 : >? 100 ↵

num type : <class 'str'> ↵

num = 100 ↵

num = 100100 ↵

(2) 문자형 숫자를 정수형으로 변환 ↵

num1 = int(input("숫자 입력 : ")) # str -> int
(형변환) ↵

print('num1 = ', num1*2) ↵

숫자 입력 : >? 100 ↵

num1 = 200 ↵

(3) 문자형 숫자를 실수형으로 변환 ↵

num2 = float(input("숫자 입력 : ")) # str -> float
(형변환) ↵

result = num1 + num2 # 실수 = 정수 + 실수 ↵

print('result = ', result) ↵

숫자 입력 : >? 200 ↵

result = 300.0 ↵

2. 표준입력장치

(1) 문자형 숫자 입력

- 숫자 100을 키보드로 입력하여 num 변수에 할당한 후 자료형을 출력하면 `<class 'str'>`라고 출력된다. 따라서 문자형의 객체이기 때문에 연산이 불가능하다. `num*2`는 곱셈 연산이 아니고, num 문자열을 2회 반복하는 연산이 실행된다.

(2) 문자형 숫자를 정수형으로 변환

- 숫자 100을 키보드로 입력받아서 정수형(int)으로 변환 후 num1 변수에 할당된다. 따라서 `num1*2`수식에 의해서 200이 계산된다.

(3) 문자형 숫자를 실수형으로 변환

- 숫자 200을 키보드로 입력받아서 실수형(float)으로 변환 후 num2 변수에 할당된다. 따라서 `num1 + num2` 수식에 의해서 300이 계산되어 result에 할당된다

3. 표준출력장치

- 표준출력장치로 대표되는 모니터는 통합개발도구(Pycharm)에서는 콘솔(Console)로 불려진다.
- 콘솔로 결과를 출력하는 print()함수의 형식은 다음과 같다.

형식 ↵ ↵

```
print(values, sep, end, file, flush)
```

builtins ↵

- print() 함수는 5개의 인수를 사용할 수 있다. values, sep, end 등의 인수를 자주 사용하고, file, flush 인수는 자주 사용되지 않는다.
- 이처럼 특정 함수를 정확히 사용하기 위해서는 함수의 용도와 인수의 역할을 정확히 알 필요가 있다.
- 하지만 파이썬에서 제공되는 수많은 함수의 사용법을 완벽하게 숙지하는 것은 불가능하다.
- 따라서 함수의 용도, 인수의 역할 등을 확인하기 위해서는 help() 함수를 이용한다.

3. 표준출력장치

- >>> help(print)
- Help on built-in function print in module builtins:
- print(...)
- print(value, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
- Prints the values to a stream, or to sys.stdout by default. Optional keyword arguments:
- file: a file-like object (stream); defaults to the current sys.stdout. sep: string inserted between values, default a space.
- end: string appended after the last value, default a newline.
- flush: whether to forcibly flush the stream.

3. 표준출력장치

- 첫 번째에 나타난 built-in function print in module builtins: 문장의 의미는 기본 모듈에 포함된 내장함수라는 의미이다.
- 기본 모듈이란 파이썬 설치 시 자동으로 설치되는 모듈을 의미한다.
- 이러한 내장함수는 별도의 추가 작업 없이 바로 사용할 수 있는 함수이다.
- 두 번째 이후 문장들은 print 함수에서 사용할 수 있는 5개 인수의 역할에 대해서 상세하고 있다.
- 아쉽게도 모든 도움말은 영문으로 제공된다.
- 단적인 예이지만 IT 관련 분야를 공부하기 위해서는 기본으로 영문법이나 독해 실력도 요구된다.
- 독해에 어려움이 있는 사용자는 포털 사이트에서 제공되는 번역기를 이용하기 바란다.

3. 표준출력장치

chapter02.lecture.step03_std_input_output.py ↵

Python Console ↵

(1) value 인수 ↵

print("value =", 10 + 20 + 30 + 40 + 50) ↵

value = 150 ↵

(2) sep 인수 : 값과 값을 특수문자로 구분 ↵

print("010", "1234", "5678", sep="-") ↵

010-1234-5678 ↵

(3) end 인수 ↵

print("value=", 10, end = ", ") ↵

print("value=", 20) ↵

value= 10, value= 20 ↵

3. 표준출력장치

(1) value 인수

print 함수에서 값(values) 속성은 콤마(,)에 의해서 구분된다. 첫 번째 값은 "value ="이 되고 두 번째 값은 $(10 + 20 + 30 + 40 + 50)$ 의 계산 결과 150이 된다.

(2) sep 인수 : 값과 값을 특수문자로 구분

sep 인수는 임의의 특수문자로 값들을 구분하는 역할을 한다. 전화번호 3개 숫자열을 하이픈(-)으로 구분하여 출력해준다.

(3) end 인수

print는 기본적으로 값을 출력한 후 줄바꿈을 수행한다. end="구분자" 형식으로 인수를 사용하면 다음에 출력되는 내용이 현재 출력되는 내용과 구분자에 의해서 같은 줄에 구분되어 출력된다.

4. format과 양식문자

- format()함수는 값(value)을 지정한 형식(format)으로 변환해주는 함수이다. 주로 수학 또는 과학 기술 관련 상수, 화폐단위 등을 출력하는 경우 print()함수 안에서 이용된다.
- format()함수는 다음 형식과 같이 주어진 값(value)을 지정한 양식(format)으로 변환해주는 함수이다.

형식 ↵ ↵

`format(value, "format")`

`builtins ↵`

4. format과 양식문자

- 양식문자는 'print("%양식문자" %(값))' 형식으로 print()함수 안에서 일정한 양식을 제공하는 양식문자를 이용하여 정수, 실수, 문자열 등을 출력할 수 있다. 주요 양식문자는 다음과 같다.

양식문자 ↴	출력 형식 ↴
%d ↴	10진수 정수 ↴
%o ↴	8진수 정수 ↴
%x ↴	16진수 정의 ↴
%f ↴	실수(%f전체자릿수.소수점자릿) ↴
%s ↴	문자열 ↴
%c ↴	단일 문자열 ↴

4. format과 양식문자

chapter02.lecture.step03_std_input_output.py ↵

```
# (4) format() 함수 인수 : format(value, "format")  
print("원주율=", format(3.14159, "8.3f")) ↵  
print("금액 =", format(10000, "10d")) ↵  
print("금액 =", format(125000, "3,d")) ↵
```

```
# (5) 양식문자 인수 : print( "%양식문자" %(값) )  
name = "홍길동" ↵  
age = 35 ↵  
price = 125.456 ↵  
print("이름 : %s, 나이 : %d, data = %.2f" %(name, age, price)) ↵
```

Python Console ↵

```
원주율=   3.142 ↵  
금액 =    10000 ↵  
금액 = 125,000 ↵
```

```
↵  
↵  
↵  
↵  
↵  
이름 : 홍길동, 나이 : 35,  
data = 125.46 ↵
```

3. 표준출력장치

(4) format() 함수 인수

원주율을 출력하기 위해서 `format(3.14159, "8.3f")` 형식으로 적용하면 원주율을 전체 8개 자릿수 를 기준으로 소수점 이하 3째 자리까지 표시 된다. 전체 자릿수에서 빈 자리는 공백으로 채워지고, 소숫점은 4째 자릿수에서 반올림된다.

(5) 양식문자 인수 :

`name`은 `%s` 양식문자, `age`는 `%d` 양식문자, `price`는 `%f` 양식문자를 이용하여 출력하는 예문이다.

`%.2f`는 실수형 자료를 대상으로 소숫점 2자리까지 표기하는 양식이다.

4. 외부상수 출력

- 'print("{}" .format(값))' 형식으로 print()함수 안에서 {}기호를 이용해서 외부 값을 순서대로 받아서 출력한다.
- '{n}' 형식으로 기호 안에 순번을 입력하여 외부 값을 순서대로 받을 수 있고, 같은 순서를 여러 번 쓸 수 있다. SQL문을 작성할 경우 유용하다.

chapter02.lecture.step03_std_input_output.py

Python Console

```
# (6) 외부 상수 인수
```

```
print("이름 : {}, 나이 : {}, data={}".format(  
    (name, age, price) )  
print("이름 : {1}, 나이 : {0}, data={2}".form  
at(age, name, price) )
```

```
이름 : 홍길동, 나이 : 35, data = 1  
25.46
```

```
이름 : 홍길동, 나이 : 35, data=12  
5.456
```

```
# (7) format 축약형 (SQL문 작성)
```

```
uid = input("id input : ")  
query = f"select * from member where uid =  
{uid}"  
print(query) # member 테이블에서 uid가 hong인  
레코드 검색
```

```
id input : >? hong  
select * from member where u  
id = hong
```

4. 외부상수 출력

(6) 외부 상수 인수

- format()의 인수로 지정한 name, age, price의 값을 {} 순서대로 받아서 출력한다. {}안에서 순번을 지정하여 값을 받을 수 있다.

(7) format 축약형(SQL문 작성)

- 'format(값)'을 '{값}' 형식으로 직접 {}기호 안에 값을 표기하고 따옴표 왼쪽에 f를 표시하는format 함수의 축약형이다. 이러한 형식은 DB의 SQL문 작성 시 유용하다.

THANK YOU