Chapter3 : 화면 입출력과 리스트 다루기

프로그램과 소통하는 방법

Graphical User Interface

Command Line Interface

1.Command Line Interface

-GUI와 달리 Text를 사용하여 컴퓨터에 명령을 입력하는 인터페이스 체계

-Console = Terminal = CMD창

콘솔창 입출력

-input() 함수는 콘솔창에서 문자열을 입력 받는 함수

-콤마(,) 사용할 경우 print 문이 연결됨

```
>>> print ("Hello World!", "Hello Again!!!")
Hello World! Hello Again!!!
```

-숫자 입력 받기

```
temperature = float(input("온도를 입력하세요 :"))

print(temperature)
```

2.Print formatting

%-format

"%datatype" % (variable) 형태로 출력 양식을 표현

```
print("I eat %d apples." % 3)
print("I eat %s apples." % "five")
number = 3; day="three"
print("I ate %d apples. I was sick for %s days."
      % (number, day))
print("Product: %s, Price per unit: %f." % ("Apple", 5.243))
```

type	설명
%s	문자열 (String)
%c	문자 1개(character)
%d	정수 (Integer)
%f	부동소수 (floating-point)
%o	8진수
%x	16진수
%%	Literal % (문자 % 자체)

Str.format()

"~~~~{datatype}~~~~".format(argument)

```
age = 36; name='Sungchul Choi'
print("I'm {0} years old.".format(age))
print("My name is {0} and {1} years old.".format(name,age))
print(
  "Product: {0}, Price per unit: {1:.3f}.".format(
    "Apple", 5.243))
```

Padding

여유 공간을 지정하여 글자배열 + 소수점 자릿수를 맞추기

```
print("Product: %5s, Price per unit: %.5f." % ("Apple",
5.243))
print("Product: {0:5s}, Price per unit:
{1:.5f}.".format("Apple", 5.243))
print("Product: %10s, Price per unit: %10.3f." % ("Apple",
5.243))
print("Product: {0:>10s}, Price per unit:
{1:10.3f}.".format("Apple", 5.243))
```

Naming

naming

해당 표시할 내용을 변수로 표시하여 입력

```
print("Product: %(name)10s, Price per unit: %(price)10.5f." %
      {"name":"Apple", "price":5.243})

print("Product: {name:>10s}, Price per unit:
{price:10.5f}.".format(name="Apple", price=5.243))
```

3. List Data Type

List : 시퀀스 자료형, 여러 데이터들의 집합, Int, Float 같은 다양한 데이터 Type 포함
인덱싱: list에 있는 값들은 주소를 가진다.

슬라이싱 : list의 값들을 잘라서 쓰는 것, list의 주소 값을 기반으로 부분 값을 반환

Editor

```
cities = ['서울', '부산', '인천', '대구', '대전', '광주', '울산', '수원']
print (cities[0:6], " AND ", a[-9:]) # a 변수의 0부터 5까지, -9부터 끝까지
print (cities[:])    # a변수의 처음부터 끝까지
print (cities[-50:50])  # 범위를 넘어갈 경우 자동으로 최대 범위를 지정
print (cities[::2], " AND ", a[::-1]) # 2칸 단위로, 역으로 슬라이싱
```

리스트 추가와 삭제

- append, extend, insert, remove, del 등 활용

Python Shell

```
# 이전 장과 연결 돼서 실행
>>> color.append("white")       # 리스트에 "white" 추가
>>> color.extend(["black","purple"]) # 리스트에 새로운 리스트 추가
>>> color.insert(0,"orange")    # 0번째 주소에 "orange" 추가
>>> print (color)
['orange', 'yellow', 'blue', 'green', 'white', 'black', 'purple']
>>> color.remove("white")       # 리스트에 "white" 삭제
>>> del color[0]                # 0번째 주소 리스트 객체 삭제
>>> print (color)
['yellow', 'blue', 'green', 'black', 'purple']
```

Python 리스트만의 특징 : 다양한 Data Type이 하나에 list에 들어감

Python Shell

```
>>> a = [5, 4, 3, 2, 1]
>>> b = [1, 2, 3, 4, 5]
>>> b = a
>>> print (b)
[5, 4, 3, 2, 1]
>>> a.sort()
>>> print (b)
[1, 2, 3, 4, 5]
>>> b = [6,7,8,9,10]
>>> print (a, b)
[1, 2, 3, 4, 5] [6, 7, 8, 9, 10]
```

'='의 의미는 같다가 아닌 메모리 주소에 해당 값을 할당한다는 의미

패킹과 언패킹

패킹 : 한 변수에 여러 개의 데이터를 넣는 것

언패킹 : 한 변수의 데이터를 각각의 변수로 반환