

10. 문자열 포매팅

포매팅

- 문자열안에 여러가지 세부 정보를 포함하여 출력
- . 문자열 사이사이에 특정 정보를 삽입하여 하나의 문자열을 만들어 내는 기능.

표식	설명
%d	정수
%f	실수
%s	문자열
%c	문자 하나
%h	16진수
%o	8진수
%%	% 문자

> 문자열 포매팅 사용 예제 1

```
Chap01_intro > 08.String[포매팅].py > ...
1 station = '삼송'
2 print('이번 정류장은 %s 역 입니다.' % station)
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/08.String[포매팅].py
이번 정류장은 삼송 역 입니다.

- * 문자열 내에 %[표식] 을 정의 해 둔 후 문자열이 끝나고 어떤 내용을 넣을것인지 매핑한다.

> 문자열 포매팅의 사용 예제 2

- * 문자열 내에 배치할 문자열 을 () 안에 나란히 정의 해 둘 수 있다.
- * 삼항 연산자 등의 로직을 추가 할 수 있다.

```
Chap01_intro > 08.String[포매팅].py > ...
1 station = '구로'
2 direction = False
3
4 print('이번 역은 %s 역 입니다. 내리실 문은 %s 입니다.'
5       % (station, '오른쪽' if direction else '왼쪽'))
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/08.String[포매팅].py
이번 역은 구로 역 입니다. 내리실 문은 왼쪽 입니다.

> 삼항 연산자

```
'오른쪽' if direction else '왼쪽'
```

- * direction 의 비교 연산 이 True 이면 '오른쪽' False 면 왼쪽

문자열의 서식

- 좌 우 폭을 지정

. % 에 옵션을 두어 왼쪽에 여백을 두는 예

```
Chap01_intro > 08.String[포맷].py > ...
1 value = 123
2 print('###%d###' % value)
3 print('###%5d###' % value)
4 print('###%10d###' % value)
5 print('###%1d###' % value)
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/Python.exe C:/Users/MasterD/AppData/Local/Programs/Python/Python311/Python.exe 08.String[포맷].py
###123###
### 123###
### 123###
### 123###
```

>2행 : '###123###' 출력

>3행 : 123 좌측에 2자리 공백을 추가후 총 5자리 포함시켜 출력

>4행 : 123 좌측에 7자리 공백을 추가후 총 10자리 포함시켜 출력

>5행 : 123 은 이미 3자리 이므로 추가할 공백없이 그냥 출력

. 활용 예제

> 자리수 가 동일하지 않는 숫자를 정렬 하여 표현 할 경우

* %7d

```
Chap01_intro > 08.String[포맷].py > ...
1 price = [30,13500,2000]
2 for p in price :
3     print('가격 %d 원' %p)
4 for p in price :
5     print('가격 %7d 원' %p)
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/Python.exe C:/Users/MasterD/AppData/Local/Programs/Python/Python311/Python.exe 08.String[포맷].py
가격 30 원
가격 13500 원
가격 2000 원
가격 30 원
가격 13500 원
가격 2000 원
```

>5행 : '가격 ' 문자열 뒤에 숫자 를 총 7자리 로 만드는데
왼쪽부터 공백을 채워 7자리로 만들어 표현

- 실수 의 서식

. 폭 지정 이외에 소수 이하의 자리수 제어 가능

```
Chap01_intro > 08.String[포맷].py > ...
1 pie = 3.1459265
2 print('%10f' % pie)
3 print('%10.8f' % pie)
4 print('%10.5f' % pie)
5 print('%10.2f' % pie)
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/Python.exe C:/Users/MasterD/AppData/Local/Programs/Python/Python311/Python.exe 08.String[포맷].py
3.14592650
3.14592650
3.14593
3.15
```

>2행 : 그대로 표현

>3행 : 소수점 포함 총 10자리, 소수 이하 8자리까지 표현 (9자리 에서 반올림)

>4행 : 소수점 포함 총 10자리, 소수 이하 5자리까지 표현 (6자리 에서 반올림)

>5행 : 소수점 포함 총 10자리, 소수 이하 2자리까지 표현 (3자리 에서 반올림)

신형 포매팅

- {} 괄호를 통한 문자열 가공

.format()

> 기본 적인 사용예시

```
Chap01_intro > 08.String[포맷].py > ...
1 Name = '관리자'
2 IDvalue = 'ADMIN'
3 MailAddress = 'System@gmail.com'
4 print('이름 : {}, ID : {}, 메일주소 : {}'.format(Name, IDvalue, MailAddress))
5
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:/Python/Chap01_intro/08.String[포맷].py
이름 : 관리자, ID : ADMIN, 메일주소 : System@gmail.com

> {} 괄호 안에 순서를 정하여 Format() 의 인수의 순서를 배치 할 수 있다.

```
Chap01_intro > 08.String[포맷].py > ...
1 Name = '관리자'
2 IDvalue = 'ADMIN'
3 MailAddress = 'System@gmail.com'
4 print('이름 : {2}, ID : {0}, 메일주소 : {1}'.format(IDvalue, MailAddress, Name))
5
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:/Python/Chap01_intro/08.String[포맷].py
이름 : 관리자, ID : ADMIN, 메일주소 : System@gmail.com

> {} 괄호 안에 변수 이름을 적어 두고 format 인수열에 변수의 값을 나열 할 수 있다.

```
Chap01_intro > 08.String[포맷].py
1 print('이름 : {Name}, ID : {IDvalue}, 메일주소 : {MailAddress}')
```

```
2     .format(IDvalue='ADMIN', MailAddress='System@gmail.com', Name = '관리자'))
3
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:/Python/Chap01_intro/08.String[포맷].py
이름 : 관리자, ID : ADMIN, 메일주소 : System@gmail.com

> Key 와 Value를 가지는 자료형 을 생성하여 매핑 할 수 있다.

```
Chap01_intro > 08.String[포맷].py > ...
1 User = {'IDvalue' : 'ADMIN',
2         'MailAddress' : 'System@gmail.com',
3         'Name' : '관리자'}
4
5 print('이름 : {0[Name]}, ID : {0[IDvalue]}, 메일주소 : {0[MailAddress]}')
```

```
6     .format(User))
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:\Users\MasterD\AppData\Local\Programs\Python\Python311\python.exe d:/Python/Chap01_intro/08.String[포맷].py
이름 : 관리자, ID : ADMIN, 메일주소 : System@gmail.com

> 서식 을 이용한 정밀한 표현 가능 1

```
Chap01_intro > 08.String[포맷].py > ...
1 MailAddress = 'System@gmail.com'
2 Name        = '관리자'
3 Age         = 24
4
5 print('이름 : {0:s}, 나이 : {1:d}, 메일주소 : {2:s}'
6       .format(Name, Age, MailAddress))
7
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

이름 : 관리자, ID : ADMIN, 메일주소 : System@gmail.com

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/08.String[포맷].py
이름 : 관리자, 나이 : 24, 메일주소 : System@gmail.com
```

* 데이터 의 위치 에 데이터 형식 을 지정 한 예

> 서식 을 이용한 정밀한 표현 가능 2

```
Chap01_intro > 08.String[포맷].py > ...
1 Name        = '관리자'
2 Age         = 24
3 height      = 177.223
4
5 print('이름 : {0:10s}, 나이 : {1:5d}, 키 : {2:8.2f}'
6       .format(Name, Age, height))
7
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/08.String[포맷].py
이름 : 관리자, 나이 : 24, 키 : 177.22
```

- > {0:10s} : 0번 인덱스 인자
총 10자리 로 함
있는 문자열 을 표현하고
나머지는 오른쪽에서 공백으로 채움
- > {1:5d} : 1번 인덱스 인자
총 5자리 로 함
있는 숫자를 표현하고
나머지는 왼쪽에서 공백으로 채움
- > {2:8.2f} : 2번 인덱스 인자
총 8자리로 함
소수 2자리까지 표현(3번째 자리에서 반올림)

> 서식 을 이용한 정밀한 표현 가능 3

```
Chap01_intro > 08.String[포맷].py > ...
1 Name        = '관리자'
2 Age         = 24
3 height      = 177.223
4
5 print('이름 : {0:^10s}, 나이 : {1:<5d}, 키 : {2:>8.2f}'
6       .format(Name, Age, height))
7
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/08.String[포맷].py
이름 : 관리자, 나이 : 24, 키 : 177.22
```

- * '^' : 가운데 정렬
- * '<' : 왼쪽 정렬
- * '>' : 오른쪽 정렬

> 서식 을 이용한 정밀한 표현 가능 4

* 공백 대신 처리할 문자 나 숫자 를 입력 지정 할 수 있다.

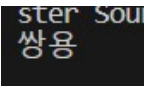
```
Chap01_intro > 08.String[포맷].py > ...
1 Name        = '관리자'
2 Age         = 24
3 height      = 177.223
4
5 print('이름 : {0:$^10s}, 나이 : {1:~<5d}, 키 : {2:~>8.2f}'
6       .format(Name, Age, height))
7
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

D:\Python>C:/Users/MasterD/AppData/Local/Programs/Python/Python311/python.exe d:/Python/Chap01_intro/08.String[포맷].py
이름 : $$$관리자$$$$$, 나이 : 24____, 키 : 00177.22
```

실습

1. "차종 : 코란도C, 제조사 : 쌍용, 배기량: 1998" 문자열에서 "제조사" 만추출하여 출력하세요.



2. 임의의 주민등록번호 13 자리를 입력받아 현재 나이 와 성별을 출력하는 프로그램을 작성해 보세요

- '-' 를 제외 한 숫자를 입력하지 않은 경우 예외 처리 한다.
- 13자리 를 입력하지 않은경우 의 예외상황 처리 한다.
- 정확한 결과를 출력할때까지 반복한다.

